NuMicro® Family
Arm® Cortex-A35-based Microprocessor

MA35D1 Series
Technical Reference Manual

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller and microprocessor based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

TABLE OF CONTENTS

1 GENERAL DESCRIPTION 29
1.1 Core Sub-system 29
1.2 Memory Sub-system 29
1.3 Security Sub-system 30
1.4 Display and Video Sub-system 30
1.5 Connectivity and I/O Sub-system 30
2 FEATURES 32
2.1 MA35D1 Series Features 32
3 PARTS INFORMATION 46
3.1 MA35D1 Series Package Types 46
3.2 MA35D1 Series Naming Rule 47
3.3 MA35D1 Series Selection Guide 48
3.3.1 LQFP Package Types (New) 48
3.3.2 BGA Package Types (New) 50
3.3.3 LQFP Package Types 52
3.3.4 BGA Package Types 54
4 PIN CONFIGURATION 56
4.1 Pin Configuration 56
4.1.1 MA35D1 Series LQFP-EP 216-Pin Pinout Diagram 56
4.1.2 MA35D1 Series BGA 312-Ball Pinout Diagram 57
4.1.3 MA35D1 Series BGA 364-Ball Pinout Diagram 58
4.2 Pin Description 59
4.2.1 MA35D1 Series LQFP-EP 216-Pin Pinout Function Table 59
4.2.2 MA35D1 Series BGA 312-Ball Pinout Function Table 75
4.2.3 MA35D1 Series BGA 364-Ball Pinout Function Table 86
4.2.4 MA35D1 Series Multi-function Description Table 99
5 BLOCK DIAGRAM 118
5.1 MA35D1 Series Block Diagram 118
6 FUNCTIONAL DESCRIPTION 119
6.1 Arm Cortex-A35 Core 119
6.1.1 Overview 119
6.1.2 Features 119
6.1.3 Block Diagram 120
6.1.4 Functional Description 121
6.1.5 Generic Timer 122
6.1.6 Generic Interrupt Controller (GIC) 123

6.2 Arm Cortex-M4 Core 132
6.2.1 Overview 132
6.2.2 Features 132
6.2.3 Block Diagram 134
6.2.4 System Timer (SysTick) 135
6.2.5 Nested Vectored Interrupt Controller (NVIC) 140
6.2.6 System Control Register 168
6.3 Arm TrustZone for Armv8-A 178
6.3.1 Overview 178
6.3.2 TrustZone of Cortex-A35 Processor 178
6.3.3 TrustZone System Architecture (SSMCC, SSPCC) 179
6.4 System Manager 181
6.4.1 Overview 181
6.4.2 System Reset 182
6.4.3 System Power Distribution 186
6.4.4 Memory Organization 188
6.4.5 Power-On Setting 193
6.4.6 Control Registers Access Attribute 196
6.4.7 Register Map 199
6.4.8 Register Description 202
6.4.9 NMI Control Register 298
6.5 Clock Controller 303
6.5.1 Overview 303
6.5.2 Clock Generator 307
6.5.3 CA35 CPU Clock, System Clock and SysTick Clock 309
6.5.4 Peripherals Clock 311
6.5.5 PLL Settings and Operation Modes 311
6.5.6 Power-down Mode Clock 315
6.5.7 Clock Output 316
6.5.8 Control Registers Access Attribute 316
6.5.9 Register Map 318
6.5.10 Register Description 320
6.6 SSPCC 370
6.6.1 Overview 370
6.6.2 Features 370
6.6.3 Block Diagram 370
6.6.4 Functional Description 370
6.6.5 Register Map 387
6.6.6 Register Description 390
6.7 SSMCC 432
6.7.1 Overview 432

6.7.2 Features 432
6.7.3 Block Diagram 432
6.7.4 Functional Description 432
6.7.5 Register Map 437
6.7.6 Register Description 442
6.8 True Random Number Generator (TRNG) 476
6.8.1 Overview 476
6.8.2 Features 476
6.9 Hardware Semaphore (HWSEM) 477
6.9.1 Overview 477
6.9.2 Features 477
6.9.3 Block Diagram 477
6.9.4 Basic Configuration 477
6.9.5 Functional Description 477
6.9.6 Register Map 480
6.9.7 Register Description 481
6.10 Wormhole Controller (WHC) 489
6.10.1 Overview 489
6.10.2 Features 489
6.10.3 Block Diagram 489
6.10.4 Basic Configuration 490
6.10.5 Functional Description 490
6.10.6 Register Map 491
6.10.7 Register Description 493
6.11 DDR Memory Controller (UMCTL2) 512
6.11.1 Overview 512
6.11.2 Features 512
6.11.3 Block Diagram 513
6.11.4 DDR Memory Controller Architecture Overview 515
6.11.5 Basic Configuration 519
6.11.6 Functional Description 519
6.11.7 Register Map 541
6.11.8 Register Description 545
6.12 DDR 3/2 PHY Controller (DDR32PHY) 697
6.12.1 Overview 697
6.12.2 Features 697
6.12.3 Block Diagram 698
6.12.4 DDR3/2 PHY SDRAM Interface 699
6.12.5 Basic Configuration 700
6.12.6 Functional Description 701
6.12.7 Register Map 707

6.12.8 Register Description 714
6.13 One Time Programming Memory Controller (OTP) 814
6.13.1 Overview 814
6.13.2 Features 814
6.13.3 Functional Description 814
6.14 External Bus Interface (EBI) 820
6.14.1 Overview 820
6.14.2 Features 820
6.14.3 Block Diagram 820
6.14.4 Basic Configuration 821
6.14.5 Functional Description 827
6.14.6 Register Map 837
6.14.7 Register Description 838
6.15 General Purpose I/O (GPIO) 842
6.15.1 Overview 842
6.15.2 Features 842
6.15.3 Block Diagram 842
6.15.4 Basic Configuration 843
6.15.5 Functional Description 844
6.15.6 Register Map 847
6.15.7 Register Description 855
6.16 PDMA Controller (PDMA) 881
6.16.1 Overview 881
6.16.2 Features 881
6.16.3 Block Diagram 881
6.16.4 Basic Configuration 882
6.16.5 Functional Description 883
6.16.6 Register Map 892
6.16.7 Register Description 894
6.17 Timer Controller (TMR) 932
6.17.1 Overview 932
6.17.2 Features 932
6.17.3 Block Diagram 933
6.17.4 Basic Configuration 938
6.17.5 Timer Functional Description 940
6.17.6 PWM Functional Description 945
6.17.7 Register Map 962
6.17.8 Register Description 976
6.18 EPWM Generator and Capture Timer (EPWM) 1030
6.18.1 Overview 1030
6.18.2 Features 1030

6.18.3 Block Diagram 1031
6.18.4 Basic Configuration 1035
6.18.5 Functional Description 1037
6.18.6 Register Map 1074
6.18.7 Register Description 1079
6.19 Enhanced Input Capture Timer (ECAP) 1151
6.19.1 Overview 1151
6.19.2 Features 1151
6.19.3 Block Diagram 1151
6.19.4 Basic Configuration 1152
6.19.5 Functional Description 1153
6.19.6 Register Map 1158
6.19.7 Register Description 1159
6.20 Quadrature Encoder Interface (QEI) 1169
6.20.1 Overview 1169
6.20.2 Features 1169
6.20.3 Block Diagram 1169
6.20.4 Basic Configuration 1170
6.20.5 Functional Description 1171
6.20.6 Register Map 1179
6.20.7 Register Description 1180
6.21 Watchdog Timer (WDT) 1191
6.21.1 Overview 1191
6.21.2 Features 1191
6.21.3 Block Diagram 1191
6.21.4 Basic Configuration 1193
6.21.5 Functional Description 1193
6.21.6 Register Map 1196
6.21.7 Register Description 1197
6.22 Window Watchdog Timer (WWDT) 1201
6.22.1 Overview 1201
6.22.2 Features 1201
6.22.3 Block Diagram 1201
6.22.4 Basic Configuration 1203
6.22.5 Functional Description 1203
6.22.6 Register Map 1207
6.22.7 Register Description 1208
6.23 Real Time Clock (RTC) 1213
6.23.1 Overview 1213
6.23.2 Features 1213
6.23.3 Block Diagram 1213

6.23.4 Basic Configuration 1214
6.23.5 Functional Description 1215
6.23.6 Register Map 1222
6.23.7 Register Description 1224
6.24 UART Interface Controller (UART) 1264
6.24.1 Overview 1264
6.24.2 Features 1264
6.24.3 Block Diagram 1265
6.24.4 Basic Configuration 1270
6.24.5 Functional Description 1281
6.24.6 Register Map 1298
6.24.7 Register Description 1300
6.25 Smart Card Host Interface (SC) 1329
6.25.1 Overview 1329
6.25.2 Features 1329
6.25.3 Block Diagram 1329
6.25.4 Basic Configuration 1331
6.25.5 Functional Description 1333
6.25.6 Register Map 1345
6.25.7 Register Description 1346
6.26 I^2C Serial Interface Controller (I^2C) 1371
6.26.1 Overview 1371
6.26.2 Features 1371
6.26.3 Block Diagram 1371
6.26.4 Basic Configuration 1372
6.26.5 Functional Description 1374
6.26.6 Register Map 1391
6.26.7 Register Description 1392
6.27 Serial Peripheral Interface (SPI) 1404
6.27.1 Overview 1404
6.27.2 Features 1404
6.27.3 Block Diagram 1404
6.27.4 Basic Configuration 1405
6.27.5 Functional Description 1408
6.27.6 Timing Diagram 1426
6.27.7 Programming Examples 1428
6.27.8 Register Map 1431
6.27.9 Register Description 1432
6.28 Quad Serial Peripheral Interface (QSPI) 1454
6.28.1 Overview 1454
6.28.2 Features 1454

6.28.3 Block Diagram 1454
6.28.4 Basic Configuration 1455
6.28.5 Functional Description 1457
6.28.6 Timing Diagram 1475
6.28.7 Programming Examples 1477
6.28.8 Register Map 1480
6.28.9 Register Description 1481
6.29 I^2S Controller (I^2S) 1498
6.29.1 Overview 1498
6.29.2 Features 1498
6.29.3 Block Diagram 1498
6.29.4 Basic Configuration 1498
6.29.5 Functional Description 1501
6.29.6 Register Map 1512
6.29.7 Register Description 1513
6.30 High Speed USB 2.0 Device Controller (HSUSBD) 1531
6.30.1 Overview 1531
6.30.2 Features 1531
6.30.3 Block Diagram 1531
6.30.4 Basic Configuration 1532
6.30.5 Functional Description 1532
6.30.6 Register Map 1534
6.30.7 Register Description 1538
6.31 USB 2.0 Host Controller (HSUSBH) 1749
6.31.1 Overview 1749
6.31.2 Features 1749
6.31.3 Block Diagram 1749
6.31.4 Basic Configuration 1750
6.31.5 Functional Description 1751
6.31.6 Register Map 1752
6.31.7 Register Description 1753
6.32 USB 1.1 Host Controller (USBH) 1770
6.32.1 Overview 1770
6.32.2 Features 1770
6.32.3 Block Diagram 1770
6.32.4 Basic Configuration 1771
6.32.5 Functional Description 1772
6.32.6 Register Map 1774
6.32.7 Register Description 1775
6.33 Controller Area Network with Feasibility Data Rate (CAN FD) 1805
6.33.1 Overview 1805

6.33.2 Features 1805
6.33.3 Block Diagram 1805
6.33.4 Basic Configuration 1807
6.33.5 Functional Description 1810
6.33.6 Register Map 1839
6.33.7 Register Description 1841
6.34 Gigabit Media Access Controller (GMAC) 1893
6.34.1 Overview 1893
6.34.2 Features 1893
6.34.3 Architecture 1894
6.34.4 Basic Configuration 1907
6.34.5 Functional Description 1909
6.34.6 Descriptors 1941
6.34.7 Register Map 1966
6.34.8 Register Description 1969
6.35 Secure Digital Host Controller (SDH) 2054
6.35.1 Overview 2054
6.35.2 Features 2054
6.35.3 Block Diagram 2054
6.35.4 Basic Configuration 2055
6.35.5 Functional Description 2056
6.35.6 Register Map 2066
6.35.7 Register Description 2068
6.36 NAND Flash Interface (NFI) 2151
6.36.1 Overview 2151
6.36.2 Features 2151
6.36.3 Block Diagram 2151
6.36.4 Basic Configuration 2152
6.36.5 Functional Description 2153
6.36.6 Register Map 2157
6.36.7 Register Description 2159
6.37 Cryptographic Accelerator (CRYPTO) 2208
6.37.1 Overview 2208
6.37.2 Features 2208
6.38 Key Store (KS) 2210
6.38.1 Overview 2210
6.38.2 Features 2210
6.39 Camera Capture Interface Controller (CCAP) 2211
6.39.1 Overview 2211
6.39.2 Features 2211
6.39.3 Block Diagram 2211

6.39.4 Basic Configuration 2212
6.39.5 Functional Description 2214
6.39.6 Register Map 2218
6.39.7 Register Description 2219
6.40 LCD Display Controller (DISP) 2245
6.40.1 Overview 2245
6.40.2 Features 2245
6.40.3 Block Diagram 2246
6.40.4 Basic Configuration 2247
6.40.5 Functional Description 2249
6.40.6 Register Map 2257
6.40.7 Register Description 2262
6.41 2D Graphic Engine (GFX) 2344
6.41.1 Overview 2344
6.41.2 Features 2344
6.41.3 Block Diagram 2345
6.41.4 Basic Configuration 2346
6.41.5 Functional Description 2347
6.41.6 Register Map 2356
6.41.7 Register Description 2357
6.42 H.264/JPEG Decoder (VC8000NanoD) 2376
6.42.1 Overview 2376
6.42.2 Features 2376
6.42.3 Block Diagram 2377
6.42.4 Functional Description 2377
6.42.5 Register Map 2380
6.42.6 Register Description 2380
6.43 Keypad Interface (KPI) 2381
6.43.1 Overview 2381
6.43.2 Features 2381
6.43.3 Block Diagram 2381
6.43.4 Basic Configuration 2381
6.43.5 Functional Description 2382
6.43.6 Register Map 2385
6.43.7 Register Description 2386
6.44 Analog to Digital Converter (ADC) 2399
6.44.1 Overview 2399
6.44.2 Features 2399
6.44.3 Block Diagram 2399
6.44.4 Basic Configuration 2400
6.44.5 Functional Description 2401

6.44.6 Register Map 2407
6.44.7 Register Description 2408
6.45 Enhanced 12-bit Analog-to-Digital Converter (EADC) 2426
6.45.1 Overview 2426
6.45.2 Features 2426
6.45.3 Block Diagram 2427
6.45.4 Basic Configuration 2427
6.45.5 Functional Description 2428
6.45.6 Register Map 2442
6.45.7 Register Description 2444
6.46 System Debug (CoreSight SoC-400) 2477
6.46.1 Overview 2477
6.46.2 Features 2477
6.46.3 Block Diagram 2477
6.46.4 Basic Configuration 2477
6.46.5 Functional Description 2478
6.47 Peripherals Interconnection 2482
6.47.1 Overview 2482
6.47.2 Peripherals Interconnect Matrix table 2482
6.47.3 Functional Description 2482
7 APPLICATION CIRCUIT 2485
8 ELECTRICAL CHARACTERISTICS 2486
9 PACKAGE DIMENSIONS 2487
9.1 LQFP-EP 216-Pin (24x24x1.4mm, footprint 2.0mm) 2487
9.2 BGA 312-Ball (15x15x1.4mm, 0.8mm pitch) 2488
9.3 BGA 364-Ball (14x14x1.4mm, 0.65mm pitch) 2489
9.4 Thermal Characteristics 2490
9.4.1 Thermal Performance of Package under Forced Convection 2490
9.4.2 Thermal Performance Terminology 2491
9.4.3 Simulation Conditions 2492
9.5 PCB Reflow Profile Suggestion 2493
9.5.1 Profile Setting Consideration 2493
9.5.2 Profile Suggestion 2495
9.5.3 PCB Assembly Considerations 2496
9.6 PKG Baking and Vacuumed 2497
10 ABBREVIATIONS 2498
10.1 Abbreviations 2498
11 REVISION HISTORY 2500

LIST OF FIGURES

Figure 3.2-1 MA35D1 Series Part Number Naming Guide 47
Figure 4.1-1 NuMicro MA35D1 Series LQFP-EP 216-Pin Pinout Diagram 56
Figure 4.1-2 NuMicro MA35D1 Series BGA 312-Ball Pinout Diagram 57
Figure 4.1-3 NuMicro MA35D1 Series BGA 364-Ball Pinout Diagram 58
Figure 5.1-1 MA35D1 Series Block Diagram 118
Figure 6.1-1 Arm Cortex-A35 Processor Block Diagram 120
Figure 6.2-1 Arm Cortex-M4 Core Block Diagram 134
Figure 6.3-1 TrustZone System Architecture Block Diagram 179
Figure 6.4-1 Power-on Reset (POR) Waveform 183
Figure 6.4-2 Low Voltage Reset (LVR) Waveform 183
Figure 6.4-3 nRESET Reset Waveform 184
Figure 6.4-4 MA35D1 Series Power Distribution Diagram 187
Figure 6.4-5 MA35D1 System Memory Map Diagram 189
Figure 6.5-1 Clock Generator Global View Diagram (1/3) 304
Figure 6.5-2 Clock Generator Global View Diagram (2/3) 305
Figure 6.5-3 Clock Generator Global View Diagram (3/3) 306
Figure 6.5-4 Clock Generator Block Diagram 308
Figure 6.5-5 CA35 CPU Clock Block Diagram 309
Figure 6.5-6 SYSCLK0 Block Diagram 309
Figure 6.5-7 SYSCLK1 Block Diagram 310
Figure 6.5-8 HXT Stop Protect Procedure 310
Figure 6.5-9 SysTick Clock Control Block Diagram 311
Figure 6.5-10 Advanced PLL Down Spreading in Spectrum Mode 313
Figure 6.5-11 Clock Output Block Diagram 316
Figure 6.6-1 SSPCC Block Diagram 370
Figure 6.6-2 System Partition Graphical View 371
Figure 6.6-3 SRAM Security Attribution Configuration 374
Figure 6.6-4 Security Violation Signal Block Diagram 378
Figure 6.6-5 DPM Architecture 380
Figure 6.6-6 Flow of Debug State 381
Figure 6.6-7 Password-based Debug Authentication Scheme 383
Figure 6.7-1 SSMCC Block Diagram 432
Figure 6.7-2 Interrupt Route of SSMCC 434
Figure 6.9-1 Semaphore Block Diagram 477
Figure 6.9-2 Synchronization between Cortex-A and Cortex-M Diagram 478
Figure 6.9-3 Synchronization between Cortex-A Cores in AMP System Diagram 478

Figure 6.10-1 Wormhole Controller Block Diagram 489
Figure 6.11-1 uMCTL2 (AXI Configuration) in System Level Diagram 514
Figure 6.11-2 uMCTL2 Block Diagram (AXI) 515
Figure 6.11-3 Example with Three Clock Domains 520
Figure 6.11-4 Software Handshake Protocol 522
Figure 6.11-5 Software Handshake Protocol (Continuous) 523
Figure 6.11-6 Simplified Relationship between Clocks and Resets for Register Initialization 525
Figure 6.11-7 Entry to Self-refresh through Hardware Low Power Interface of DDRC 532
Figure 6.11-8 Exit from Self-refresh through Hardware Low Power Interface of DDRC 532
Figure 6.11-9 Block Diagram of DDR Top Wrapper Module 533
Figure 6.11-10 Clock Removal Using Hardware Low Power Interface, Exit Triggered by Assertion of csysreq_n 534
Figure 6.11-11 Clock Removal Using Hardware Low Power Interface, Exit Triggered by Assertion of cactive_n 535
Figure 6.11-12 Rejection of Low Power State Request 535
Figure 6.12-1 DDR 3/2 PHY Block Diagram 699
Figure 6.12-2 PHY Initialization Flow 702
Figure 6.12-3 DFI Block Diagram 705
Figure 6.12-4 Initialization Block Diagram 706
Figure 6.14-1 EBI Block Diagram 821
Figure 6.14-2 Connection of 16-bit EBI Data Width with 16-bit Device 828
Figure 6.14-3 Connection of 8-bit EBI Data Width with 8-bit Device 828
Figure 6.14-4 Connection of 16-bit EBI Data Width with 16-bit Device in Separate mode 829
Figure 6.14-5 Connection of 8-bit EBI Data Width with 8-bit Device in Separate mode 829
Figure 6.14-6 Timing Control Waveform for 16-bit Data Width 831
Figure 6.14-7 Timing Control Waveform for 8-bit Data Width 832
Figure 6.14-8 Timing Control Waveform for Byte Write in 16-bit Data Mode 833
Figure 6.14-9 Timing Control Waveform for Insert Idle Cycle 834
Figure 6.14-10 Timing Control Waveform for 16-bit Data Width for Separate Mode 835
Figure 6.14-11 Timing Control Waveform for Continuous Data Access Mode 836
Figure 6.15-1 GPIO Block Diagram 843
Figure 6.15-2 Push-Pull Output 844
Figure 6.15-3 Open-Drain Output 845
Figure 6.15-4 GPIO Rising Edge Trigger Interrupt 845
Figure 6.15-5 GPIO Falling Edge Trigger Interrupt 846
Figure 6.16-1 PDMA Controller Block Diagram 882
Figure 6.16-2 Descriptor Table Entry Structure 883
Figure 6.16-3 Basic Mode Finite State Machine 884

Figure 6.16-4 Descriptor Table Link List Structure 885
Figure 6.16-5 Scatter-gather Mode Finite State Machine 886
Figure 6.16-6 Example of Single Transfer Type and Burst Transfer Type in Basic Mode 887
Figure 6.16-7 Example of PDMA Channel 0 Time-out Counter Operation 888
Figure 6.16-8 Stride Function Block Transfer 889
Figure 6.16-9 Block Transfer with Interval =3 and Repeat Count =2 890
Figure 6.17-1 Timer Controller Block Diagram 933
Figure 6.17-2 Clock Source of Timer Controller 934
Figure 6.17-3 PWM Generator Overview Block Diagram 935
Figure 6.17-4 PWM System Clock Source Control 936
Figure 6.17-5 PWM Counter Clock Source Control 936
Figure 6.17-6 PWM Independent Mode Architecture Diagram 937
Figure 6.17-7 PWM Complementary Mode Architecture Diagram 937
Figure 6.17-8 Continuous Counting Mode 942
Figure 6.17-9 External Capture Mode on TMx_EXT Pin 943
Figure 6.17-10 External Reset Counter Mode 943
Figure 6.17-11 Internal Timer Trigger 944
Figure 6.17-12 Inter-Timer Trigger Capture Timing 945
Figure 6.17-13 PWM Prescale Waveform in Up Count Type 945
Figure 6.17-14 PWM Up Count Type 946
Figure 6.17-15 PWM Down Count Type 946
Figure 6.17-16 PWM Up-Down Count Type 947
Figure 6.17-17 PWM Comparator Events in Up-Down Count Type 948
Figure 6.17-18 Period Loading Mode with Up Count Type 949
Figure 6.17-19 Immediately Loading Mode with Up Count Type 950
Figure 6.17-20 PWM Pulse Generation in Up-Down Count Type 950
Figure 6.17-21 PWM Pulse Generation in Up Count Type 951
Figure 6.17-22 PWM Pulse Generation in Down Count Type 951
Figure 6.17-23 PWM 0% to 100% Duty Cycle in Up Count Type and Up-Down Count Type 952
Figure 6.17-24 PWM Independent Mode Output Waveform 952
Figure 6.17-25 PWM Complementary Mode Output Waveform 953
Figure 6.17-26 PWMx_CH0 Output Control in Independent Mode 953
Figure 6.17-27 PWMx_CH0 and PWMx_CH1 Output Control in Complementary Mode 953
Figure 6.17-28 Dead-Time Insertion 954
Figure 6.17-29 PWM Output Mask Control Waveform 954
Figure 6.17-30 Brake Pin Noise Filter Block Diagram 955
Figure 6.17-31 Brake Event Block Diagram for PWMx_CH0 and PWMx_CH1 956

Figure 6.17-32 Edge Detector Brake Waveform for PWMx_CH0 and PWMx_CH1 957
Figure 6.17-33 Level Detector Brake Waveform for PWMx_CH0 and PWMx_CH1 958
Figure 6.17-34 Brake Source Block Diagram 959
Figure 6.17-35 System Fail Brake Block Diagram 959
Figure 6.17-36 PWMx_CH0 and PWMx_CH1 Polarity Control with Dead-Time Insertion 960
Figure 6.17-37 PWM Interrupt Architecture Diagram 961
Figure 6.17-38 PWM Trigger EADC Block Diagram 961
Figure 6.18-1 EPWM Generator Overview Block Diagram 1032
Figure 6.18-2 EPWM Clock Source Control 1033
Figure 6.18-3 EPWM Clock Source Control 1033
Figure 6.18-4 EPWM Independent Mode Architecture Diagram 1034
Figure 6.18-5 EPWM Complementary Mode Architecture Diagram 1035
Figure 6.18-6 EPWM_CH0 Prescaler Waveform in Up Counter Type 1037
Figure 6.18-7 EPWM Counter Waveform when Setting Clear Counter 1038
Figure 6.18-8 EPWM Up Counter Type 1038
Figure 6.18-9 EPWM Down Counter Type 1039
Figure 6.18-10 EPWM Up-Down Counter Type 1040
Figure 6.18-11 EPWM Compared point Events in Up-Down Counter Type 1041
Figure 6.18-12 EPWM Double Buffering Illustration 1042
Figure 6.18-13 Period Loading in Up-Count Mode 1043
Figure 6.18-14 Immediately Loading in Up-Count Mode 1044
Figure 6.18-15 Window Loading in Up-Count Mode 1045
Figure 6.18-16 Center Loading in Up-Down-Count Mode 1046
Figure 6.18-17 EPWM One-shot Mode Output Waveform 1047
Figure 6.18-18 EPWM Pulse Generation 1048
Figure 6.18-19 EPWM 0% to 100% Pulse Generation 1048
Figure 6.18-20 EPWM Independent Mode Waveform 1050
Figure 6.18-21 EPWM Complementary Mode Waveform 1050
Figure 6.18-22 EPWM Group Function Waveform 1051
Figure 6.18-23 EPWM SYNC_IN Noise Filter Block Diagram 1052
Figure 6.18-24 EPWM Counter Synchronous Function Block Diagram 1053
Figure 6.18-25 EPWM Synchronous Function with Synchronize Source from SYNC_IN Signal 1054
Figure 6.18-26 EPWMx_CH0 Output Control in Independent Mode 1054
Figure 6.18-27 EPWMx_CH0 and EPWMx_CH1 Output Control in Complementary Mode 1055
Figure 6.18-28 Dead-Time Insertion 1056
Figure 6.18-29 Illustration of Mask Control Waveform 1056
Figure 6.18-30 Brake Noise Filter Block Diagram 1057

Figure 6.18-31 Brake Block Diagram for EPWMx_CH0 and EPWMx_CH1 Pair 1058
Figure 6.18-32 Edge Detector Waveform for EPWMx_CH0 and EPWMx_CH1 Pair 1059
Figure 6.18-33 Level Detector Waveform for EPWMx_CH0 and EPWMx_CH1 Pair 1059
Figure 6.18-34 Brake Source Block Diagram 1060
Figure 6.18-35 Brake System Fail Block Diagram 1060
Figure 6.18-36 Initial State and Polarity Control with Rising Edge Dead-Time Insertion 1061
Figure 6.18-37 EPWMx_CH0 Accumulate Interrupt Waveform 1062
Figure 6.18-38 EPWMx_CH0 and EPWMx_CH1 Pair Interrupt Architecture Diagram 1063
Figure 6.18-39 Fault Detect Function Interrupt Architecture Diagram 1064
Figure 6.18-40 EPWMx_CH0 and EPWMx_CH1 Pair Trigger EADC Events 1065
Figure 6.18-41 EPWMx_CH0 Trigger EADC Block Diagram 1065
Figure 6.18-42 EPWM Trigger EADC in Up-Down Counter Type Timing Waveform 1066
Figure 6.18-43 EPWM_CH0 Capture Block Diagram 1067
Figure 6.18-44 Capture Operation Waveform 1068
Figure 6.18-45 Capture PDMA Operation Waveform of Channel 0 1069
Figure 6.18-46 Accumulator PDMA Function Architecture 1070
Figure 6.18-47 EPWM Accumulator Stop Mode Waveform 1071
Figure 6.18-48 Fault Detect Function Architecture 1072
Figure 6.18-49 EPWM Mask Function Waveform 1073
Figure 6.18-50 EPWM Deglitch Function Waveform 1073
Figure 6.19-1 Input Capture Timer/Counter Architecture 1151
Figure 6.19-2 Input Capture Timer/Counter Clock Source Control 1153
Figure 6.19-3 Noise Filter Sampling Clock Selection 1154
Figure 6.19-4 Input Capture Timer/Counter Function Block 1155
Figure 6.19-5 Input Capture Timer/Counter Interrupt Architecture Diagram 1157
Figure 6.20-1 QEI Block Diagram 1169
Figure 6.20-2 QEI Clock Source Control 1170
Figure 6.20-3 Noise Filter 1171
Figure 6.20-4 Noise Filter Sampling Clock Selection 1172
Figure 6.20-5 QEA/QEB/IDX Timing Requirement through Noise Filter 1172
Figure 6.20-6 X4 Counting Mode 1173
Figure 6.20-7 X2 Counting Mode 1174
Figure 6.20-8 Compare Operation 1175
Figure 6.20-9 QEI_CNT Reload/Reset Control 1176
Figure 6.20-10 Trigger Control of Capturing QEI Counter 1176
Figure 6.20-11 Capture and Latch QEI Counter 1177
Figure 6.20-12 Quadrature Encoder Interface Interrupt Architecture Diagram 1178

Figure 6.21-1 Watchdog Timer Time-out Interval and Reset Period Timing 1194
Figure 6.22-1 WWDT Clock Control 1202
Figure 6.22-2 WWDT Block Diagram 1202
Figure 6.22-3 WWDT Reset Control Block Diagram 1203
Figure 6.22-4 WWDT Reset and Reload Behavior 1204
Figure 6.22-5 WWDT Reload Counter When CNTDAT > CMPDAT 1205
Figure 6.22-6 WWDT Reload Counter When CNTDAT < CMPDAT 1205
Figure 6.22-7 WWDT Interrupt and Reset Signals 1206
Figure 6.23-1 RTC Block Diagram 1214
Figure 6.23-2 Dynamic Rate Definition 1219
Figure 6.23-3 Backup I/O Control Diagram 1221
Figure 6.24-1 UART Clock Control Diagram 1268
Figure 6.24-2 UART Block Diagram 1268
Figure 6.24-3 Auto-Baud Rate Measurement 1285
Figure 6.24-4 Transmit Delay Time Operation 1285
Figure 6.24-5 UART nCTS Wake-up Case1 1286
Figure 6.24-6 UART nCTS Wake-up Case2 1286
Figure 6.24-7 UART Data Wake-up 1287
Figure 6.24-8 UART Received Data FIFO Reached Threshold Wake-up 1287
Figure 6.24-9 UART RS-485 AAD Mode Address Match Wake-up 1288
Figure 6.24-10 UART Received Data FIFO Threshold Time-out Wake-up 1288
Figure 6.24-11 Auto-Flow Control Block Diagram 1291
Figure 6.24-12 UART nCTS Auto-Flow Control Enabled 1292
Figure 6.24-13 UART nRTS Auto-Flow Control Enabled 1292
Figure 6.24-14 UART nRTS Auto-Flow with Software Control 1293
Figure 6.24-15 IrDA Control Block Diagram 1293
Figure 6.24-16 IrDA TX/RX Timing Diagram 1294
Figure 6.24-17 RS-485 nRTS Driving Level in Auto Direction Mode 1295
Figure 6.24-18 RS-485 nRTS Driving Level with Software Control 1296
Figure 6.24-19 Structure of RS-485 Frame 1296
Figure 6.25-1 SC Clock Control Diagram (7-bit Pre-scale Counter in Clock Controller) 1330
Figure 6.25-2 SC Controller Block Diagram 1330
Figure 6.25-3 SC Data Character 1333
Figure 6.25-4 SC Activation Sequence 1335
Figure 6.25-5 SC Warm Reset Sequence 1336
Figure 6.25-6 SC Deactivation Sequence 1337
Figure 6.25-7 Basic Operation Flow 1338

Figure 6.25-8 Initial Character TS 1339
Figure 6.25-9 SC Error Signal 1340
Figure 6.25-10 Transmit Direction Block Guard Time Operation 1343
Figure 6.25-11 Receive Direction Block Guard Time Operation 1343
Figure 6.25-12 Extra Guard Time Operation 1343
Figure 6.26-1 I²C Controller Block Diagram 1372
Figure 6.26-2 I²C Bus Timing 1375
Figure 6.26-3 I²C Protocol 1375
Figure 6.26-4 START and STOP Conditions 1376
Figure 6.26-5 Bit Transfer on the I²C Bus 1376
Figure 6.26-6 Acknowledge on the I²C Bus 1377
Figure 6.26-7 Master Transmits Data to Slave by 7-bit 1377
Figure 6.26-8 Master Reads Data from Slave by 7-bit 1377
Figure 6.26-9 Master Transmits Data to Slave by 10-bit 1378
Figure 6.26-10 Master Reads Data from Slave by 10-bit 1378
Figure 6.26-11 Control I²C Bus according to the Current I²C Status 1379
Figure 6.26-12 Master Transmitter Mode Control Flow 1379
Figure 6.26-13 Master Receiver Mode Control Flow 1380
Figure 6.26-14 Slave Mode Control Flow 1381
Figure 6.26-15 GC Mode 1382
Figure 6.26-16 Arbitration Lost 1383
Figure 6.26-17 Setup Time Wrong Adjustment 1384
Figure 6.26-18 Hold Time Wrong Adjustment 1385
Figure 6.26-19 I²C Data Shifting Direction 1386
Figure 6.26-20 I²C Time-out Count Block Diagram 1387
Figure 6.26-21 I²C Wake-Up Related Signals Waveform 1388
Figure 6.26-22 EEPROM Random Read 1389
Figure 6.26-23 Protocol of EEPROM Random Read 1389
Figure 6.27-1 SPI Block Diagram 1405
Figure 6.27-2 SPI Peripheral Clock 1408
Figure 6.27-3 SPI Full-Duplex Master Mode Application Block Diagram 1409
Figure 6.27-4 SPI Full-Duplex Slave Mode Application Block Diagram 1409
Figure 6.27-5 32-bit in One Transaction 1410
Figure 6.27-6 Automatic Slave Selection (SSACTPOL = 0, SUSPITV > 0x2) 1411
Figure 6.27-7 Automatic Slave Selection (SSACTPOL = 0, SUSPITV < 0x3) 1411
Figure 6.27-8 Byte Reorder Function 1412
Figure 6.27-9 Timing Waveform for Byte Suspend 1412

Figure 6.27-10 SPI Half-Duplex Master Mode Application Block Diagram 1413
Figure 6.27-11 SPI Half-Duplex Slave Mode Application Block Diagram 1413
Figure 6.27-12 FIFO Threshold Comparator 1415
Figure 6.27-13 Transmit FIFO Buffer Example for 8~16 Bits of Data Length 1416
Figure 6.27-14 Transmit FIFO Buffer Example for 17~32 Bits of Data Length 1416
Figure 6.27-15 Receive FIFO Buffer Example for 16 Bits of Data Length 1417
Figure 6.27-16 Receive FIFO Buffer Example for 32 Bits of Data Length 1418
Figure 6.27-17 TX Underflow Event and Slave Under Run Event 1418
Figure 6.27-18 TX Underflow Event (SPI Slave 3-Wire Mode Enabled) 1419
Figure 6.27-19 Slave Mode Bit Count Error and Effective Bit Number of Uncompleted RX Data 1420
Figure 6.27-20 I²S Data Format Timing Diagram 1422
Figure 6.27-21 MSB Justified Data Format Timing Diagram 1422
Figure 6.27-22 PCM Mode A Timing Diagram 1422
Figure 6.27-23 PCM Mode B Timing Diagram 1423
Figure 6.27-24 FIFO Contents for Various I²S Modes 1424
Figure 6.27-25 SPI Timing in Master Mode 1427
Figure 6.27-26 SPI Timing in Master Mode (Alternate Phase of SPIx_CLK) 1427
Figure 6.27-27 SPI Timing in Slave Mode 1428
Figure 6.27-28 SPI Timing in Slave Mode (Alternate Phase of SPIx_CLK) 1428
Figure 6.28-1 QSPI Block Diagram 1455
Figure 6.28-2 QSPI Peripheral Clock 1457
Figure 6.28-3 QSPI Full-Duplex Master Mode Application Block Diagram 1457
Figure 6.28-4 QSPI Full-Duplex Slave Mode Application Block Diagram 1458
Figure 6.28-5 32-bit in One Transaction 1459
Figure 6.28-6 Automatic Slave Selection (SSACTPOL = 0, SUSPITV > 0x2) 1460
Figure 6.28-7 Automatic Slave Selection (SSACTPOL = 0, SUSPITV < 0x3) 1460
Figure 6.28-8 Byte Reorder Function 1461
Figure 6.28-9 Timing Waveform for Byte Suspend 1461
Figure 6.28-10 QSPI Half-Duplex Master Mode Application Block Diagram 1462
Figure 6.28-11 QSPI Half-Duplex Slave Mode Application Block Diagram 1462
Figure 6.28-12 Two-bit Transfer Mode System Architecture 1464
Figure 6.28-13 Two-bit Transfer Mode Timing (Master Mode) 1464
Figure 6.28-14 Bit Sequence of Dual Output Mode 1465
Figure 6.28-15 Bit Sequence of Dual Input Mode 1465
Figure 6.28-16 Bit Sequence of Quad Output Mode 1466
Figure 6.28-17 Bit Sequence of Quad Input Mode 1466
Figure 6.28-18 Timing Diagram of QSPI Output Data and Output Bus clock for Double Transfer

Rate Mode 1467
Figure 6.28-19 Timing Diagram of QSPI Input Data and Output Bus clock for Double Transfer Rate Mode 1468
Figure 6.28-20 FIFO Threshold Comparator 1469
Figure 6.28-21 Transmit FIFO Buffer Example 1470
Figure 6.28-22 Receive FIFO Buffer Example 1471
Figure 6.28-23 TX Underflow Event and Slave Under Run Event 1471
Figure 6.28-24 Two-bit Transfer Mode FIFO Buffer Example 1472
Figure 6.28-25 TX Underflow Event (QSPI0 Slave 3-Wire Mode Enabled) 1472
Figure 6.28-26 Slave Mode Bit Count Error and Effective Bit Number of Uncompleted RX Data 1473
Figure 6.28-27 Slave Time-out Event 1474
Figure 6.28-28 QSPI Timing in Master Mode 1476
Figure 6.28-29 QSPI Timing in Master Mode (Alternate Phase of QSPIx_CLK) 1476
Figure 6.28-30 QSPI Timing in Slave Mode 1477
Figure 6.28-31 QSPI Timing in Slave Mode (Alternate Phase of QSPIx_CLK) 1477
Figure 6.29-1 I²S Controller Block Diagram 1498
Figure 6.29-2 I²S Clock Control Diagram 1501
Figure 6.29-3 Master Mode Interface Block Diagram 1502
Figure 6.29-4 Slave Mode Interface Block Diagram 1502
Figure 6.29-5 I²S Channel Width and Data Width (CHWIDTH≦DATWIDTH) 1503
Figure 6.29-6 I²S Channel Width and Data Width (CHWIDTH > DATWIDTH) 1503
Figure 6.29-7 I²S Data Format Timing Diagram (FORMAT = 0x0; CHWIDTH≦DATWIDTH) .. 1503
Figure 6.29-8 MSB Justified Data Format (FORMAT = 0x1; CHWIDTH > DATWIDTH) 1504
Figure 6.29-9 LSB Justified Data Format (FORMAT = 0x2; CHWIDTH > DATWIDTH) 1504
Figure 6.29-10 Standard PCM Audio Timing Diagram (FORMAT = 0x4; CHWIDTH≦DATWIDTH) 1504
Figure 6.29-11 PCM with MSB Justified Data Format (FORMAT = 0x5; CHWIDTH > DATWIDTH) 1505
Figure 6.29-12 PCM with LSB Justified Data Format (FORMAT = 0x6; CHWIDTH > DATWIDTH) 1505
Figure 6.29-13 TDM 6-channel Audio Format with 24-bit Data in 32-bit Channel Block (PCM Standard Data Format; FORMAT=0x4) 1506
Figure 6.29-14 TDM 6-channel Audio Format with 24-bit Data in 32-bit Channel Block (PCM with MSB Justified; FORMAT=0x5) 1506
Figure 6.29-15 TDM 6-channel Audio Format with 24-bit Data in 32-bit Channel Block (PCM with LSB Justified; FORMAT=0x6) 1506
Figure 6.29-16 I²S Interrupts 1507
Figure 6.29-17 FIFO Contents for Various 2-channel Audio Modes 1508
Figure 6.29-18 FIFO Contents for Various 4-channel Audio Modes 1509

Figure 6.29-19 FIFO Contents for Various 6-channel Audio Modes (Part-1) 1510
Figure 6.29-20 FIFO Contents for Various 6-channel Audio Modes (Part-2) 1511
Figure 6.30-1 USB 2.0 HS Device Controller Block Diagram 1532
Figure 6.31-1 USB 2.0 HS Host Controller Block Diagram 1750
Figure 6.32-1 USB 1.1 Host Controller Block Diagram 1771
Figure 6.33-1 CAN FD Controller Block Diagram 1807
Figure 6.33-2 Transceiver Delay Measurement 1814
Figure 6.33-3 Pin Controller in Bus Monitor Mode 1815
Figure 6.33-4 Pin Control in Loop Back Modes 1817
Figure 6.33-5 Standard Message ID Filter Path 1820
Figure 6.33-6 Extended Message ID Filter Path 1821
Figure 6.33-7 Rx FIFO Status 1822
Figure 6.33-8 Rx FIFO Overflow Handling 1823
Figure 6.33-9 Example of Mixed Configuration Dedicated Tx Buffers / Tx FIFO 1827
Figure 6.33-10 Example of Mixed Configuration Dedicated Tx Buffers / Tx Queue 1827
Figure 6.33-11 Message RAM Configuration 1830
Figure 6.33-12 Bit Timing 1837
Figure 6.34-1 GMAC Block Diagram 1895
Figure 6.34-2 Descriptor Ring and Chain Structure 1896
Figure 6.34-3 Tx DMA Operation by default Mode 1900
Figure 6.34-4 Tx DMA Operation in OSF Mode 1902
Figure 6.34-5 Receive DMA Operation 1905
Figure 6.34-6 Networked Time Synchronization 1910
Figure 6.34-7 System Time Update Using Fine Method 1912
Figure 6.34-8 Propagation Delay Calculation in Clocks Supporting Peer-to-Peer Path Correction 1916
Figure 6.34-9 LPI Transitions (Transmit) 1925
Figure 6.34-10 LPI Transitions (Receive) 1926
Figure 6.34-11 RMII Block Diagram 1936
Figure 6.34-12 Transmission Bit Ordering 1936
Figure 6.34-13 Receive Bit Ordering 1937
Figure 6.34-14 RGMII Block Diagram 1937
Figure 6.34-15 RGMII Transmit Timing 1940
Figure 6.34-16 RGMII Receive Timing 1941
Figure 6.34-17 MAC Interrupt Masking Scheme 1941
Figure 6.34-18 Rx/Tx Descriptors in Same-Endian Mode for 64-Bit, Little-Endian Format Data Bus 1942
Figure 6.34-19 Transmit Descriptor Format in Little-Endian Mode With a 32-Bit Data Bus 1943

Figure 6.34-20 Receive Descriptor Format in Little-Endian Mode with a 32-Bit Data Bus 1947
Figure 6.34-21 Descriptor Fields when DMA Clears the Own Bit 1951
Figure 6.34-22 Descriptor Fields when DMA Clears the Own Bit 1952
Figure 6.34-23 Receive Timestamp Fields 1953
Figure 6.34-24 Transmit Descriptor Fields - Alternate (Enhanced) Format 1955
Figure 6.34-25 Transmit Descriptor Fetch (Read) for Alternate (Enhanced) Format 1955
Figure 6.34-26 Receive Descriptor Fields - Alternate (Enhanced) Format 1960
Figure 6.35-1 SD Host Controller Block Diagram 2055
Figure 6.35-2 ADMA2 Block Diagram 2057
Figure 6.35-3 ADMA2 Descriptor Table 2058
Figure 6.35-4 Tuning 2059
Figure 6.35-5 Pass Step and Fail Step Scenarios 2060
Figure 6.35-6 Sampling of Data In PFP 2061
Figure 6.35-7 Threshold Based Selection Tuning Schema 2061
Figure 6.35-8 Large Sampling Window Tuning Schema 2062
Figure 6.35-9 Auto-Tuning Top Level Block Diagram 2062
Figure 6.35-10 Auto-Tuning Interface Timing 2063
Figure 6.35-11 Glitch-Free Phase Switching 2063
Figure 6.35-12 Signal Voltage Switch Sequence 2064
Figure 6.36-1 NFI Block Diagram 2151
Figure 6.36-2 Data Arrangement for 2 KB Page Size NAND Flash 2154
Figure 6.36-3 Data Arrangement for 4 KB Page Size NAND Flash 2154
Figure 6.36-4 Data Arrangement for 8 KB Page Size NAND Flash 2155
Figure 6.36-5 NAND Flash Read Path Timing Diagram (EDO Mode) 2155
Figure 6.36-6 Timing Controlled by NFI_NANDTMCTL Register 2172
Figure 6.36-7 NAND Flash Read Path Timing Diagram (Non-EDO mode) 2172
Figure 6.36-8 NAND Flash Read Path Timing Diagram (EDO Mode) 2172
Figure 6.39-1 Camera Capture Interface Controller Functional Block Diagram 2211
Figure 6.40-1 Display Controller Block Diagram 2247
Figure 6.40-2 Timing Diagram 2249
Figure 6.40-3 Cursor 2250
Figure 6.40-4 Examples of Various Cursor Position in Relation to a Display Window 2251
Figure 6.41-1 GC520L Block Diagram 2346
Figure 6.41-2 Line 2347
Figure 6.41-3 Bit Blit 2348
Figure 6.41-4 Alpha Blender for One Pixel 2353
Figure 6.42-1 VC8000NanoD Decoder Diagram 2377

Figure 6.43-1 Keypad Controller Block Diagram 2381
Figure 6.43-2 Keypad Connection 2383
Figure 6.43-3 Keypad Interface 2383
Figure 6.43-4 8 x8 Keypad Scan Timing Diagram 2384
Figure 6.44-1 ADC Functional Block Diagram 2400
Figure 6.44-2 ADC Transfer Function 2401
Figure 6.44-3 Simplified Sampling Diagram 2402
Figure 6.44-4 ADC Timing Diagram 2404
Figure 6.44-54-wire Touch Screen Connection Diagram 2405
Figure 6.44-65-wire Touch Screen Connection Diagram 2405
Figure 6.45-1 EADC Converter Block Diagram 2427
Figure 6.45-2 Sample Module 0~3 Block Diagram 2429
Figure 6.45-3 Sample Module 4~7 Block Diagram 2430
Figure 6.45-4 Sample Module 8 Block Diagram 2430
Figure 6.45-5 EADC Clock Control 2431
Figure 6.45-6 Example EADC Conversion Timing Diagram, n=0~8 2432
Figure 6.45-7 Sample Module Conversion Priority Arbitrator Diagram 2433
Figure 6.45-8 Specific Sample Module EADC EOC Signal for ADINT0~3 Interrupt 2435
Figure 6.45-9 EADC0_ST De-bounce Timing Diagram 2436
Figure 6.45-10 EPWM-triggered EADC Start Conversion 2436
Figure 6.45-11 External triggered EADC Start Conversion 2436
Figure 6.45-12 Conversion Start Delay Timing Diagram 2437
Figure 6.45-13 EADC Extend Sampling Timing Diagram 2438
Figure 6.45-14 EADC Conversion Result Monitor Logics Diagram 2439
Figure 6.45-15 EADC Controller Interrupts 2440
Figure 6.45-16 EADC Start up Sequence with Calibration 2441
Figure 6.46-1 CoreSight Block Diagram 2477
Figure 7-1 Application Circuit 2485
Figure 9.1-1 LQFP-EP 216-Pin Package Demension 2487
Figure 9.2-1 BGA 312-Ball Package Demension 2488
Figure 9.3-1 BGA 364-Ball Package Demension 2489
Figure 9.4-1 Junction to Ambient Thermal Resistance 2491
Figure 9.4-2 Junction to Case Thermal Resistance 2491
Figure 9.5-1 PCB Reflow Profile Diagram 2493
Figure 9.5-2 Profile Suggestion for MA35D1 Series 2495
Figure 9.6-1 Cautions for PKG Baking 2497

LIST OF TABLES

Table 3.1-1 MA35D1 Series Package Types (New) 46
Table 3.1-2 MA35D1 Series Package Types 46
Table 6.1-1 GIC Interrupt Number Table 128
Table 6.2-1 Exception Model 141
Table 6.2-2 Interrupt Number Table 144
Table 6.2-3 Priority Grouping 173
Table 6.3-1 Cortex-A35 AArch64 Exception Levels vs. Security States and Typical Function of Each State 178
Table 6.4-1 Address Space Assignments for On-Chip Controllers 193
Table 6.4-2 Power-On Setting Bit Description 195
Table 6.4-3 System Manager Control Registers Access Attribute 198
Table 6.5-1 Advanced PLL Settings Example 315
Table 6.5-2 Clock Controller Registers Access Attribute 317
Table 6.6-1 Bits for Security Attribution Configuration 371
Table 6.6-2 List of Security Attribution of Peripherals 374
Table 6.6-3 Security Attribution of GPIO Register by Port Basis 374
Table 6.6-4 SSPCC Special Registers 375
Table 6.6-5 Security Memory Access Policy (Master is in TrustZone Domain) 376
Table 6.6-6 Security Memory Access Policy (Master is in SubM Domain) 377
Table 6.6-7 Master ID Number List 379
Table 6.6-8 Encoding of Debug State 381
Table 6.6-9 List of Debug Functions and Corresponding Register 382
Table 6.6-10 List of Classic Debug Functions 383
Table 6.6-11 DPM Bit Description in OTP 384
Table 6.6-12 List of Functions of Each PLM Stage 385
Table 6.6-13 Lifecycle OTP Bits 386
Table 6.7-1 TZC400 Filter Channels and Corresponding Masters 433
Table 6.7-2 Security Attribution of DRAM 435
Table 6.7-3 Example of SSMCC Region Configuration 435
Table 6.7-4 Security Configuration Registers of TZC400 436
Table 6.11-1 DWC_ddr_umctl2 and Memory Initialization with PUB 527
Table 6.11-2 Software Clock Removal Sequence 536
Table 6.11-3 Re-enabling the Clocks 536
Table 6.11-4 Logical to Physical Address Mapping (Examples for DDR3) 538
Table 6.12-1 Compatible Standards 698
Table 6.12-2 DDR3/2 PHY I/O list 700
Table 6.13-1 OTP Block Address Range and Access Path 815

Table 6.16-1 Channel Priority Table 884
Table 6.17-1 TIMER01 Pin Configuration 938
Table 6.17-2 TIMER23 Pin Configuration 939
Table 6.17-3 PWM Pulse Generation Event Priority in Up Count Type 951
Table 6.17-4 PWM Pulse Generation Event Priority in Down Count Type 952
Table 6.17-5 PWM Pulse Generation Event Priority in Up-Down Count Type 952
Table 6.18-1 EPWM Pulse Generation Event Priority for Up-Counter 1049
Table 6.18-2 EPWM Pulse Generation Event Priority for Down-Counter 1049
Table 6.18-3 EPWM Pulse Generation Event Priority for Up-Down-Counter 1049
Table 6.19-1 Typical Case of Noise Filter Settings 1154
Table 6.20-1 Direction of Counting 1175
Table 6.21-1 Watchdog Timer Time-out Interval Period Selection 1194
Table 6.22-1 WWDT Prescaler Value Selection 1204
Table 6.22-2 CMPDAT Setting Limitation 1206
Table 6.23-1 RTC Read/Write Enable 1216
Table 6.23-2 12/24 Hour Time Scale Selection 1217
Table 6.23-3 Registers Value after Powered On 1218
Table 6.23-4 Dynamic Pattern Source Selection 1219
Table 6.23-5 Tamper Control Bit Effection for Pair 0 1220
Table 6.24-1 NuMicro MA35D1 Series UART Features 1265
Table 6.24-2 UART Interrupt 1270
Table 6.24-3 UART Interface Controller Pin 1281
Table 6.24-4 UART controller Baud Rate Equation Table 1282
Table 6.24-5 UART controller Baud Rate Parameter Setting Example Table 1282
Table 6.24-6 UART controller Baud Rate Register Setting Example Table 1283
Table 6.24-7 Baud Rate Compensation Example Table 1 1283
Table 6.24-8 Baud Rate Compensation Example Table 2 1284
Table 6.24-9 UART controller Interrupt Source and Flag List 1290
Table 6.24-10 UART Line Control of Word and Stop Length Setting 1290
Table 6.24-11 UART Line Control of Parity Bit Setting 1291
Table 6.25-1 SC Host Controller Pin Description 1331
Table 6.25-2 UART Pin Description 1331
Table 6.25-3 Timer0/Timer1/Timer2 Operation Mode 1343
Table 6.26-1 Relationship between I^2C Baud Rate and PCLK 1384
Table 6.26-2 I^2C Status Code Description 1387
Table 6.27-1 SPI/I^2S Interface Controller Pin Description (SPI0~SPI3) 1408
Table 6.27-2 Dummy Data Number for I^2S / PCM Master Mode and Monaural Mode 1425

Table 6.27-3 Dummy Data Number for I²S / PCM Master Mode and Stereo Mode 1425
Table 6.27-4 Dummy Data Number for I²S Slave Mode and Monaural Mode 1425
Table 6.27-5 Dummy Data Number for PCM Slave Mode and Monaural Mode 1426
Table 6.27-6 Dummy Data Number for I²S Slave Mode and Stereo Mode 1426
Table 6.27-7 Dummy Data Number for PCM Slave Mode and Stereo Mode 1426
Table 6.33-1 DLC Coding in CAN FD Format 1812
Table 6.33-2 Rx Buffer / FIFO Element Size 1823
Table 6.33-3 Example Filter Configuration for Rx Buffers 1824
Table 6.33-4 Possible Configuration for Frame Transmission 1825
Table 6.33-5 Tx Buffer / FIFO / Queue Element Size 1826
Table 6.33-6 Rx Buffer and FIFO Element 1830
Table 6.33-7 Rx Buffer and FIFO Element Description 1832
Table 6.33-8 Tx Buffer and FIFO Element 1832
Table 6.33-9 Tx Buffer Element Description 1833
Table 6.33-10 Tx Event FIFO Element 1834
Table 6.33-11 Tx Event FIFO Element Description 1834
Table 6.33-12 Standard Message ID Filter Element 1835
Table 6.33-13 Standard Message ID Filter Element Description 1836
Table 6.33-14 Extended Message ID Filter Element 1836
Table 6.33-15 Extended Message ID Filter Element Description 1837
Table 6.34-1 Minimum PTP Clock Frequency Example 1915
Table 6.34-2 Ordinary Clock: PTP Messages for Snapshot 1917
Table 6.34-3 End to End Transparent Clock: PTP Messages for Snapshot 1918
Table 6.34-4 Peer-to-Peer Transparent Clock: PTP Messages for Snapshot 1918
Table 6.34-5 Message Format Defined in IEEE 1588-2008 1919
Table 6.34-6 IPv4-UDP PTP Frame Fields Required for Control and Status 1920
Table 6.34-7 IPv6-UDP PTP Frame Fields Required for Control and Status 1920
Table 6.34-8 Ethernet PTP Frame Fields Required for Control And Status 1921
Table 6.34-9 CRC Replacement Based on TDES0 Bit 27 and Bit 24 1928
Table 6.34-10 Transmit Checksum Offload Engine Functions for Different Packet Types 1931
Table 6.34-11 Receive Checksum Offload Engine (Type 2) Functions for Different Packet Types 1932
Table 6.34-12 MDIO Frame Structure 1934
Table 6.34-13 Encoding of gmii_txen, gmii_txer, rgmii_tctl_o, and rgmii_txd_o Signals 1938
Table 6.34-14 Decoding of rgmii_rctl_i and rgmii_rxd_i 1940
Table 6.34-15 Transmit Descriptor 0 1945
Table 6.34-16 Transmit Descriptor 1 1946
Table 6.34-17 Transmit Descriptor 2 1946

Table 6.34-18 Transmit Descriptor 3 1946
Table 6.34-19 Receive Descriptor 0 1949
Table 6.34-20 Receive Descriptor 0 when COE (Type 2) is Enabled..... 1949
Table 6.34-21 Receive Descriptor 1 1950
Table 6.34-22 Receive Descriptor 2 1950
Table 6.34-23 Receive Descriptor 3 1951
Table 6.34-24 Transmit Timestamp Status – Normal Descriptor Format Case (TDES0)..... 1952
Table 6.34-25 Transmit Timestamp Control – Normal Descriptor Format Case (TDES1) 1952
Table 6.34-26 Transmit Descriptor Fields (TDES2)..... 1953
Table 6.34-27 Transmit Descriptor Fields (TDES3)..... 1953
Table 6.34-28 Receive Descriptor Fields (RDES2) 1953
Table 6.34-29 Receive Descriptor Fields (RDES3) 1954
Table 6.34-30 Transmit Descriptor Word 0 (TDES0)..... 1958
Table 6.34-31 Transmit Descriptor Word 1 (TDES1)..... 1958
Table 6.34-32 Transmit Descriptor 2 (TDES2) 1959
Table 6.34-33 Transmit Descriptor 3 (TDES3) 1959
Table 6.34-34 Transmit Descriptor 6 (TDES6) 1959
Table 6.34-35 Transmit Descriptor 7 (TDES7) 1959
Table 6.34-36 Receive Descriptor Fields 0 (RDES0) 1962
Table 6.34-37 Receive Descriptor Fields 1 (RDES1) 1962
Table 6.34-38 Receive Descriptor Fields 2 (RDES2) 1963
Table 6.34-39 Receive Descriptor Fields 3 (RDES3) 1963
Table 6.34-40 Receive Descriptor Fields 4 (RDES4) 1965
Table 6.34-41 Receive Descriptor Fields 6 (RDES6) 1965
Table 6.34-42 Receive Descriptor Fields 7 (RDES7) 1965
Table 6.35-1 SD0 Pin Configuration 2056
Table 6.35-2 SD1 Pin Configuration 2056
Table 6.35-3 The Format of Descriptor..... 2058
Table 6.36-1 NAND Flash Pin Configuration 2152
Table 6.36-2 Number of Parity (Byte) for Each BCH Algorithm..... 2154
Table 6.40-1 Expected Input Organization for RGB Data..... 2252
Table 6.40-2 ColorKey Organization..... 2252
Table 6.40-3 Display Input Formats..... 2254
Table 6.40-4 Alignment and Base Addressing..... 2254
Table 6.40-5 Blending Modes Fractions Description 2256
Table 6.41-1 Supported Data Formats 2350
Table 6.41-2 YUV Output Support for Packed or Planar YUV Source per 2D Operation 2350

Table 6.41-3 Pre-multiplied Modes 2351
Table 6.41-4 Blending Modes Fractions Description 2354
Table 6.42-1 AVC(H.264) / MVC / SVC Decoding Features 2378
Table 6.42-2 JPEG Decoding Features 2378
Table 6.42-3 Post-Processing Features 2380
Table 6.44-1 Relation between Reference Voltage and Reference Selection 2403
Table 6.45-1 Relation between Resolution and Conversion Cycles 2433
Table 6.45-2 EADC Differential Model Channel Selection 2439
Table 6.45-3 EADC Power Saving Mode 2440
Table 6.45-4 EADC Start-up with Calibration 2441
Table 6.46-1 SWD/JTAG Pin Table 2478
Table 6.46-2 Trace Pin Table 2478
Table 6.46-3 CoreSight ROM Table 2479
Table 6.46-4 Cortex-A35 ROM Table 2479
Table 6.46-5 Cortex-M4 ROM Table 2481
Table 6.47-1 Peripherals Interconnect Matrix table 2482
Table 9.4-1 Thermal Performance of Package 2490
Table 9.4-2 Thermal Characteristics Simulation Conditions 2492
Table 9.5-1 PCB Reflow Profile Parameters 2494

1 GENERAL DESCRIPTION

The NuMicro MA35D1 series is a heterogeneous multi-core microprocessor targeted to high-end edge IIoT gateway. It is based on dual 64-bit Arm Cortex-A35 cores with speed up to 800 MHz, and one 180 MHz Arm Cortex-M4 core. Based on the high-performance cores, the MA35D1 series facilitates the tiny AI/ML for edge computing.

The MA35D1 series has a built-in 128 KB IBR (Internal Boot ROM) and supports secure booting from USB, SD/eMMC, NAND, and SPI Flash (SPI NOR/SPI NAND) modes. The MA35D1 series supports 16-bit DDR2 and DDR3/DDR3L SDRAM. For an easy system design and manufacture, the MA35D1 series also offers LQFP and BGA packages stacked with the DDR2/DDR3L SDRAM and density up to 512 MB to significantly reduce PCB layer, size and electromagnetic interference (EMI).

The MA35D1 series is a trusted system for IoT products' security requirements. It includes several advanced security mechanisms such as Nuvoton Trusted Secure Island (TSI), an isolated secure hardware unit, TrustZone®, secure boot, tamper-detection, built-in cryptographic accelerators with AES, SHA, ECC, RSA, SM2/3/4, and a TRNG, as well as Key Store and OTP memory. All the security operations are performed in the TSI to protect sensitive and high-value data. The features also satisfy customers in IEC 62443 certification requirements.

For high-end edge IIoT gateway requirements, the MA35D1 series integrates PDMA to increase the system performance. The PDMA can access system memory directly without the intervention of the CPU. Also, the MA35D1 series provides multiple advanced and high-speed connection interfaces, such as Gigabit Ethernet, SDIO3.0, USB 2.0 HS, and CAN FD, for edge gateway and new energy applications.

For HMI applications, the MA35D1 series provides a LCD display controller with the resolution up to 1920x1080 at 60 FPS, a 2D graphic engine, a JPEG and a H.264 decoder integrated for better graphical HMI effects and video playback.

The MA35D1 series has highly integrated functions and it is grouped into five sub-systems, Core sub-system, Memory sub-system, Security sub-system, Display, Video sub-system, Connectivity, and I/O sub-system.

1.1 Core Sub-system

The MA35D1 series is equipped with dual 64/32-bit Arm Cortex-A35 cores for Armv8-A architecture running up to 800 MHz for high-performance processing data in edge, 32/32 KB I/D L1 cache for each core, and a 512 KB shared L2 cache with Snoop Control Unit (SCU)-L2 cache protection, with Arm TrustZone, Arm Cryptographic Extension Arm NEON™ SIMD Engine and FPU (Floating Point Unit) supported.

The real-time processor Arm Cortex-M4 core runs up to 180 MHz for industrial control in real-time. It includes 16/16 KB I/D cache, FPU (Floating Point Unit), MPU (Memory Protect Unit), and a dedicated 128 KB SRAM for program code execution and data access. Besides, it can extend access to the system SDRAM up to 3.875 MB.

In addition to giving data processing and real-time control in edge, the computing power also facilities the basic AI/ML applications.

The MA35D1 series also provides the WWDT (Window Watchdog Timer), RTC (Real Time Clock), hardware semaphore and Wormhole for Cortex-A35 and Cortex-M4 communication, and most peripherals can be allocated to either the Cortex-A35 or Cortex-M4 cores to meet wide application requirements.

1.2 Memory Sub-system

The MA35D1 series has a built-in 128 KB mask ROM and supports secure booting from USB, SD/eMMC, NAND, and SPI Flash (SPI NOR/SPI NAND).

There are 384 (128 + 256) KB on-chip SRAM and 16-bit DDR2/DDR3/DDR3L SDRAM interface running up to 533 MHz, and up to 2 GB of SDRAM.

The MA35D1 series also supports stacking a DDR2/DDR3L SDRAM device into LQFP and BGA package to ease PCB design with lowered EMI and reduced BOM cost.

1.3 Security Sub-system

The MA35D1 series is a trusted system for IoT products. There are advanced security features to prevent sensitive and high-value data from being eavesdropped on and tampered. MA35D1 series can easily establish encrypted communications, secure data storage, and a secure environment for applications.

Execution Security

The Cortex-A35 is an ARMv8-A architecture with the TrustZone technology, preventing non-secure programs from accessing secure hardware functions to provide a secure environment for applications.

The MA35D1 series also supports secure boot to ensure system image's authenticity and integrity, and avoid executing the malware or unauthorized software implanted on the system.

The MA35D1 series is equipped with a TSI (Trusted Secure Island), which is an isolated secure hardware unit and its operation is not affected the MA35D1 system. All the cryptographic accelerators, Key Store, and OTP memory are built in the TSI, which perform all the security operations, including secure boot and tamper pins detection. The TSI interacts with the MA35D1 through a secure channel that effectively reduces the interfaces of malware attacks and preventing confidential data from being eavesdropped on and tampered with. The TSI also offers RTIC (Run-Time Integrity Checker) mechanism. It periodically checks the integrity of code or data sections in the background without interfering with the MA35D1's normal operation, preventing a Trojan system from being inserted or tampered with.

Communication Security

The MA35D1 series is equipped with several hardware cryptographic accelerators such as AES, SHA, ECC, RSA, SM2/3/4, and TRNG (True Random Number Generator), which facilitates exchanging secret keys, data encryption, and decryption between communications, and reduces the processor load.

Chip-level Storage Security

The MA35D1 series provides Key Store for key management and write-protected OTP (One-Time Programmable) memory for secret key or sensitive data storage to prevent tamper. Key Store and OTP memory can be accessed by the cryptographic engines, retrieving secret keys without the need of CPU access to reduce further risk of leakage of the sensitive cryptographic keys.

Physical Security

There is one pair of dynamic loop tamper pins or two individual tamper pins for tamper detection. Binding the tamper pins with PCB, the data and keys are destroyed if the system is threatened.

1.4 Display and Video Sub-system

For HMI applications, the MA35D1 series supports TFT-LCD display in a parallel RGB interface, the resolution up to 1920x1080 at 60 fps, and the display controller also supports the hardware cursor and OSD (On Screen Display). It also integrates a 2D graphic engine for graphics acceleration, such as BitBlt, rotation, multi-source alpha blending, and color format converting. A hardware JPEG decoder can decode compressed image size from 48x48 to 16368x16368. Another hardware H.264 decoder can decode compressed video stream, which supports resolution up to 1920x1080 at 45 FPS.

There are two sets of camera inputs supporting the video and image input for the image and video capture with machine learning applications.

1.5 Connectivity and I/O Sub-system

For high-end industrial control and edge gateway requirements, the MA35D1 series integrates four sets of PDMA to increase the system performance for peripherals to direct access the system memory without the intervention of the CPU. Also, it provides multiple advanced and high-speed connection

interfaces, such as Gigabit Ethernet, SDIO3.0, USB 2.0 HS, and CAN FD.

The MA35D1 series is suitable for a wide range of applications such as:

- Edge Gateway
  - Smart Factory
  - Smart City
  - Smart Building
  - Smart Retail
  - Smart Agriculture
- Tiny AI (Artificial Intelligent) / ML (Machine Learning)
  - Audio Sensing
  - Image Classification
  - Intelligent Data Security
- HMI (Human Machine Interface) & Industrial Control
  - Factory Automation
  - Industrial HMI
  - Smart Home
  - Smart Appliance
- New Energy Applications
  - Smart Grid
  - Smart Power
  - Energy System

2 FEATURES

2.1 MA35D1 Series Features

Core Sub-system	
Arm Cortex-A35	• Dual 64/32-bit Arm Cortex-A35 core running up to 800 MHz • Built-in 32 Kbytes instruction and 32 Kbytes data L1 cache for each core with Memory Management Unit (MMU) • Built-in 512 Kbytes shared L2 cache with Snoop Control Unit (SCU)-L2 cache protection • Arm TrustZone • Arm NEON SIMD Engine and FPU • Arm Cryptographic Extension • Armv8 debug logic • Supports Generic Interrupt Controller (GIC) CPU interface • Supports 64-bit count input for Generic Timer
Arm Cortex-M4	• 32-bit Arm Cortex-M4 processor core running up to 180 MHz • Built-in 16 Kbytes instruction and 16 Kbytes data cache • Built-in Memory Protection Unit (MPU) • Built-in Nested Vectored Interrupt Controller (NVIC) • Hardware IEEE 754 compliant Floating-point Unit (FPU) • DSP extension with hardware divider and single-cycle 32-bit hardware multiplier • 24-bit system tick timer • Programmble and maskable interrupt • Low Power Sleep mode by WFI and WFE instructions

Arm CoreLink™ GIC-400 Generic Interrupt Controller (GIC)	• Up to 170 interrupts
	  – 16 Software Generated Interrupt (SGIs)
	  – 4 external Private Peripheral Interrupts (PPIs) for each processor
	  – 1 internal PPI for each processor
	  – 130 Shared Peripheral Interrupt (SPIs)
	• Interrupt Enabled or Disabled
	• Interrupt Prioritized
	• Interrupt to dual Arm Cortex-A35 processor core
	• Level-sensitive interrupt
	• Security Extensions
	  – Group 0 interrupt as Secure interrupts
	  – Group 1 interrupt as Non-secure interrupts
	• Virtualization Extensions

Hardware Semaphore	• Eight hardware semaphores for inter-processor synchronization • Support interrupt when semaphore released

Wormhole	• Supports two wormhole controllers for inter-processor communication • Supports interrupt for reset event and power state change event • Four general event interrupts • Eight 16-byte unidirectional message channels, four for each processor • Supports message recall mechanism
Watchdog	• Three Watchdogs, one for TrustZone Secure (TZS), one for TrustZone Secure/Non-Secure (TZS/TZNS) and the other for SubM • 20-bit free running up counter for WDT time-out interval • Supports multiple clock sources from LIRC (default selection), PCLK/4096 or LXT with 9 selectable time-out period • Able to wake up system from Power-down or Idle mode • Time-out event to trigger interrupt or reset system • Supports four WDT reset delay periods, including 1026, 130, 18 or 3 WDT_CLK reset delay period • Configured to force WDT enabled on chip power-on or reset
Window Watchdog	• Three Window Watchdogs, one for TrustZone Secure (TZS), one for TrustZone Secure/Non-Secure (TZS/TZNS) and the other for SubM • Clock sourced from LIRC (default selection) or PCLK/4096; the window set by 6-bit counter with 11-bit prescale • Suspended in Idle/Power-down mode
Real-Time Clock (RTC)	• Real-Time Clock with a separate power domain (V_{BAT}) • RTC clock source including Low-speed external crystal oscillator (LXT) • RTC block including 64 bytes backup registers • Able to wake up CPU • Supports ±5ppm within 5 seconds software clock accuracy compensation • Supports Alarm registers (second, minute, hour, day, month, year) • Supports RTC Time Tick and Alarm Match interrupt • Automatic leap year recognition • Supports 1 Hz clock to be Timer capture source for calibration • Supports 1 pairs dynamic loop tamper pin or 2 individual tamper pin
External Clock Source	• 24 MHz High-speed external crystal oscillator (HXT) for precise timing operation • 32.7688 kHz Low-speed external crystal oscillator (LXT) for RTC function and low-power operation • Supports clock failure detection for external crystal oscillators and exception generatation (NMI)
Internal Clock Source	• 32 kHz Low-speed Internal RC oscillator (LIRC) • 12 MHz High-speed Internal RC oscillator (HIRC) trimmed to 5% accuracy at -40 to 85°C

	• Six on-chip PLLs up to 800 MHz on-chip PLL, sourced from HXT, allows CPU operation up to the maximim CPU frequency
System Security Memory Configuration Controller (SSMCC)	• Three Arm CoreLink TZC-400 TrustZone Address Space Controllers for seven AXI channels • Configurable security attribution of SDRAM memory • Security violation detection, report and interrupt generation • Programmable SDRAM region for SubM system • Write Protect of security configuration
System Security Peripheral Configuration Controller (SSPCC)	• Configurable security attribution of SRAM by boundary • Configurable security attribution of GPIO by pin • Configurable security attribution of peripherals • Security violation detection, report and interrupt generation • Write Protect of security configuration • Debug interface protection mechanism • Product Life-cycle Management
Temperature Sensor	• Built-in temperature sensor with ±5°C accuracy
Low Voltage Detect (LVD)	• Two-level LVD with low voltage detect interrupt (2.8V/2.6V)
Low Voltage Reset (LVR)	• LVR with 2.4V threshold voltage level.
Memory Sub-system	
Boot Loader	• Factory pre-loaded 128 Kbytes mask ROM supporting four booting modes – Boot from USB (as device/host) – Boot from SD/eMMC – Boot from NAND Flash – Boot from SPI Flash (SPI-NOR/SPI-NAND)
SRAM	• Up to 384 (128 + 256) KB on-chip SRAM • Supports byte-, half-word- and word-access • Supports PDMA operation
SDRAM	• Supports DDR2 (Double-Data-Rate 2), DDR3 (Double-Data-Rate 3) and DDR3L (DDR3 Low Voltage) type of SDRAM • Clock speed up to 533 MHz • Supports 16-bit data width • SDRAM burst length of 8 • Up to 2 memory ranks • Maximum SDRAM size up to 2 GB (each chip select for 1 GB)
Security Sub-system	
Trusted Secure Island (TSI)	• An isolated secure hardware unit and operation is not affected by MA35D1 system • Built-in cryptographic accelerators, Key Store and OTP memory

	• Performs all the security operations, including secure boot and tamper-pin detection • Refer to the TSI Application Note document for more detailed information
True Random Number Generator (TRNG)	• Compliant with NIST SP800-90A/B/C and BSI AIS 20/31 • 128-bit random number generation • 128-bit or 256-bit of security strength • Background noise collection to speed reseeding operations • Internal random seeding operation • Start-up, continuous and on-demand health tests
Pseudo Random Number Generator (PRNG)	• Supports 128, 163, 192, 224, 233, 255, 256, 283, 384, 409, 512, 521 and 571 bits random number generation (283~571 bits only generate for Key Store) • Can take the true random number seed from TRNG • Can take the true random number from TRNG (only for Key Store)
Advanced Encryption Standard (AES)	• Hardware AES accelerator • Supports FIPS NIST 197 • Supports SP800-38A and addendum • Supports 128, 192, and 256 bits key • Supports both encryption and decryption • Supports ECB, CBC, CFB, OFB, CTR, CBC-CS1, CBC-CS2 and CBC-CS3 modes • Supports CCM mode, GCM mode and GHASH function • Supports SM4 block cipher algorithm • Supports key expander • Supports one technique to improve side-channel attack protection ability
Secure Hash Algorithm (SHA)	• Hardware SHA accelerator • Supports FIPS NIST 180, 180-2, 180-4 • Supports MD5 • Supports SHA-160, SHA-224, SHA-256, SHA-384, SHA-512 and SHA-512/t • Supports SHA3-224, SHA3-256, SHA3-384, SHA3-512, SHAKE128 and SHAKE256 • Supports SM3 Cryptographic Hash Algorithm
Elliptic Curve Cryptography (ECC)	• Hardware ECC accelerator • Supports both prime field GF(p) and binary filed GF(2^m) • Supports NIST P-192, P-224, P-256, P-384, and P-521 • Supports NIST B-163, B-233, B-283, B-409, and B-571 • Supports NIST K-163, K-233, K-283, K-409, and K-571 • Supports Curve25519

	• Supports Public Key Cryptographic Algorithm SM2 Based on Elliptic Curves • Supports point multiplication, addition and doubling operations in GF(p) and GF(2^m) • Supports modulus division, multiplication, addition and subtraction operations in GF(p) • Supports three techniques to improve side-channel attack protection ability
Rivest、Shamir and Adleman Cryptography (RSA)	• Hardware RSA accelerator • Supports both encryption and decryption with 1024, 2048, 3072 and 4096 bits • Supports CRT decryption with 2048, 3072 and 4096 bits • Supports three techniques to improve side-channel attack protection ability
Key Store	• Supports programming interface for key management • Supports key size required for Cryptography from 128 bits to 4096 bits • Supports 32 keys for SRAM and 9 keys for OTP at most • Supports crypto engine access or store key in key store directly • Supports ECDH operation with ECC and PRNG engine • Supports to store middle data for RSA CRT and SCAP mode • Supports revoke operation • Supports erase key in SRAM and revoke key in OTP while tamper detected • Supports integrity checking • Supports data scrambling • Supports data remanence prevention at SRAM • Supports silent access for side-channel protection at SRAM
OTP Controller	• Supports 32 bits programming function and reading function • Supports 8kbits Secure OTP memory • Supports data retention more than 10 years • Supports fault tolerant mechanism • Supports read only lock bit • Supports side-band handshaking signals with KeyStore
Display and Video Sub-system	

TFT LCD Display Interface	• Display Interface
	  – Supports parallel pixel output with 24-bit Data, HSync, VSync, Data enable
	  – Supports DPI 24-bit, 18-bit and 16-bit
	  – Supports i80/m68 MPU type interface with optional VSYNC or TE signal
	• Frame rate up to 1920 x 1080 @ 60 fps

	• Input Format
	  – ARGB2101010, A/XRGB8888, A/XRGB1555, RGB565, A/XRGB4444
	  – Index1/2/4/8
	  – YUV422 packed and semi planar (YUV2, UYVY, NV16)
	  – YUV420 semi-planar (YUY2(P010), NV12 and YUV420 semi-planar 10-bit)
	• Output Format
	  – DPI_D16CFG1, DPI_D16CFG2, DPI_D16CFG3
	  – DPI_D18CFG1, DPI_D18CFG2
	  – DPI_D24
	• Color Space Conversion BT.2020 and BT.709
	• Supports ARGB888 and Mask cursor formats for hardware cursor
	• Supports On Screen Display (OSD)

2D Graphic Engine (GFX)	• BitBLT, Stretch Blit and Filter Blit
	• Line drawing, Rectangle fill and clear
	• Mono expansion for text rendering
	• ROP2, ROP3 and ROP4
	• Alpha blending, including Java 2 Porter-Duff compositing blending rules
	• 32K x 32K coordinate system
	• 90 / 180 / 270 degree rotation
	• Transparency by monochrome mask, chroma key, or pattern mask
	• Supports 2x2 in 4x4 tile format
	• Supports XMajor and YMajor Super Tile 64x64 format
	• A8 output with rotation in filter blit and bit blit
	• Supports Source and Destination color key full bypass
	• Multi source blending
	  – Full support for Multi source blending with variable block size
	  – Up to 8 sources
	  – Programmable block size
	  – Supports 90, 180, 270 degree rotation with different block size
	• Supports format converting for non-planar YUV to planar YUV
	• YUV422 output with alpha blending

Video and Image Decoder	**AVC(H.264) / MVC / SVC Decoding**
	• Input stream format
	  – AVC(H.264) stream including Byte stream and NAL unit stream
	  – MVC stream
	  – SVC stream
	• Output picture format
	  – YCbCr 4:2:0 semi-planar raster-scan
	  – YCbCr 4:2:0 semi-planar 8x4 tiled
	  – YCbCr 4:0:0 (monochrome)

- Frame by frame (field by field) and slice by slice decoding scheme
- Picture size from 48 x 48 to 1920 x 1080 with step size 16 pixels
- Frame rate up to 1920 x 1080 @ 45 fps

JPEG Decoding

- Input picture format
  - JFIF file format 1.02
  - YCbCr 4:0:0, 4:2:0, 4:2:2, 4:4:0, 4:1:1 and 4:4:4 sampling formats
- Output picture format
  - YCbCr4:2:0 semi-planar raster scan
- Picture size from 48 x 48 to 16368 x 16368 with step size 8 pixels
- Supports JPEG compressed thumbnails

Post Processing

- Input data format
  - YCbCr 4:2:0 semi-planar raster-scan
  - YCbCr 4:2:0 semi-planar 8x4 tiled
  - YCbCr 4:2:0 planar
  - YCbCr 4:0:0 (monochrome)
  - YCbYCr 4:2:2, YCrYCb 4:2:2
  - CbYCrY 4:2:2, CrYCbY 4:2:2
- Output data format
  - YCbCr 4:2:0 semi-planar
  - YCbYCr 4:2:2 raster-scan or 4x4 tiled
  - YCrYCb 4:2:2 raster-scan or 4x4 tiled
  - CbYCrY 4:2:2 raster-scan or 4x4 tiled
  - CrYCbY 4:2:2 raster-scan or 4x4 tiled
  - Fully configurable ARGB channel lengths and locations inside 32 bits, such as ARGB 32-bit (8-8-8-8), RGB 16-bit (5-6-5), ARGB 16-bit (4-4-4-4)
- Input image size from 48 x 48 to 16368 x 16368 with step size 8 pixels
- Output image size from 16 x 16 to 1920 x 1080 with horizontal step size 8 and vertical step size 2
- Down Scaling
  - Arbitrary, non-integer scaling ratio separately for both dimensions
  - Unlimited down-scaling ratio
- Up Scaling
  - Arbitrary, non-integer scaling ratio separately for both dimensions
  - Maximum output width is 3x the input width (within the maximum output image size limit)
  - Maximum output height is 3x the input height – 2 pixels (within the maximum output image size limit)
- YCbCr to RGB color conversion
  - BT.601-5 compliant

	- BT.709 compliant - User definable conversion coefficient • Dithering • Alpha blending • De-interlacing • Contrast, brightness and color saturation adjustment for RGB image • Supports Image cropping and digital zoom • Picutre in picture, output image masking • Image rotation - Rotation 90, 180 or 270 degrees • Horizontal and vertical flip
CMOS Sensor Interface	• Two sets of CMOS sensor interfaces supporting CCIR601 and CCIR656 type sensor • Resolution up to 3M pixels • Supports YUV422 and RGB565 color format for data output by CMOS image sensor • Supports YUV422, RGB565, RGB555 and Y-only color format with planar and packet data format for data storing to system memory • Supports image cropping and cropping window up to 4096x2048 • Supports vertical and horizontal scaling-down with N/M scaling factor • Supports Negative, Sepia and Posterization color effects
Connectivity and I/O Sub-system	
Peripheral DMA (PDMA)	• Four sets of PDMA with 10 independent and configurable channels for automatic data transfer between memories and peripherals • Basic and Scatter-Gather transfer modes • Each channel supporting circular buffer management using Scatter-Gather Transfer mode • Stride function for rectangle image data movement • Fixed-priority and Round-robin priorities modes • Single and burst transfer types • Byte-, half-word- and word tranfer unit with count up to 65536 • Incremental or fixed source and destination address
Gigabit Ethernet MAC (GMAC)	• Two sets of Gigabit Ethernet MAC • Compliant with IEEE Std 802.3-2008 for Ethernet MAC • Compliant with IEEE Std 1588-2008 for precision networked clock synchronization • Compliant with IEEE Std 802.3az-2010 for Energy Efficient Ethernet (EEE) • Compliant with RGMII specification version 2.6 from HP/Marvell • Compliant with RMII specification version 1.2 from RMII consortium • Full-duplex operation

	- IEEE 802.3x flow control automatic transmission of zero-quanta Pause frame on flow control input de-assertion - Forwarding of received Pause frames to the user application • Half-duplex operation - CSMA/CD Protocol support - Flow control using backpressure support - Frame bursting and frame extension in 1000 Mbps half-duplex operation • Programmable frame length to support Standard or Jumbo Ethernet frames with up to 16 Kbytes of size • IEEE 802.1Q VLAN tag detection for reception frames • Receive module for checksum off-load for received IPv4 and TCP packets encapsulated by the Ethernet frame (Type 1) • Enhanced Receive module for checking IPv4 header checksum and TCP, UDP, or ICMP checksum encapsulated in IPv4 or IPv6 datagrams (Type 2) • CRC replacement, Source Address field insertion or replacement, and VLAN insertion, replacement, and deletion in transmitted frames with per-frame control • Serial management interface (MDC/MDIO) master mode for PHY device configuration and management • Supports Magic Packet recognition to wake system up from Power-down mode
USB 2.0 High Speed with on-chip transceiver	**USB 2.0 High Speed Host/Device** • One set of on-chip USB 2.0 high speed dual role transceiver configurable as host, device or ID-dependent • One set of on-chip USB 2.0 high speed transceiver with host only **USB 2.0 High Speed Host Controller** • Compliant with USB Revision 2.0 Specification • Compatible with EHCI (Enhanced Host Controller Interface) Revision 1.0 • Compatible with OHCI (Open Host Controller Interface) Revision 1.0 • Supports high-speed (480Mbps), full-speed (12Mbps) and low-speed (1.5Mbps) USB devices • Integrated with a port routing logic to route full/low speed device to OHCI controller • Supports an integrated Root Hub • Built-in DMA **USB 2.0 High Speed Device Controller** • Compliant with USB Revision 2.0 Specification • Supports up to eight bi-directional endpoints, in addition to control endpoint 0 • Supports Control, Bulk, Interrupt and Isochronous transfers • Supports Descriptor (Scatter-Gather) DMA operation • Supports LPM feature • Supports V_{BUS}/Resume wake-up from system power-down mode

Secure Digital Host Controller (SDHC)	• Two sets of Secure Digital Host Controllers • Supports 1-bit and 4-bit data bus width for SD memory card specification version 3.0. (SDR104 speed limited to maximum allowed I/O speed SPI mode, DDR50 and UHS-II mode not supported) • Supports 1-bit, 4-bit and 8-bit data bus widths for the eMMC interface(HS200 speed limited to maximum allowed I/O speed and HS400 not supported) • Supports SD/SDHC/SDXC/SDIO, eMMC card • Supports SD/eMMC tuning, CMD19(SD) or CMD21(eMMC) • Supports 200 MHz to achieve eMMC HS200 at 1.8V I/O operation
CAN FD	• Four sets of CAN FD controllers • Compliant with CAN protocol version 2.0 part A, B and ISO 11898-1: 2015 • Compliant with CAN FD version 1.0 with up to 64 data bytes supported • Supports CAN Error logging, AUTOSAR and SAE J1938 • Built-in 2K word (32-bit) Message SRAM for each CAN FD controller • Supports power-down
Quad SPI	• Two sets of SPI Quad controllers with Master/Slave mode, up to 100 MHz • Supports Dual and Quad I/O Transfer mode • Supports one data channel half-duplex transfer • Supports receive-only mode • Configurable bit length of a transfer word from 8 to 32-bit • Provides separate 8-level depth transmit and receive FIFO buffers • Supports MSB first or LSB first transfer sequence • Supports the byte reorder function • Supports Byte or Word Suspend mode • Supports 3-wired, no slave select signal, bi-direction interface • PDMA operation
SPI/I^{2}S	• Four sets of SPI/I^{2}S controllers with Master/Slave mode • For SPI PDMA function disable, provides separate 8-level of 32-bit or 16-level of 16-bit transmit and receive FIFO buffers • For SPI PDMA function enable, provides separate 8-level of 32-bit, 16-level of 16-bit or 32-level of 8-bit transmit and receive FIFO buffers **SPI** • Up to 100 MHz in Master mode • Configurable bit length of a transfer word from 8 to 32-bit • MSB first or LSB first transfer sequence • Byte reorder function

	• Supports Byte or Word Suspend mode • Supports one data channel half-duplex transfer • Supports receive-only mode • Supports 3-wired, no slave select signal, bi-direction interface **I²S** • Supports mono and stereo audio data with 8-, 16-, 24- and 32-bit audio data sizes • Supports PCM mode A, PCM mode B, I²S and MSB justified data format • PDMA operation
I²S	• Two sets of I²S interfaces with Master/Slave mode • Supports I²S audio sampling frequencies up to 192 kHz • Supports mono and stereo audio data with 8-, 16-, 24- and 32-bit word sizes • Two 16-level FIFO data buffers, one for transmitting and the other for receiving • Supports I²S protocols: Philips standard, MSB-justified, and LSB-justified data format • Supports PCM protocols: PCM standard, MSB-justified, and LSB-justified data format • PCM protocol supporting TDM multi-channel transmission in one audio sample; the number of data channel can be set as 2, 4, 6 or 8 • PDMA operation
Low-power UART	• 17 sets of UARTs with up to 9.5 MHz baud rate • Auto-Baud Rate measurement and baud rate compensation function • Supports low power UART (LPUART): baud rate clock from LXT(32.768 kHz) with 9600bps in Power-down mode even system clock is stopped • Separated receive and transmit 32/32-byte FIFOs with receive FIFO programmable level trigger supported • Auto flow control (nCTS and nRTS) • Supports IrDA (SIR) function • Supports RS-485 9-bit mode and direction control • Supports nCTS, incoming data, Received Data FIFO reached threshold and RS-485 Address Match (AAD mode) wake-up function in idle mode • Supports hardware or software enables to program nRTS pin to control RS-485 transmission direction • Supports wake-up function • 8-bit receiver FIFO time-out detection function • Supports break error, frame error, parity error and receive/transmit FIFO overflow detection function • Supports Single-wire mode • PDMA operation

I²C	• Six sets of I²C devices with Master/Slave mode • Supports Standard mode (100 kbps), Fast mode (400 kbps) and Fast mode plus (1 Mbps) • Supports 10 bits mode • Programmable clocks allowing for versatile rate control • Supports multiple address recognition (four slave address with mask option) • Supports multi-address power-down wake-up function • PDMA operation
Enhanced PWM (EPWM)	• Eighteen 16-bit counters with 12-bit clock prescale for eighteen PWM output channels • Up to 18 independent input capture channels with 16-bit resolution counter • Supports dead zone with maximum divided 12-bit prescale • Up, down or up-down PWM counter type • Supports complementary mode for 3 complementary paired PWM output channels • Synchronous function for phase control • Counter synchronous start function • Brake function with auto recovery mechanism • Mask function and tri-state output for each PWM channel • Able to trigger EADC to start conversion
32-bit Timer	**Timer** • Twelve sets of 32-bit timers with 24-bit up counter and one 8-bit pre-scale counter from independent clock source • One-shot, Periodic, Toggle and Continuous Counting operation modes • Event counting function to count the event from external pin • Input capture function to capture or reset counter value • External capture pin event for interval measurement • External capture pin event to reset 24-bit up counter • Chip wake-up from Idle/Power-down mode if a timer interrupt signal is generated • Timer interrupt flag or external capture interrupt flag to trigger EPWM, EADC and PDMA • internal capture triggered from internal clock (HIRC, LIRC) or external clock (HXT, LXT) • Inter-Timer trigger capture mode **PWM** • Twelve16-bit PWM counters with 12-bit clock prescale • Supports 12-bit deadband (dead zone) • Up, down or up-down PWM counter type • Supports brake function

	• Supports mask function and tri-state output for each PWM channel
Quadrature Encoder Interface (QEI)	• Three sets of QEIs with two QEI phase inputs (QEI_A, QEI_B) and one Index input (QEI_INDEX) each • Supports 2/4 times free-counting mode and 2/4 compare-counting mode • Supports encoder pulse width measurement mode with ECAP
Smart Card Interface	• Two sets of ISO-7816-3 compliant with ISO-7816-3 T=0, T=1 • Supports full duplex UART function • 4-byte FIFOs with programmable level trigger • Programmable guard time selection (11 ETU ~ 266 ETU) • One 24-bit and two 8 bit time-out counters for Answer to Request (ATR) and waiting times processing • Auto inverse convention function • Stop clock level and clock stop (clock keep) function • Transmitter and receiver error retry function • Supports hardware activation, deactivation and warm reset sequence process • Supports hardware auto deactivation sequence after card removal
Enhanced Capture (ECAP)	• Three sets of ECAPs • Input Capture Timer/Counter • Supports three input channels with independent capture counter hold register • 24-bit Input Capture up-counting timer/counter supporting captured events reset and/or reload capture counter • Supports rising edge, falling edge and both edge detector options with noise filter in front of input ports • Supports compare-match function
NAND Flash Controller	• Supports SLC and MLC type NAND Flash device • Supports 2 Kbytes, 4 Kbytes and 8 Kbytes page size NAND Flash device • 8-bit data width • Supports EDO mode • Supports ECC8, ECC12 and ECC24 BCH algorithm with ECC code generation, error detection and error correction • Supports dedicated DMA master with Scatter-Gather function to accelerate the data transfer between system memory and NAND Flash
KeyPad Interface (KPI)	• Matrix keypad interface with up to 8x8 array • Programmable de-bounce time • Low-power wake-up mode • Programmable three-key reset • Generate interrupt and update press/release status of all keys once key press or release detected

GPIO	• Supports three I/O modes
	  – Push-Pull output mode
	  – Open-Drain output mode
	  – Input only with high impendence mode
	• Selectable TTL/Schmitt trigger input
	• Configured as interrupt source with edge/level trigger setting
	• Supports independent pull-up/pull-down control

Enhanced Analog-to-Digital Converter (EADC)	• One 12-bit, 4.7 MSPS SAR EADC with up to 8 single-ended input channels or 4 differential input pairs; 10-bit accuracy is guaranteed • One internal channels for Battery power (V_{BAT}) • Supports external V_{REF} pin or internal reference voltage • Analog-to-Digital conversion can be triggered by software enable, external pin, Timer 0~11 overflow pulse trigger, ADINT0 and ADINT1 EOC interrupt pulse trigger, or EPWM trigger • Configurable EADC sampling time • Up to 9 sample modules, supporting double data buffers for sample module 0~3 • Supports PDMA operation
Analog-to-Digital Converter (ADC)	• One 12-bit, 8-ch 500k SPS SAR ADC with up to 8 single-ended input channels; 10-bit accuracy is guaranteed • Supports 4-wire or 5-wire touch screen • Supports external V_{REF} pin
External Bus Interface (EBI)	• Supports up to three memory banks with individual adjustment of timing parameter • Each bank supporting dedicated external chip select pin with polarity control and up to 1 Mbytes addressing space • 8-/16-bit data width • Supports byte write enable in 16-bit data width mode • Configurable idle cycle for different access condition: Idle of Write command finish (W2X) and Idle of Read-to-Read (R2R) • Supports Address/Data multiplexed mode • Supports address bus and data bus separate mode • PDMA operation

3 PARTS INFORMATION

3.1 MA35D1 Series Package Types

Package is Halogen-free, RoHS-compliant and TSCA-compliant.

Part No.	LQFP-EP 216-Pin	BGA 312-Ball	BGA 364-Ball
MA35D1	MA35D16FH87C[*1] / MA35D16FI87C[*1] / MA35D16FJ87C[*1] /	MA35D16AI87C[*1] / MA35D16AJ87C[*1]	MA35D16A087C[*2]
Note 1: This part number is stacked with the DDR3L-type SDRAM (v1). The supply voltage of MV_{DD} power input pin is 1.35V for the internal DDR3L-type SDRAM and MV_{REF} = 1/2MV_{DD}. **Note 2:** The supply voltage of MV_{DD} power input pin depends on the external SDRAM type and MV_{REF_CA} = MV_{REF_DQ} = 1/2MV_{DD}.			

Table 3.1-1 MA35D1 Series Package Types (New)

Part No.	LQFP-EP 216-Pin	BGA 312-Ball
MA35D1	MA35D16F787C[*1] / MA35D16F887C[*2] / MA35D16F987C[*2] / MA35D14F884[*2] / MA35D14F984[*2]	MA35D16A887C[*2] /
Note 1: This part number is stacked with the DDR2-type SDRAM (v0). The supply voltage of MV_{DD} power input pin is 1.8V for the internal DDR2-type SDRAM and MV_{REF} = 1/2MV_{DD}. **Note 2:** This part number is stacked with the DDR3L-type SDRAM (v0). The supply voltage of MV_{DD} power input pin is 1.35V for the internal DDR3L-type SDRAM and MV_{REF} = 1/2MV_{DD}.		

Table 3.1-2 MA35D1 Series Package Types

3.2 MA35D1 Series Naming Rule

M	A35	D1	6	A	8	8	7	C
MPU	CPU	Series Name	Function	Package	DRAM Size	CPU Speed	Cores	CAN FD
Nuvoton Microprocessor	Arm A35 Core	D1: High-Performance Edge IIoT Series	4: Ethernet + Display + H.264 decoder 6: Ethernet + Display + H.264 decoder	F: LQFP216 A: BGA	0: Non 7: 128 MB v0 8: 256 MB v0 9: 512 MB v0 H: 128 MB v1 I: 256 MB v1 J: 512 MB v1	8: 800 MHz	A35 M4 TSI 4 2 x x 7 2 O O	C: CAN FD N/A

Figure 3.2-1 MA35D1 Series Part Number Naming Guide

3.3 MA35D1 Series Selection Guide

3.3.1 LQFP Package Types (New)

Part Number		MA35D16FH87C / MA35D16FI87C / MA35D16FJ87C
	Cortex-A35	x2
	CPU Frequency	800 MHz
	L1-Cache I/D (KB)	32+32
	L2-Cache (KB)	512
	Real-time Processor with Cortex-M4	x1
	Cortex-M4 I/D Cache (KB)	16+16
	SRAM (KB) (Cortex-A35 System + Cortex-M4 RTP)	384 (256+128)
	Stacked DDR Size (MB)	128 (DDR3L[*1]) / 256 (DDR3L[*1]) / 512 (DDR3L[*1])
	LCD Display (RGB)	24-bit/18-bit
	2D Graphic and JPEG Decoder	√
	H.264 Decoder	√
	CMOS Sensor Interface	2
	10 / 100Mb / 1Gb Ethernet MAC	10/100M x1 + 1G x1
	USB 2.0 HS Host	1
	USB 2.0 HS Host / Device	1
	Secure Boot	√
Booting	USB	√
	NAND Flash	√
	SPI-NOR Flash (1/4 bit)	√
	SPI-NAND (1/4 bit)	√
	SD/eMMC	√
Connectivity	UART	17
	ISO-7816 (Smart Card I/F)	2
	Quad SPI	2
	SPI/I^2S	4
	I^2S	2
	I^2C	6
	CAN FD	4
	SDHC/SDIO/eMMC	2

Part Number		MA35D16FH87C / MA35D16FI87C / MA35D16FJ87C
Crypto	PRNG 512	√
	AES 256	√
	RSA 4096	√
	ECC	√
	TRNG 128	√
	SHA 512	√
	SM 2/3/4	√
	Key Store	√
	OTP 8K-bit	√
	Tamper pin	--
External Bus Interface		√
32-bit Timer		12
Enhanced Capture (ECAP)		3
Enhanced PWM (EPWM)		18
Quadrature Encoder Interface (QEI)		2
12-bit ADC channel		8
12-bit EADC channel		--
GPIO		154
INT		4
Watchdog		3
Window Watchdog		3
RTC (V_{BAT} 3.3V)		√
Power Control Pin for PMIC		√
Temperature Sensor		√
Operating Junction Temperature (T_J)		-40°C to 105°C
Package		LQFP-EP 216-Pin 24x24x1.4mm, 0.4mm pitch

Note 1: For the internal DDR3L-type SDRAM (v1), the supply voltage of MV_{DD} power input pin is 1.35V and MV_{REF} = 1/2MV_{DD}.

3.3.2 BGA Package Types (New)

Part Number		MA35D16AI87C / MA35D16AJ87C	MA35D16A087C
	Cortex-A35	x2	x2
	CPU Frequency	800 MHz	800 MHz
	L1-Cache I/D (KB)	32+32	32+32
	L2-Cache (KB)	512	512
	Real-time Processor with Cortex-M4	x1	x1
	Cortex-M4 I/D Cache (KB)	16+16	16+16
	SRAM (KB) (Cortex-A35 System + Cortex-M4 RTP)	384 (256+128)	384 (256+128)
	Stacked DDR Size (MB)	256 (DDR3L[*1]) 512 (DDR3L[*1])	External[*2]
	LCD Display (RGB)	24-bit/18-bit	24-bit/18-bit
	2D Graphic and JPEG Decoder	√	√
	H.264 Decoder	√	√
	CMOS Sensor Interface	2	2
	10 / 100Mb / 1Gb Ethernet MAC	1G x2	1G x2
	USB 2.0 HS Host	1	1
	USB 2.0 HS Host / Device	1	1
	Secure Boot	√	√
Booting	USB	√	√
	NAND Flash	√	√
	SPI-NOR Flash (1/4 bit)	√	√
	SPI-NAND (1/4 bit)	√	√
	SD/eMMC	√	√
Connectivity	UART	17	17
	ISO-7816 (Smart Card I/F)	2	2
	Quad SPI	2	2
	SPI/I^2S	4	4
	I^2S	2	2
	I^2C	6	6
	CAN FD	4	4
	SDHC/SDIO/eMMC	2	2

Part Number		MA35D16AI87C / MA35D16AJ87C	MA35D16A087C
Crypto	PRNG 512	√	√
	AES 256	√	√
	RSA 4096	√	√
	ECC	√	√
	TRNG 128	√	√
	SHA 512	√	√
	SM 2/3/4	√	√
	Key Store	√	√
	OTP 8K-bit	√	√
	Tamper pin	2	2
External Bus Interface		√	√
32-bit Timer		12	12
Enhanced Capture (ECAP)		3	3
Enhanced PWM (EPWM)		18	18
Quadrature Encoder Interface (QEI)		3	3
12-bit ADC channel		8	8
12-bit EADC channel		8	8
GPIO		208	208
INT		4	4
Watchdog		3	3
Window Watchdog		3	3
RTC (V_{BAT} 3.3V)		√	√
Power Control Pin for PMIC		√	√
Temperature Sensor		√	√
Operating Junction Temperature (T_J)		-40°C to 105°C	-40°C to 105°C
Package		LFBGA 312-Ball 15x15x1.4mm, 0.8mm pitch	LFBGA 364-Ball 14x14x1.4mm, 0.65mm pitch

Note 1: For the internal DDR3L-type SDRAM (v1), the supply voltage of MV_{DD} power input pin is 1.35V and MV_{REF} = 1/2MV_{DD}.
Note 2: The supply voltage of MV_{DD} power input pin depends on the external SDRAM type and MV_{REF_CA} = MV_{REF_DQ} = 1/2MV_{DD}.

3.3.3 LQFP Package Types

Part Number		MA35D14F884 / MA35D14F984	MA35D16F787C / MA35D16F887C / MA35D16F987C
	Cortex-A35	x2	x2
	CPU Frequency	800 MHz	800 MHz
	L1-Cache I/D (KB)	32+32	32+32
	L2-Cache (KB)	512	512
	Real-time Processor with Cortex-M4	--	x1
	Cortex-M4 I/D Cache (KB)	16+16	16+16
	SRAM (KB) (Cortex-A35 System + Cortex-M4 RTP)	384	384 (256+128)
	Stacked DDR Size (MB)	256 (DDR3L[*2]) / 512 (DDR3L[*2])	128 (DDR2[*1]) / 256 (DDR3L[*2]) / 512 (DDR3L[*2])
	LCD Display (RGB)	24-bit/18-bit	24-bit/18-bit
	2D Graphic and JPEG Decoder	√	√
	H.264 Decoder	√	√
	CMOS Sensor Interface	2	2
	10 / 100Mb / 1Gb Ethernet MAC	10/100M x1 + 1G x1	10/100M x1 + 1G x1
	USB 2.0 HS Host	1	1
	USB 2.0 HS Host / Device	1	1
	Secure Boot	√	√
Booting	USB	√	√
	NAND Flash	√	√
	SPI-NOR Flash (1/4 bit)	√	√
	SPI-NAND (1/4 bit)	√	√
	SD/eMMC	√	√
Connectivity	UART	17	17
	ISO-7816 (Smart Card I/F)	2	2
	Quad SPI	2	2
	SPI/I^2S	4	4
	I^2S	2	2
	I^2C	6	6
	CAN FD	--	4
	SDHC/SDIO/eMMC	2	2

Part Number		MA35D14F884 / MA35D14F984	MA35D16F787C / MA35D16F887C / MA35D16F987C
Crypto	PRNG 512	√	√
	AES 256	√	√
	RSA 4096	√	√
	ECC	√	√
	TRNG 128	√	√
	SHA 512	√	√
	SM 2/3/4	√	√
	Key Store	√	√
	OTP 8K-bit	√	√
	Tamper pin	--	--
External Bus Interface		√	√
32-bit Timer		12	12
Enhanced Capture (ECAP)		3	3
Enhanced PWM (EPWM)		18	18
Quadrature Encoder Interface (QEI)		2	2
12-bit ADC channel		8	8
12-bit EADC channel		--	--
GPIO		154	154
INT		4	4
Watchdog		3	3
Window Watchdog		3	3
RTC (V_{BAT} 3.3V)		√	√
Power Control Pin for PMIC		√	√
Temperature Sensor		√	√
Operating Junction Temperature (T_J)		-40°C to 105°C	-40°C to 105°C
Package		LQFP-EP 216-Pin 24x24x1.4mm, 0.4mm pitch	LQFP-EP 216-Pin 24x24x1.4mm, 0.4mm pitch

Note 1: For the internal DDR2-type SDRAM (v0), the supply voltage of MV_{DD} power input pin is 1.8V and MV_{REF} = 1/2MV_{DD}.
Note 2: For the internal DDR3L-type SDRAM (v0), the supply voltage of MV_{DD} power input pin is 1.35V and MV_{REF} = 1/2MV_{DD}.

3.3.4 BGA Package Types

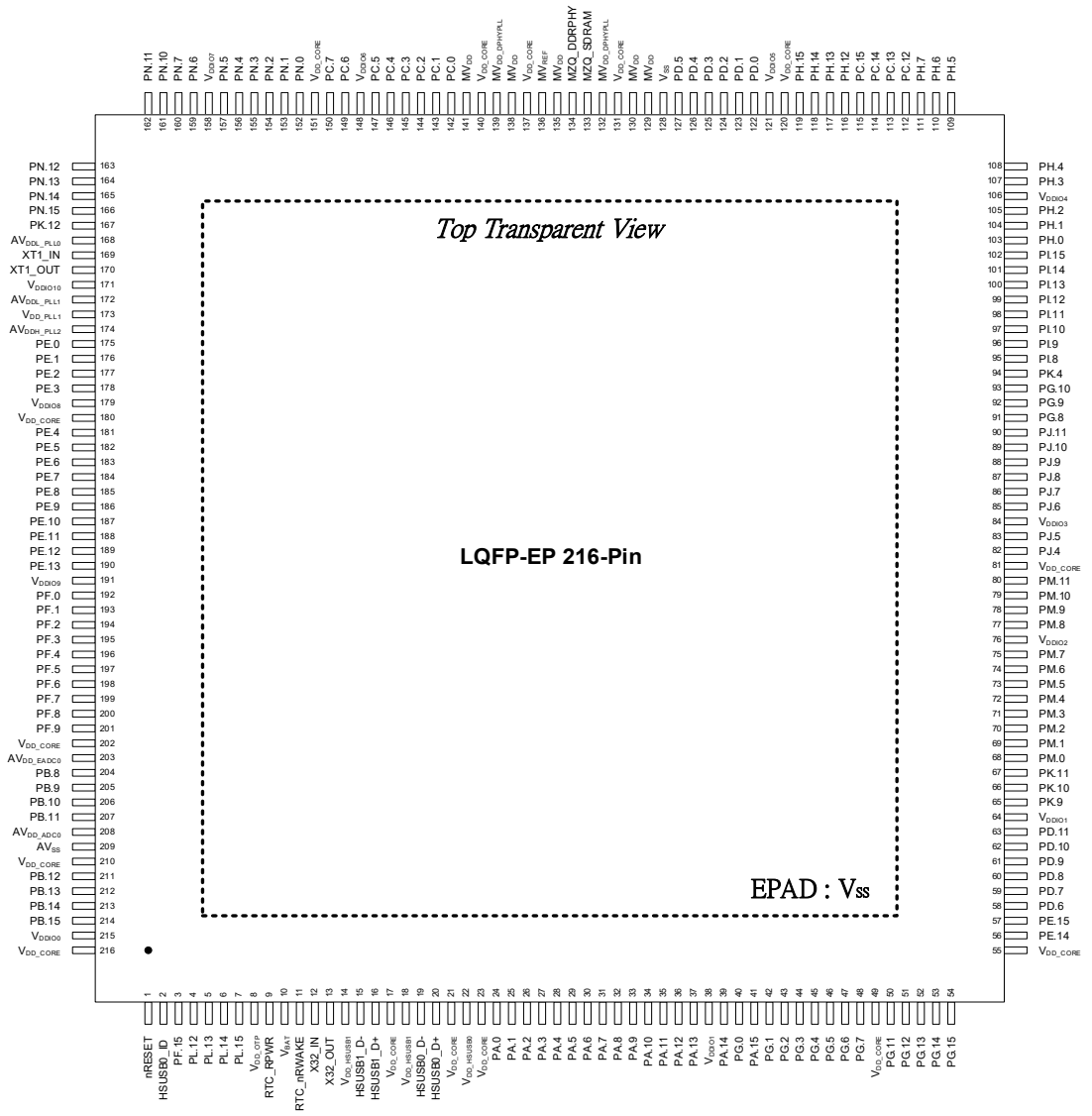
Part Number		MA35D16A887C
Cortex-A35		x2
CPU Frequency		800 MHz
L1-Cache I/D (KB)		32+32
L2-Cache (KB)		512
Real-time Processor with Cortex-M4		x1
Cortex-M4 I/D Cache (KB)		16+16
SRAM (KB) (Cortex-A35 System + Cortex-M4 RTP)		384 (256+128)
Stacked DDR Size (MB)		256 (DDR3L[*1])
LCD Display (RGB)		24-bit/18-bit
2D Graphic and JPEG Decoder		√
H.264 Decoder		√
CMOS Sensor Interface		2
10 / 100Mb / 1Gb Ethernet MAC		1G x2
USB 2.0 HS Host		1
USB 2.0 HS Host / Device		1
Secure Boot		√
Booting	USB	√
	NAND Flash	√
	SPI-NOR Flash (1/4 bit)	√
	SPI-NAND (1/4 bit)	√
	SD/eMMC	√
Connectivity	UART	17
	ISO-7816 (Smart Card I/F)	2
	Quad SPI	2
	SPI/I^2S	4
	I^2S	2
	I^2C	6
	CAN FD	4
	SDHC/SDIO/eMMC	2

Part Number		MA35D16A887C
Crypto	PRNG 512	√
	AES 256	√
	RSA 4096	√
	ECC	√
	TRNG 128	√
	SHA 512	√
	SM 2/3/4	√
	Key Store	√
	OTP 8K-bit	√
	Tamper pin	2
External Bus Interface		√
32-bit Timer		12
Enhanced Capture (ECAP)		3
Enhanced PWM (EPWM)		18
Quadrature Encoder Interface (QEI)		3
12-bit ADC channel		8
12-bit EADC channel		8
GPIO		208
INT		4
Watchdog		3
Window Watchdog		3
RTC (V_{BAT} 3.3V)		√
Power Control Pin for PMIC		√
Temperature Sensor		√
Operating Junction Temperature (T_J)		-40°C to 105°C
Package		LFBGA 312-Ball 15x15x1.4mm, 0.8mm pitch
Note 1: For the internal DDR3L-type SDRAM (v0), the supply voltage of MV_{DD} power input pin is 1.35V and MV_{REF} = 1/2MV_{DD}.		

4 PIN CONFIGURATION

4.1 Pin Configuration

4.1.1 MA35D1 Series LQFP-EP 216-Pin Pinout Diagram



Note: When soldering SMD, please make sure the EPAD on the bottom of LQFP-EP 216-Pin package is tightly soldered to the ground of the PCB.

Figure 4.1-1 NuMicro MA35D1 Series LQFP-EP 216-Pin Pinout Diagram

Note: To make the EPAD and PCB soldered tightly, please download the MA35 Series LQFP-EP 216-Pin Package Application Note document for reference in PCB design and layout as well as PCBA assembly and soldering.

4.1.2 MA35D1 Series BGA 312-Ball Pinout Diagram

	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	T	U	V
18	V_{SS}	PN.2	PN.3	PC.1	PC.2	PC.0	PC.3	PC.4	PL.5	PL.4	PL.3	PL.0	PL.1	PL.2	PH.14	PH.15	PH.13	V_{SS}
17	PN.5	PN.0	PN.1	V_{SS}	PC.7	PC.6	PC.5	PL.6	PL.7	PL.8	PL.9	PL.10	PL.11	PK.8	PH.12	PC.13	PC.15	PD.1
16	PN.11	PN.7	PN.4	V_{SS}	V_{SS}	PI.6	PK.1	PK.0	PK.3	PK.2	PJ.12	PJ.13	PJ.15	PJ.14	PC.14	PC.12	PD.0	PD.2
15	PN.12	PN.10	PN.6	V_{SS}	V_{SS}	PD.15	PA.15	PI.1	PI.2	PI.0	PI.3	PI.5	PI.4	PI.7	V_{SS}	PD.5	PD.3	PD.4
14	PN.14	PN.13	PK.14	PK.13			PD.13	MV_{REF}	MZQ_DDRPHY	MZQ_SDRAM	PD.14	PD.12			PH.7	PH.6	PH.5	PH.4
13	XT1_IN	XT1_OUT	PK.12	PN.15		V_{DDIO7}	V_{DDIO6}	MV_{DD}	MV_{DD}	MV_{DD}	MV_{DD}	V_{DDIO5}	V_{DDIO4}		PH.3	PH.2	PH.1	PH.0
12	PE.2	PE.1	PE.0	PK.15	AV_{DDL_PLL2}	V_{DD_PLL1}	V_{DDIO10}	V_{DD_CORE}	V_{DD_CORE}	V_{DD_CORE}	V_{DD_CORE}	$MV_{DD_DPHYPLL}$	V_{DDIO3}	PK.7	PI.14	PI.13	PI.12	PI.11
11	PE.5	PE.6	PE.4	PE.3	V_{DDIO8}	AV_{DDL_PLL1}	AV_{DDL_PLL0}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{DD_CORE}	PK.6	PI.9	PI.8	PI.15	PI.10
10	PE.8	PE.9	PE.7	PE.10	V_{DDIO9}	AV_{DDH_PLL2}	AV_{DDH_PLL1}	AV_{SS_PLL}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{DD_CORE}	PK.5	PK.4	PG.8	PG.9	PG.10
9	PE.11	PE.12	PE.13	PF.0	PH.8	V_{REF_EADC0}	AV_{DD_EADC0}	AV_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{DD_CORE}	PJ.4	PJ.9	PJ.6	PJ.8	PJ.10
8	PF.1	PF.2	PF.3	PF.4	PH.9	AV_{DD}	AV_{DD_ADC0}	V_{DD_CORE}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{DD_CPU}	PJ.5	PJ.3	PJ.2	PJ.11	PJ.7
7	PF.6	PF.7	PF.9	PF.8	RTC_RPWR	V_{BAT}	V_{DD_HSUSB0}	AV_{DDL_ROSC}	V_{DD_CORE}	V_{DD_CORE}	V_{DD_CPU}	V_{DD_CPU}	V_{DD_CPU}	PM.1	PJ.1	PM.8	PM.10	PJ.0
6	PF.5	PF.13	PF.12	PF.10		V_{DD_HSUSB1}	V_{DD_OTP}	V_{DD_CORE}	V_{DD_CORE}	V_{DD_CORE}	V_{DDIO0}	V_{DDIO1}	V_{DDIO2}		PM.9	PM.7	PM.5	PM.6
5	PF.11	EADC0_CH1	EADC0_CH0	PF.14			RTC_nRWAKE	nRESET	HSUSB0_ID	PA.10	PG.6	PA.14			PM.11	PM.3	PM.4	PM.2
4	EADC0_CH2	EADC0_CH5	EADC0_CH4	V_{SS}	PF.15	PA.3	PA.5	PA.7	PA.8	PA.12	PG.4	PG.3	PG.7	PG.11	V_{SS}	PK.11	PM.0	PK.10
3	EADC0_CH6	PB.8	EADC0_CH7	EADC0_CH3	PL.13	PL.15	PA.0	PA.1	PA.6	PA.9	PG.2	PG.5	PM.12	PG.15	PD.7	PD.11	PD.10	PK.9
2	PB.9	PB.11	PB.13	PB.10	PL.14	V_{SS}	X32_OUT	HSUSB1_D-	HSUSB0_D-	PA.4	PA.11	PG.0	PM.14	PG.12	PG.13	PE.15	PD.8	PD.9
1	V_{SS}	PB.12	PB.15	PB.14	PL.12	V_{SS}	X32_IN	HSUSB1_D+	HSUSB0_D+	PA.2	PA.13	PG.1	PM.13	PM.15	PG.14	PE.14	PD.6	V_{SS}

Top Transparent View

Figure 4.1-2 NuMicro MA35D1 Series BGA 312-Ball Pinout Diagram

4.1.3 MA35D1 Series BGA 364-Ball Pinout Diagram

	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	T	U	V	W	Y
20	V_{SS}	PC.3	MD14	MD12	MDMask1	MDQS1_N	MD10	MD8	MDQS0_N	MD3	MD0	MCK_P	MD6	MD4	MD5	V_{SS}	PD.3	PD.1	PH.14	V_{SS}
19	PC.1	PC.5	V_{SS}	MD15	MD13	MDQS1_P	MD9	MD11	MDQS0_P	MD2	MD1	MCK_N	MDMask0	MD7	V_{SS}	PD.0	PD.5	PH.12	PH.15	PD.4
18	PN.13	PC.2	PC.0	V_{SS}	MBA2	MA5	MA12	MA1	MCASn	MWEn	MA6	MA4	MA9	MA14	MA8	V_{SS}	PH.13	PD.2	PC.13	PH.4
17	PN.2	PN.0	V_{SS}	V_{SS}	MBA1	MA10	MA3	MA0	MBA0	MRASn	MA2	MA11	MA7	MA15	MA13	V_{SS}	PC.12	PC.15	PH.6	PH.3
16	PN.6	PN.4	V_{SS}	V_{SS}	PC.7	PC.6	V_{SS}	MCS1n	MCKE1	MCKE0	MODT1	MODT0	MRESETn	MCS0n	V_{SS}	PK.7	PH.7	PH.5	PH.1	PI.15
15	XT1_IN	XT1_OUT	PN.5	PC.4	PK.12											PK.5	PH.0	PI.14	PI.13	PI.11
14	PN.11	PN.12	PN.10	PN.3	PN.15		V_{DDIO6}	V_{DD_PLL1}	V_{DDIO10}	$MV_{DD_DPHYPLL}$	MV_{REF_DQ}	MV_{REF_CA}	MZQ_DDRPHY	V_{DDIO5}		PK.6	PI.10	PI.12	PC.14	PI.9
13	PN.1	PK.15	PK.14	V_{SS}	PK.13		V_{DDIO7}	AV_{DDL_PLL0}	V_{DD_CORE}	MV_{DD}	MV_{DD}	MV_{DD}	MV_{DD}	V_{DDIO4}		PJ.5	PG.9	PK.4	PG.10	PH.2
12	PN.7	PE.1	PE.2	PE.0	PN.14		AV_{DDL_PLL2}	AV_{DDL_PLL1}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{DD_CORE}	V_{DDIO3}		PJ.4	PJ.9	PI.8	PJ.10	PG.8
11	PE.3	PE.8	PE.10	PE.4	EADC0_CH0		AV_{DDH_PLL2}	AV_{DDH_PLL1}	AV_{SS_PLL}	V_{SS}	V_{SS}	V_{SS}	V_{DD_CORE}	V_{DD_CORE}		PL.5	PM.9	PJ.7	PJ.6	PJ.8
10	PE.5	PE.7	PF.2	PE.12	EADC0_CH4		V_{REF_EADC0}	AV_{DD_EADC0}	AV_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{DD_CPU}	V_{DD_CORE}		PL.0	PM.5	PM.6	PJ.11	PJ.2
9	PE.9	PE.13	PF.4	PE.6	PB.8		V_{BAT}	AV_{DD}	AV_{DD_ADC0}	V_{SS}	V_{SS}	V_{SS}	V_{DD_CPU}	V_{DD_CORE}		PM.1	PM.3	PM.4	PJ.1	PJ.0
8	PE.11	PF.7	PF.6	PF.0	PB.10		V_{DD_HSUSB1}	V_{DD_HSUSB0}	V_{DD_CORE}	AV_{DDL_ROSC}	V_{DD_CORE}	V_{DD_CPU}	V_{DD_CPU}	V_{DDIO2}		PL.1	PK.11	PM.2	PM.10	PJ.3
7	PF.1	PF.9	PF.8	PL.15	nRESET		V_{DDIO8}	V_{DDIO9}	V_{DD_OTP}	V_{DD_CORE}	V_{DD_CORE}	V_{DD_CORE}	V_{DDIO0}	V_{DDIO1}		PD.8	PM.11	PM.7	PM.0	PM.8
6	PF.3	PF.12	EADC0_CH1	PF.15	PL.13											PL.7	PE.15	PK.9	PL.4	PK.10
5	PF.5	PF.14	EADC0_CH5	PB.13	PH.9	PH.8	PA.1	PA.3	PA.8	PA.10	PA.14	PA.15	PM.14	PG.15	PK.3	PJ.13	PL.11	PL.3	PL.2	PK.8
4	PF.10	PF.13	PB.11	PB.14	PL.14	RTC_nRWAKE	PA.5	PA.6	PD.13	PA.12	PG.2	PG.6	PG.11	PI.1	PK.1	PJ.15	PL.9	PD.11	PD.6	PD.10
3	PF.11	EADC0_CH2	PB.9	PL.12	RTC_RPWR	HSUSB0_ID	PA.0	PA.9	PA.13	PD.12	PG.4	PM.12	PG.13	PI.3	PI.0	PI.5	PI.7	PD.9	PL.8	PD.7
2	EADC0_CH3	EADC0_CH6	PB.12	V_{SS}	X32_OUT	HSUSB1_D-	HSUSB0_D-	PA.4	PA.7	PD.15	PD.14	PG.7	PM.13	PG.14	PI.4	PJ.14	PJ.12	PK.2	PL.6	PL.10
1	V_{SS}	EADC0_CH7	PB.15	V_{SS}	X32_IN	HSUSB1_D+	HSUSB0_D+	PA.2	PA.11	PG.0	PG.1	PG.3	PG.5	PM.15	PG.12	PI.2	PI.6	PK.0	PE.14	V_{SS}

Top Transparent View

Figure 4.1-3 NuMicro MA35D1 Series BGA 364-Ball Pinout Diagram

4.2 Pin Description

4.2.1 MA35D1 Series LQFP-EP 216-Pin Pinout Function Table

4.2.1.1 *MA35D14F884 / MA35D14F984 Series*

Pin	Type	MA35D14F884 / MA35D14F984 (LQFP-EP 216-Pin, MCP with 256 / 512 MB DDR3L) Pin Function
1	I	nRESET / WDT_nRST
2	I	HSUSB0_ID
3	I/O	PF.15 / HSUSB0_VBUSVLD
4	I/O	PL.12 / EPWM0_SYNC_IN / UART7_nCTS / ECAP1_IC0 / UART14_RXD / SPI0_SS0 / I2S1_LRCK / SC1_CLK / EBI_AD0 / HSUSBH_PWREN / I2C2_SDA / TM0 / EPWM0_CH2 / EBI_AD11 / RGMII0_PPS / RMII0_PPS
5	I/O	PL.13 / EPWM0_SYNC_OUT / UART7_nRTS / ECAP1_IC1 / UART14_TXD / SPI0_CLK / I2S1_BCLK / SC1_DAT / EBI_AD1 / HSUSBH_OVC / I2C2_SCL / TM0_EXT / EPWM0_CH3 / EBI_AD12 / RMII1_PPS
6	I/O	PL.14 / EPWM0_CH2 / UART7_RXD / SPI0_MOSI / I2S1_DI / SC1_RST / EBI_AD2 / TM2 / INT0 / EBI_AD13
7	I/O	PL.15 / EPWM0_CH1 / UART7_TXD / SPI0_MISO / I2S1_DO / SC1_PWR / EBI_AD3 / TM2_EXT / INT2 / EBI_AD14
8	P	V_{DD_OTP}
9	O	RTC_RPWR
10	P	V_{BAT}
11	I	RTC_nRWAKE
12	I	X32_IN
13	O	X32_OUT
14	P	V_{DD_HSUSB1}
15	A	HSUSB1_D-
16	A	HSUSB1_D+
17	P	V_{DD_CORE}
18	P	V_{DD_HSUSB1}
19	A	HSUSB0_D-
20	A	HSUSB0_D+
21	P	V_{DD_CORE}
22	P	V_{DD_HSUSB0}
23	P	V_{DD_CORE}
24	I/O	PA.0 / UART1_nCTS / UART16_RXD / NAND_DATA0 / EBI_AD0 / EBI_ADR0
25	I/O	PA.1 / UART1_nRTS / UART16_TXD / NAND_DATA1 / EBI_AD1 / EBI_ADR1
26	I/O	PA.2 / UART1_RXD / NAND_DATA2 / EBI_AD2 / EBI_ADR2
27	I/O	PA.3 / UART1_TXD / NAND_DATA3 / EBI_AD3 / EBI_ADR3
28	I/O	PA.4 / UART3_nCTS / UART2_RXD / NAND_DATA4 / EBI_AD4 / EBI_ADR4

Pin	Type	MA35D14F884 / MA35D14F984 (LQFP-EP 216-Pin, MCP with 256 / 512 MB DDR3L) Pin Function
29	I/O	PA.5 / UART3_nRTS / UART2_TXD / NAND_DATA5 / EBI_AD5 / EBI_ADR5
30	I/O	PA.6 / UART3_RXD / NAND_DATA6 / EBI_AD6 / EBI_ADR6
31	I/O	PA.7 / UART3_TXD / NAND_DATA7 / EBI_AD7 / EBI_ADR7
32	I/O	PA.8 / UART5_nCTS / UART4_RXD / NAND_RDY0 / EBI_AD8 / EBI_ADR8
33	I/O	PA.9 / UART5_nRTS / UART4_TXD / NAND_nRE / EBI_AD9 / EBI_ADR9
34	I/O	PA.10 / UART5_RXD / NAND_nWE / EBI_AD10 / EBI_ADR10
35	I/O	PA.11 / UART5_TXD / NAND_CLE / EBI_AD11 / EBI_ADR11
36	I/O	PA.12 / UART7_nCTS / UART8_RXD / NAND_ALE / EBI_AD12 / EBI_ADR12
37	I/O	PA.13 / UART7_nRTS / UART8_TXD / NAND_nCS0 / EBI_AD13 / EBI_ADR13
38	P	V_{DDIO1}
39	I/O	PA.14 / UART7_RXD / NAND_nWP / EBI_AD14 / EBI_ADR14
40	I/O	PG.0 / EPWM0_CH0 / UART7_TXD / SPI0_SS0 / EBI_AD15 / I2S1_MCLK / QEI0_INDEX / TM1 / CLKO / INT0 / EBI_ADR15
41	I/O	PA.15 / EPWM0_CH2 / UART9_nCTS / UART6_RXD / I2C4_SDA / EBI_ALE / QEI0_A / TM1 / RGMII0_PPS / RMII0_PPS **Note:** This pin is dedicated as PMIC_nIRQ input function from external PMIC by default.
42	I/O	PG.1 / EPWM0_CH3 / UART9_nRTS / UART6_TXD / I2C4_SCL / EBI_nCS0 / QEI0_B / TM1_EXT / RMII1_PPS
43	I/O	PG.2 / EPWM0_CH4 / UART9_RXD / SPI0_SS1 / EBI_ADR16 / EBI_nCS2 / QEI0_A / TM3 / INT1
44	I/O	PG.3 / EPWM0_CH5 / UART9_TXD / SPI0_I2SMCLK / EBI_ADR17 / EBI_nCS1 / EBI_MCLK / QEI0_B / TM3_EXT / I2S1_MCLK
45	I/O	PG.4 / EPWM1_CH0 / UART5_nCTS / UART6_RXD / SPI3_SS0 / EBI_ADR18 / EBI_nCS0 / I2S1_DO / SC1_CLK / TM4 / INT2 / ECAP1_IC2
46	I/O	PG.5 / EPWM1_CH1 / UART5_nRTS / UART6_TXD / SPI3_CLK / ECAP0_IC0 / EBI_ADR19 / EBI_ALE / I2S1_DI / SC1_DAT / TM4_EXT
47	I/O	PG.6 / EPWM1_CH2 / UART5_RXD / SPI3_MOSI / ECAP0_IC1 / EBI_nRD / I2S1_BCLK / SC1_RST / TM7 / INT3
48	I/O	PG.7 / EPWM1_CH3 / UART5_TXD / SPI3_MISO / ECAP0_IC2 / EBI_nWR / I2S1_LRCK / SC1_PWR / TM7_EXT
49	P	V_{DD_CORE}
50	I/O	PG.11 / JTAG_TDO / I2S0_MCLK / NAND_RDY1 / EBI_nWRH / EBI_nCS1 / EBI_AD0 **Note:** When the chip enters debug mode, this pin will be automatically forced to JTAG functionality by hardware, and other pinout functions of this pin will be disabled.
51	I/O	PG.12 / JTAG_TCK/SW_CLK / I2S0_LRCK / EBI_nWRL / EBI_AD1 **Note:** When the chip enters debug mode, this pin will be automatically forced to JTAG/SWD functionality by hardware, and other pinout functions of this pin will be disabled.
52	I/O	PG.13 / JTAG_TMS/SW_DIO / I2S0_BCLK / EBI_MCLK / EBI_AD2 **Note:** When the chip enters debug mode, this pin will be automatically forced to JTAG/SWD functionality by hardware, and other pinout functions of this pin will be disabled.
53	I/O	PG.14 / JTAG_TDI / I2S0_DI / NAND_nCS1 / EBI_ALE / EBI_AD3 **Note:** When the chip enters debug mode, this pin will be automatically forced to JTAG functionality by hardware, and other pinout functions of this pin will be disabled.

Pin	Type	MA35D14F884 / MA35D14F984 (LQFP-EP 216-Pin, MCP with 256 / 512 MB DDR3L) Pin Function
54	I/O	PG.15 / JTAG_nTRST / I2S0_DO / EBI_nCS0 / EBI_AD4 **Note:** When the chip enters debug mode, this pin will be automatically forced to JTAG functionality by hardware, and other pinout functions of this pin will be disabled.
55	P	V_{DD_CORE}
56	I/O	PE.14 / UART0_TXD
57	I/O	PE.15 / UART0_RXD **Note:** This GPIO PE.15 (UART0_RXD) pin must be pulled to high level through an external 10kΩ resistor or an internal pull-up resistor in the external device (such as a transceiver or MCU).
58	I/O	PD.6 / EPWM0_SYNC_IN / UART1_RXD / QSPI1_MOSI1 / I2C0_SDA / I2S0_MCLK / EPWM0_CH0 / EBI_AD5 / SPI3_SS1 **Note:** This pin is dedicated as I2C0_SDA function for controlling external PMIC by default.
59	I/O	PD.7 / EPWM0_SYNC_OUT / UART1_TXD / QSPI1_MISO1 / I2C0_SCL / I2S1_MCLK / EPWM0_CH1 / EBI_AD6 / SC1_nCD **Note:** This pin is dedicated as I2C0_SCL function for controlling external PMIC by default.
60	I/O	PD.8 / EPWM0_BRAKE0 / UART16_nCTS / UART15_RXD / QSPI1_SS0 / I2S1_LRCK / EPWM0_CH2 / EBI_AD7 / SC1_CLK / TM0
61	I/O	PD.9 / EPWM0_BRAKE1 / UART16_nRTS / UART15_TXD / QSPI1_CLK / I2S1_BCLK / EPWM0_CH3 / EBI_AD8 / SC1_DAT / TM0_EXT
62	I/O	PD.10 / EPWM1_BRAKE0 / UART16_RXD / QSPI1_MOSI0 / I2S1_DI / EPWM0_CH4 / EBI_AD9 / SC1_RST / TM2
63	I/O	PD.11 / EPWM1_BRAKE1 / UART16_TXD / QSPI1_MISO0 / I2S1_DO / EPWM0_CH5 / EBI_AD10 / SC1_PWR / TM2_EXT
64	P	V_{DDIO1}
65	I/O	PK.9 / I2C3_SCL / CCAP0_SCLK / EBI_AD0 / EBI_ADR0
66	I/O	PK.10 / CCAP0_PIXCLK / EBI_AD1 / EBI_ADR1
67	I/O	PK.11 / CCAP0_HSYNC / EBI_AD2 / EBI_ADR2
68	I/O	PM.0 / I2C4_SDA / CCAP0_VSYNC / EBI_AD3 / EBI_ADR3
69	I/O	PM.1 / I2C4_SCL / SPI3_I2SMCLK / CCAP0_SFIELD / EBI_AD4 / EBI_ADR4
70	I/O	PM.2 / CCAP0_DATA0 / EBI_AD5 / EBI_ADR5
71	I/O	PM.3 / CCAP0_DATA1 / EBI_AD6 / EBI_ADR6
72	I/O	PM.4 / I2C5_SDA / CCAP0_DATA2 / EBI_AD7 / EBI_ADR7
73	I/O	PM.5 / I2C5_SCL / CCAP0_DATA3 / EBI_AD8 / EBI_ADR8
74	I/O	PM.6 / CCAP0_DATA4 / EBI_AD9 / EBI_ADR9
75	I/O	PM.7 / CCAP0_DATA5 / EBI_AD10 / EBI_ADR10
76	P	V_{DDIO2}
77	I/O	PM.8 / I2C0_SDA / CCAP0_DATA6 / EBI_AD11 / EBI_ADR11
78	I/O	PM.9 / I2C0_SCL / CCAP0_DATA7 / EBI_AD12 / EBI_ADR12
79	I/O	PM.10 / EPWM1_CH2 / SPI3_SS0 / CCAP0_DATA8 / SPI2_I2SMCLK / EBI_AD13 / EBI_ADR13
80	I/O	PM.11 / EPWM1_CH3 / SPI3_SS1 / CCAP0_DATA9 / SPI2_SS1 / EBI_AD14 / EBI_ADR14

Pin	Type	MA35D14F884 / MA35D14F984 (LQFP-EP 216-Pin, MCP with 256 / 512 MB DDR3L) Pin Function
81	P	V_{DD_CORE}
82	I/O	PJ.4 / I2C3_SDA / SD1_WP
83	I/O	PJ.5 / I2C3_SCL / SD1_nCD **Note:** An external 10kΩ pull-down resistor is necessary on this pin when these PJ.6~PJ.11 pins (eMMC1_xxx) are used to connect with eMMC device and act as the booting source.
84	P	V_{DDIO3}
85	I/O	PJ.6 / SD1_CMD/eMMC1_CMD
86	I/O	PJ.7 / SD1_CLK/eMMC1_CLK
87	I/O	PJ.8 / I2C4_SDA / SD1_DAT0/eMMC1_DAT0
88	I/O	PJ.9 / I2C4_SCL / SD1_DAT1/eMMC1_DAT1
89	I/O	PJ.10 / SD1_DAT2/eMMC1_DAT2
90	I/O	PJ.11 / SD1_DAT3/eMMC1_DAT3
91	I/O	PG.8 / EPWM1_CH4 / UART12_RXD / SPI2_SS0 / LCM_VSYNC/LCM_MPU_RD/EN / I2C3_SDA / EBI_AD7 / EBI_nCS0
92	I/O	PG.9 / EPWM1_CH5 / UART12_TXD / SPI2_CLK / LCM_HSYNC/LCM_MPU_WR/RW / I2C3_SCL / EBI_AD8 / EBI_nCS1
93	I/O	PG.10 / UART12_nRTS / UART13_TXD / SPI2_MOSI / LCM_CLK / EBI_AD9 / EBI_nWRH
94	I/O	PK.4 / UART12_nCTS / UART13_RXD / SPI2_MISO / LCM_DEN/LCM_MPU_RS / EBI_AD10 / EBI_nWRL
95	I/O	PI.8 / UART4_nCTS / UART3_RXD / LCM_DATA0/LCM_MPU_D0 / EBI_AD11
96	I/O	PI.9 / UART4_nRTS / UART3_TXD / LCM_DATA1/LCM_MPU_D1 / EBI_AD12
97	I/O	PI.10 / UART4_RXD / LCM_DATA2/LCM_MPU_D2 / EBI_AD13
98	I/O	PI.11 / UART4_TXD / LCM_DATA3/LCM_MPU_D3 / EBI_AD14
99	I/O	PI.12 / UART6_nCTS / UART5_RXD / LCM_DATA4/LCM_MPU_D4
100	I/O	PI.13 / UART6_nRTS / UART5_TXD / LCM_DATA5/LCM_MPU_D5
101	I/O	PI.14 / UART6_RXD / LCM_DATA6/LCM_MPU_D6
102	I/O	PI.15 / UART6_TXD / LCM_DATA7/LCM_MPU_D7
103	I/O	PH.0 / UART8_nCTS / UART7_RXD / LCM_DATA8/LCM_MPU_D8
104	I/O	PH.1 / UART8_nRTS / UART7_TXD / LCM_DATA9/LCM_MPU_D9
105	I/O	PH.2 / UART8_RXD / LCM_DATA10/LCM_MPU_D10
106	P	V_{DDIO4}
107	I/O	PH.3 / UART8_TXD / LCM_DATA11/LCM_MPU_D11
108	I/O	PH.4 / UART10_nCTS / UART9_RXD / LCM_DATA12/LCM_MPU_D12
109	I/O	PH.5 / UART10_nRTS / UART9_TXD / LCM_DATA13/LCM_MPU_D13
110	I/O	PH.6 / UART10_RXD / LCM_DATA14/LCM_MPU_D14
111	I/O	PH.7 / UART10_TXD / LCM_DATA15/LCM_MPU_D15
112	I/O	PC.12 / UART12_nCTS / UART11_RXD / LCM_DATA16/LCM_MPU_D16

Pin	Type	MA35D14F884 / MA35D14F984 (LQFP-EP 216-Pin, MCP with 256 / 512 MB DDR3L) Pin Function
113	I/O	PC.13 / UART12_nRTS / UART11_TXD / LCM_DATA17/LCM_MPU_D17
114	I/O	PC.14 / UART12_RXD / LCM_DATA18/LCM_MPU_CS
115	I/O	PC.15 / UART12_TXD / LCM_DATA19 / LCM_MPU_TE / LCM_MPU_VSYNC
116	I/O	PH.12 / UART14_nCTS / UART13_RXD / LCM_DATA20
117	I/O	PH.13 / UART14_nRTS / UART13_TXD / LCM_DATA21
118	I/O	PH.14 / UART14_RXD / LCM_DATA22
119	I/O	PH.15 / UART14_TXD / LCM_DATA23
120	P	V_{DD_CORE}
121	P	V_{DDIO5}
122	I/O	PD.0 / UART3_nCTS / UART4_RXD / QSPI0_SS0
123	I/O	PD.1 / UART3_nRTS / UART4_TXD / QSPI0_CLK
124	I/O	PD.2 / UART3_RXD / QSPI0_MOSI0
125	I/O	PD.3 / UART3_TXD / QSPI0_MISO0
126	I/O	PD.4 / UART1_nCTS / UART2_RXD / I2C2_SDA / QSPI0_MOSI1
127	I/O	PD.5 / UART1_nRTS / UART2_TXD / I2C2_SCL / QSPI0_MISO1
128	P	V_{SS}
129	P	MV_{DD}
130	P	MV_{DD}
131	P	V_{DD_CORE}
132	P	$MV_{DD_DPHYPLL}$
133	P	MZQ_SDRAM
134	P	MZQ_DDRPHY
135	P	MV_{DD}
136	P	MV_{REF}
137	P	V_{DD_CORE}
138	P	MV_{DD}
139	P	$MV_{DD_DPHYPLL}$
140	P	V_{DD_CORE}
141	P	MV_{DD}
142	I/O	PC.0 / I2C4_SDA / SD0_CMD/eMMC0_CMD
143	I/O	PC.1 / I2C4_SCL / SD0_CLK/eMMC0_CLK
144	I/O	PC.2 / SD0_DAT0/eMMC0_DAT0
145	I/O	PC.3 / SD0_DAT1/eMMC0_DAT1
146	I/O	PC.4 / I2C5_SDA / SD0_DAT2/eMMC0_DAT2

Pin	Type	MA35D14F884 / MA35D14F984 (LQFP-EP 216-Pin, MCP with 256 / 512 MB DDR3L) Pin Function
147	I/O	PC.5 / I2C5_SCL / SD0_DAT3/eMMC0_DAT3
148	P	V_{DDIO6}
149	I/O	PC.6 / SD0_nCD **Note:** An external 10kΩ pull-down resistor is necessary on this pin when these PC.0~PC.5 pins (eMMC0_xxx) are used to connect with eMMC device and act as the booting source.
150	I/O	PC.7 / SD0_WP
151	P	V_{DD_CORE}
152	I/O	PN.0 / I2C2_SDA / CCAP1_DATA0
153	I/O	PN.1 / I2C2_SCL / CCAP1_DATA1
154	I/O	PN.2 / CCAP1_DATA2
155	I/O	PN.3 / CCAP1_DATA3
156	I/O	PN.4 / I2C1_SDA / CCAP1_DATA4
157	I/O	PN.5 / I2C1_SCL / CCAP1_DATA5
158	P	V_{DDIO7}
159	I/O	PN.6 / CCAP1_DATA6
160	I/O	PN.7 / CCAP1_DATA7
161	I/O	PN.10 / CCAP1_SCLK
162	I/O	PN.11 / CCAP1_PIXCLK
163	I/O	PN.12 / UART6_nCTS / UART12_RXD / I2C5_SDA / CCAP1_HSYNC
164	I/O	PN.13 / UART6_nRTS / UART12_TXD / I2C5_SCL / CCAP1_VSYNC
165	I/O	PN.14 / UART6_RXD / SPI1_SS1 / CCAP1_SFIELD / SPI1_I2SMCLK
166	I/O	PN.15 / EPWM2_CH4 / UART6_TXD / I2S0_MCLK / SPI1_SS1 / SPI1_I2SMCLK / SC0_nCD / CLKO / TM6
167	I/O	PK.12 / EPWM2_CH0 / UART1_nCTS / UART13_RXD / I2C4_SDA / I2S0_LRCK / SPI1_SS0 / SC0_CLK / TM10 / INT2
168	P	AV_{DDL_PLL0}
169	I	XT1_IN
170	O	XT1_OUT
171	P	V_{DDIO10}
172	P	AV_{DDL_PLL1}
173	P	V_{DD_PLL1}
174	P	AV_{DDH_PLL2}
175	I/O	PE.0 / UART9_nCTS / UART8_RXD / CCAP1_DATA0 / RGMII0_MDC / RMII0_MDC
176	I/O	PE.1 / UART9_nRTS / UART8_TXD / CCAP1_DATA1 / RGMII0_MDIO / RMII0_MDIO
177	I/O	PE.2 / UART9_RXD / CCAP1_DATA2 / RGMII0_TXCTL / RMII0_TXEN
178	I/O	PE.3 / UART9_TXD / CCAP1_DATA3 / RGMII0_TXD0 / RMII0_TXD0

Pin	Type	MA35D14F884 / MA35D14F984 (LQFP-EP 216-Pin, MCP with 256 / 512 MB DDR3L) Pin Function
179	P	V_{DDIO8}
180	P	V_{DD_CORE}
181	I/O	PE.4 / UART4_nCTS / UART3_RXD / CCAP1_DATA4 / RGMII0_TXD1 / RMII0_TXD1
182	I/O	PE.5 / UART4_nRTS / UART3_TXD / CCAP1_DATA5 / RGMII0_RXCLK / RMII0_REFCLK
183	I/O	PE.6 / UART4_RXD / CCAP1_DATA6 / RGMII0_RXCTL / RMII0_CRSDV
184	I/O	PE.7 / UART4_TXD / CCAP1_DATA7 / RGMII0_RXD0 / RMII0_RXD0
185	I/O	PE.8 / UART13_nCTS / UART12_RXD / CCAP1_SCLK / RGMII0_RXD1 / RMII0_RXD1
186	I/O	PE.9 / UART13_nRTS / UART12_TXD / CCAP1_PIXCLK / RGMII0_RXD2 / RMII0_RXERR
187	I/O	PE.10 / UART15_nCTS / UART14_RXD / SPI1_SS0 / CCAP1_HSYNC / RGMII0_RXD3
188	I/O	PE.11 / UART15_nRTS / UART14_TXD / SPI1_CLK / CCAP1_VSYNC / RGMII0_TXCLK
189	I/O	PE.12 / UART15_RXD / SPI1_MOSI / CCAP1_DATA8 / RGMII0_TXD2
190	I/O	PE.13 / UART15_TXD / SPI1_MISO / CCAP1_DATA9 / RGMII0_TXD3
191	P	V_{DDIO9}
192	I/O	PF.0 / UART2_nCTS / UART1_RXD / RGMII0_RXD3 / RMII1_MDC / KPI_COL0
193	I/O	PF.1 / UART2_nRTS / UART1_TXD / RGMII0_TXCLK / RMII1_MDIO / KPI_COL1
194	I/O	PF.2 / UART2_RXD / RGMII0_TXD2 / RMII1_TXEN / KPI_COL2
195	I/O	PF.3 / UART2_TXD / RGMII0_TXD3 / RMII1_TXD0 / KPI_COL3
196	I/O	PF.4 / UART11_nCTS / UART10_RXD / I2S0_LRCK / SPI1_SS0 / RMII1_TXD1 / KPI_ROW0
197	I/O	PF.5 / UART11_nRTS / UART10_TXD / I2S0_BCLK / SPI1_CLK / RMII1_REFCLK / KPI_ROW1
198	I/O	PF.6 / UART11_RXD / I2S0_DI / SPI1_MOSI / RMII1_CRSDV / I2C4_SDA / SC0_CLK / KPI_ROW2
199	I/O	PF.7 / UART11_TXD / I2S0_DO / SPI1_MISO / RMII1_RXD0 / I2C4_SCL / SC0_DAT / KPI_ROW3
200	I/O	PF.8 / UART13_RXD / I2C5_SDA / SPI0_SS0 / RMII1_RXD1 / SC0_RST / KPI_COL4
201	I/O	PF.9 / UART13_TXD / I2C5_SCL / SPI0_SS1 / RMII1_RXERR / SC0_PWR / KPI_COL5
202	P	V_{DD_CORE}
203	P	AV_{DD_EADC0}
204	I/O	PB.8 / EPWM2_BRAKE0 / UART2_nCTS / UART1_RXD / I2C2_SDA / SPI0_SS1 / SPI0_I2SMCLK / ADC0_CH0 / EBI_nCS0 / TM4 / QEI2_INDEX **Note:** When this pin be chosen as ADC0_CH0 pinout function, it has another function which is the reference voltage (V_{REF}) input of ADC0 by setting the control register REFSEL (ADC_CONF[7:6]) to 2'b00. In this case, it needs to provide a reference voltage for ADC0 on this pin and the ADC0 channel 0 analog input function will be disabled.
205	I/O	PB.9 / EPWM2_CH4 / UART2_nRTS / UART1_TXD / I2C2_SCL / SPI0_CLK / I2S0_MCLK / CCAP1_HSYNC / ADC0_CH1 / EBI_ALE / EBI_AD13 / TM0_EXT / I2S1_MCLK / SC0_nCD / QEI2_A
206	I/O	PB.10 / EPWM2_CH5 / UART2_RXD / SPI0_MOSI / EBI_MCLK / CCAP1_VSYNC / ADC0_CH2 / EBI_ADR15 / EBI_AD14 / TM5 / I2C1_SDA / INT1 / QEI2_B
207	I/O	PB.11 / EPWM2_BRAKE1 / UART2_TXD / SPI0_MISO / I2S1_MCLK / CCAP1_SFIELD / ADC0_CH3 / EBI_nCS2 / EBI_ALE / TM5_EXT / I2C1_SCL / INT2 / QEI2_INDEX
208	P	AV_{DD_ADC0}

Pin	Type	MA35D14F884 / MA35D14F984 (LQFP-EP 216-Pin, MCP with 256 / 512 MB DDR3L) Pin Function
209	P	AV_{SS}
210	P	V_{DD_CORE}
211	I/O	PB.12 / EPWM2_CH0 / UART4_nCTS / UART3_RXD / I2C3_SDA / I2S1_LRCK / ADC0_CH4 / EBI_ADR16 / ECAP2_IC0
212	I/O	PB.13 / EPWM2_CH1 / UART4_nRTS / UART3_TXD / I2C3_SCL / I2S1_BCLK / ADC0_CH5 / EBI_ADR17 / ECAP2_IC1
213	I/O	PB.14 / EPWM2_CH2 / UART4_RXD / I2C4_SDA / I2S1_DI / ADC0_CH6 / EBI_ADR18 / ECAP2_IC2
214	I/O	PB.15 / EPWM2_CH3 / UART4_TXD / I2C4_SCL / I2S1_DO / ADC0_CH7 / EBI_ADR19
215	P	V_{DDIO0}
216	P	V_{DD_CORE}
EPAD	P	V_{SS} **Note:** When soldering SMD, please make sure the EPAD on the bottom of LQFP-EP 216-Pin package is tightly soldered to the ground of the PCB.

4.2.1.2 *MA35D16F787C / MA35D16F8(I)87C / MA35D16F9(J)87C Series*

Pin	Type	MA35D16F787C / MA35D16F8(I)87C / MA35D16F9(J)87C (LQFP-EP 216-Pin, MCP with 128 / 256 / 512 MB DDR) Pin Function
1	I	nRESET / WDT_nRST
2	I	HSUSB0_ID
3	I/O	PF.15 / HSUSB0_VBUSVLD
4	I/O	PL.12 / EPWM0_SYNC_IN / UART7_nCTS / ECAP1_IC0 / UART14_RXD / SPI0_SS0 / I2S1_LRCK / SC1_CLK / EBI_AD0 / HSUSBH_PWREN / I2C2_SDA / TM0 / EPWM0_CH2 / EBI_AD11 / RGMII0_PPS / RMII0_PPS
5	I/O	PL.13 / EPWM0_SYNC_OUT / UART7_nRTS / ECAP1_IC1 / UART14_TXD / SPI0_CLK / I2S1_BCLK / SC1_DAT / EBI_AD1 / HSUSBH_OVC / I2C2_SCL / TM0_EXT / EPWM0_CH3 / EBI_AD12 / RMII1_PPS
6	I/O	PL.14 / EPWM0_CH2 / UART7_RXD / CAN1_RXD / SPI0_MOSI / I2S1_DI / SC1_RST / EBI_AD2 / TM2 / INT0 / EBI_AD13
7	I/O	PL.15 / EPWM0_CH1 / UART7_TXD / CAN1_TXD / SPI0_MISO / I2S1_DO / SC1_PWR / EBI_AD3 / TM2_EXT / INT2 / EBI_AD14
8	P	V_{DD_OTP}
9	O	RTC_RPWR
10	P	V_{BAT}
11	I	RTC_nRWAKE
12	I	X32_IN
13	O	X32_OUT
14	P	V_{DD_HSUSB1}
15	A	HSUSB1_D-
16	A	HSUSB1_D+
17	P	V_{DD_CORE}
18	P	V_{DD_HSUSB1}
19	A	HSUSB0_D-
20	A	HSUSB0_D+
21	P	V_{DD_CORE}
22	P	V_{DD_HSUSB0}
23	P	V_{DD_CORE}
24	I/O	PA.0 / UART1_nCTS / UART16_RXD / NAND_DATA0 / EBI_AD0 / EBI_ADR0
25	I/O	PA.1 / UART1_nRTS / UART16_TXD / NAND_DATA1 / EBI_AD1 / EBI_ADR1
26	I/O	PA.2 / UART1_RXD / NAND_DATA2 / EBI_AD2 / EBI_ADR2
27	I/O	PA.3 / UART1_TXD / NAND_DATA3 / EBI_AD3 / EBI_ADR3
28	I/O	PA.4 / UART3_nCTS / UART2_RXD / NAND_DATA4 / EBI_AD4 / EBI_ADR4
29	I/O	PA.5 / UART3_nRTS / UART2_TXD / NAND_DATA5 / EBI_AD5 / EBI_ADR5
30	I/O	PA.6 / UART3_RXD / NAND_DATA6 / EBI_AD6 / EBI_ADR6

Pin	Type	MA35D16F787C / MA35D16F8(I)87C / MA35D16F9(J)87C (LQFP-EP 216-Pin, MCP with 128 / 256 / 512 MB DDR) Pin Function
31	I/O	PA.7 / UART3_TXD / NAND_DATA7 / EBI_AD7 / EBI_ADR7
32	I/O	PA.8 / UART5_nCTS / UART4_RXD / NAND_RDY0 / EBI_AD8 / EBI_ADR8
33	I/O	PA.9 / UART5_nRTS / UART4_TXD / NAND_nRE / EBI_AD9 / EBI_ADR9
34	I/O	PA.10 / UART5_RXD / NAND_nWE / EBI_AD10 / EBI_ADR10
35	I/O	PA.11 / UART5_TXD / NAND_CLE / EBI_AD11 / EBI_ADR11
36	I/O	PA.12 / UART7_nCTS / UART8_RXD / NAND_ALE / EBI_AD12 / EBI_ADR12
37	I/O	PA.13 / UART7_nRTS / UART8_TXD / NAND_nCS0 / EBI_AD13 / EBI_ADR13
38	P	V_{DDIO1}
39	I/O	PA.14 / UART7_RXD / CAN3_RXD / NAND_nWP / EBI_AD14 / EBI_ADR14
40	I/O	PG.0 / EPWM0_CH0 / UART7_TXD / CAN3_TXD / SPI0_SS0 / EBI_AD15 / I2S1_MCLK / QEI0_INDEX / TM1 / CLKO / INT0 / EBI_ADR15
41	I/O	PA.15 / EPWM0_CH2 / UART9_nCTS / UART6_RXD / I2C4_SDA / CAN2_RXD / EBI_ALE / QEI0_A / TM1 / RGMII0_PPS / RMII0_PPS **Note:** This pin is dedicated as PMIC_nIRQ input function from external PMIC by default.
42	I/O	PG.1 / EPWM0_CH3 / UART9_nRTS / UART6_TXD / I2C4_SCL / CAN2_TXD / EBI_nCS0 / QEI0_B / TM1_EXT / RMII1_PPS
43	I/O	PG.2 / EPWM0_CH4 / UART9_RXD / CAN0_RXD / SPI0_SS1 / EBI_ADR16 / EBI_nCS2 / QEI0_A / TM3 / INT1
44	I/O	PG.3 / EPWM0_CH5 / UART9_TXD / CAN0_TXD / SPI0_I2SMCLK / EBI_ADR17 / EBI_nCS1 / EBI_MCLK / QEI0_B / TM3_EXT / I2S1_MCLK
45	I/O	PG.4 / EPWM1_CH0 / UART5_nCTS / UART6_RXD / SPI3_SS0 / EBI_ADR18 / EBI_nCS0 / I2S1_DO / SC1_CLK / TM4 / INT2 / ECAP1_IC2
46	I/O	PG.5 / EPWM1_CH1 / UART5_nRTS / UART6_TXD / SPI3_CLK / ECAP0_IC0 / EBI_ADR19 / EBI_ALE / I2S1_DI / SC1_DAT / TM4_EXT
47	I/O	PG.6 / EPWM1_CH2 / UART5_RXD / CAN1_RXD / SPI3_MOSI / ECAP0_IC1 / EBI_nRD / I2S1_BCLK / SC1_RST / TM7 / INT3
48	I/O	PG.7 / EPWM1_CH3 / UART5_TXD / CAN1_TXD / SPI3_MISO / ECAP0_IC2 / EBI_nWR / I2S1_LRCK / SC1_PWR / TM7_EXT
49	P	V_{DD_CORE}
50	I/O	PG.11 / JTAG_TDO / I2S0_MCLK / NAND_RDY1 / EBI_nWRH / EBI_nCS1 / EBI_AD0 **Note:** When the chip enters debug mode, this pin will be automatically forced to JTAG functionality by hardware, and other pinout functions of this pin will be disabled.
51	I/O	PG.12 / JTAG_TCK/SW_CLK / I2S0_LRCK / EBI_nWRL / EBI_AD1 **Note:** When the chip enters debug mode, this pin will be automatically forced to JTAG/SWD functionality by hardware, and other pinout functions of this pin will be disabled.
52	I/O	PG.13 / JTAG_TMS/SW_DIO / I2S0_BCLK / EBI_MCLK / EBI_AD2 **Note:** When the chip enters debug mode, this pin will be automatically forced to JTAG/SWD functionality by hardware, and other pinout functions of this pin will be disabled.
53	I/O	PG.14 / JTAG_TDI / I2S0_DI / NAND_nCS1 / EBI_ALE / EBI_AD3 **Note:** When the chip enters debug mode, this pin will be automatically forced to JTAG functionality by hardware, and other pinout functions of this pin will be disabled.
54	I/O	PG.15 / JTAG_nTRST / I2S0_DO / EBI_nCS0 / EBI_AD4

Pin	Type	MA35D16F787C / MA35D16F8(I)87C / MA35D16F9(J)87C (LQFP-EP 216-Pin, MCP with 128 / 256 / 512 MB DDR) Pin Function
		Note: When the chip enters debug mode, this pin will be automatically forced to JTAG functionality by hardware, and other pinout functions of this pin will be disabled.
55	P	V_{DD_CORE}
56	I/O	PE.14 / UART0_TXD
57	I/O	PE.15 / UART0_RXD **Note:** This GPIO PE.15 (UART0_RXD) pin must be pulled to high level through an external 10kΩ resistor or an internal pull-up resistor in the external device (such as a transceiver or MCU).
58	I/O	PD.6 / EPWM0_SYNC_IN / UART1_RXD / QSPI1_MOSI1 / I2C0_SDA / I2S0_MCLK / EPWM0_CH0 / EBI_AD5 / SPI3_SS1 **Note:** This pin is dedicated as I2C0_SDA function for controlling external PMIC by default.
59	I/O	PD.7 / EPWM0_SYNC_OUT / UART1_TXD / QSPI1_MISO1 / I2C0_SCL / I2S1_MCLK / EPWM0_CH1 / EBI_AD6 / SC1_nCD **Note:** This pin is dedicated as I2C0_SCL function for controlling external PMIC by default.
60	I/O	PD.8 / EPWM0_BRAKE0 / UART16_nCTS / UART15_RXD / QSPI1_SS0 / I2S1_LRCK / EPWM0_CH2 / EBI_AD7 / SC1_CLK / TM0
61	I/O	PD.9 / EPWM0_BRAKE1 / UART16_nRTS / UART15_TXD / QSPI1_CLK / I2S1_BCLK / EPWM0_CH3 / EBI_AD8 / SC1_DAT / TM0_EXT
62	I/O	PD.10 / EPWM1_BRAKE0 / UART16_RXD / QSPI1_MOSI0 / I2S1_DI / EPWM0_CH4 / EBI_AD9 / SC1_RST / TM2
63	I/O	PD.11 / EPWM1_BRAKE1 / UART16_TXD / QSPI1_MISO0 / I2S1_DO / EPWM0_CH5 / EBI_AD10 / SC1_PWR / TM2_EXT
64	P	V_{DDIO1}
65	I/O	PK.9 / I2C3_SCL / CCAP0_SCLK / EBI_AD0 / EBI_ADR0
66	I/O	PK.10 / CAN1_RXD / CCAP0_PIXCLK / EBI_AD1 / EBI_ADR1
67	I/O	PK.11 / CAN1_TXD / CCAP0_HSYNC / EBI_AD2 / EBI_ADR2
68	I/O	PM.0 / I2C4_SDA / CCAP0_VSYNC / EBI_AD3 / EBI_ADR3
69	I/O	PM.1 / I2C4_SCL / SPI3_I2SMCLK / CCAP0_SFIELD / EBI_AD4 / EBI_ADR4
70	I/O	PM.2 / CAN3_RXD / CCAP0_DATA0 / EBI_AD5 / EBI_ADR5
71	I/O	PM.3 / CAN3_TXD / CCAP0_DATA1 / EBI_AD6 / EBI_ADR6
72	I/O	PM.4 / I2C5_SDA / CCAP0_DATA2 / EBI_AD7 / EBI_ADR7
73	I/O	PM.5 / I2C5_SCL / CCAP0_DATA3 / EBI_AD8 / EBI_ADR8
74	I/O	PM.6 / CAN0_RXD / CCAP0_DATA4 / EBI_AD9 / EBI_ADR9
75	I/O	PM.7 / CAN0_TXD / CCAP0_DATA5 / EBI_AD10 / EBI_ADR10
76	P	V_{DDIO2}
77	I/O	PM.8 / I2C0_SDA / CCAP0_DATA6 / EBI_AD11 / EBI_ADR11
78	I/O	PM.9 / I2C0_SCL / CCAP0_DATA7 / EBI_AD12 / EBI_ADR12
79	I/O	PM.10 / EPWM1_CH2 / CAN2_RXD / SPI3_SS0 / CCAP0_DATA8 / SPI2_I2SMCLK / EBI_AD13 / EBI_ADR13
80	I/O	PM.11 / EPWM1_CH3 / CAN2_TXD / SPI3_SS1 / CCAP0_DATA9 / SPI2_SS1 / EBI_AD14 / EBI_ADR14

Pin	Type	MA35D16F787C / MA35D16F8(I)87C / MA35D16F9(J)87C (LQFP-EP 216-Pin, MCP with 128 / 256 / 512 MB DDR) Pin Function
81	P	V_{DD_CORE}
82	I/O	PJ.4 / I2C3_SDA / SD1_WP
83	I/O	PJ.5 / I2C3_SCL / SD1_nCD **Note:** An external 10kΩ pull-down resistor is necessary on this pin when these PJ.6~PJ.11 pins (eMMC1_xxx) are used to connect with eMMC device and act as the booting source.
84	P	V_{DDIO3}
85	I/O	PJ.6 / CAN3_RXD / SD1_CMD/eMMC1_CMD
86	I/O	PJ.7 / CAN3_TXD / SD1_CLK/eMMC1_CLK
87	I/O	PJ.8 / I2C4_SDA / SD1_DAT0/eMMC1_DAT0
88	I/O	PJ.9 / I2C4_SCL / SD1_DAT1/eMMC1_DAT1
89	I/O	PJ.10 / CAN0_RXD / SD1_DAT2/eMMC1_DAT2
90	I/O	PJ.11 / CAN0_TXD / SD1_DAT3/eMMC1_DAT3
91	I/O	PG.8 / EPWM1_CH4 / UART12_RXD / CAN3_RXD / SPI2_SS0 / LCM_VSYNC/LCM_MPU_RD/EN / I2C3_SDA / EBI_AD7 / EBI_nCS0
92	I/O	PG.9 / EPWM1_CH5 / UART12_TXD / CAN3_TXD / SPI2_CLK / LCM_HSYNC/LCM_MPU_WR/RW / I2C3_SCL / EBI_AD8 / EBI_nCS1
93	I/O	PG.10 / UART12_nRTS / UART13_TXD / SPI2_MOSI / LCM_CLK / EBI_AD9 / EBI_nWRH
94	I/O	PK.4 / UART12_nCTS / UART13_RXD / SPI2_MISO / LCM_DEN/LCM_MPU_RS / EBI_AD10 / EBI_nWRL
95	I/O	PI.8 / UART4_nCTS / UART3_RXD / LCM_DATA0/LCM_MPU_D0 / EBI_AD11
96	I/O	PI.9 / UART4_nRTS / UART3_TXD / LCM_DATA1/LCM_MPU_D1 / EBI_AD12
97	I/O	PI.10 / UART4_RXD / LCM_DATA2/LCM_MPU_D2 / EBI_AD13
98	I/O	PI.11 / UART4_TXD / LCM_DATA3/LCM_MPU_D3 / EBI_AD14
99	I/O	PI.12 / UART6_nCTS / UART5_RXD / LCM_DATA4/LCM_MPU_D4
100	I/O	PI.13 / UART6_nRTS / UART5_TXD / LCM_DATA5/LCM_MPU_D5
101	I/O	PI.14 / UART6_RXD / LCM_DATA6/LCM_MPU_D6
102	I/O	PI.15 / UART6_TXD / LCM_DATA7/LCM_MPU_D7
103	I/O	PH.0 / UART8_nCTS / UART7_RXD / LCM_DATA8/LCM_MPU_D8
104	I/O	PH.1 / UART8_nRTS / UART7_TXD / LCM_DATA9/LCM_MPU_D9
105	I/O	PH.2 / UART8_RXD / LCM_DATA10/LCM_MPU_D10
106	P	V_{DDIO4}
107	I/O	PH.3 / UART8_TXD / LCM_DATA11/LCM_MPU_D11
108	I/O	PH.4 / UART10_nCTS / UART9_RXD / LCM_DATA12/LCM_MPU_D12
109	I/O	PH.5 / UART10_nRTS / UART9_TXD / LCM_DATA13/LCM_MPU_D13
110	I/O	PH.6 / UART10_RXD / LCM_DATA14/LCM_MPU_D14
111	I/O	PH.7 / UART10_TXD / LCM_DATA15/LCM_MPU_D15

Pin	Type	MA35D16F787C / MA35D16F8(I)87C / MA35D16F9(J)87C (LQFP-EP 216-Pin, MCP with 128 / 256 / 512 MB DDR) Pin Function
112	I/O	PC.12 / UART12_nCTS / UART11_RXD / LCM_DATA16/LCM_MPU_D16
113	I/O	PC.13 / UART12_nRTS / UART11_TXD / LCM_DATA17/LCM_MPU_D17
114	I/O	PC.14 / UART12_RXD / LCM_DATA18/LCM_MPU_CS
115	I/O	PC.15 / UART12_TXD / LCM_DATA19 / LCM_MPU_TE / LCM_MPU_VSYNC
116	I/O	PH.12 / UART14_nCTS / UART13_RXD / LCM_DATA20
117	I/O	PH.13 / UART14_nRTS / UART13_TXD / LCM_DATA21
118	I/O	PH.14 / UART14_RXD / LCM_DATA22
119	I/O	PH.15 / UART14_TXD / LCM_DATA23
120	P	V_{DD_CORE}
121	P	V_{DDIO5}
122	I/O	PD.0 / UART3_nCTS / UART4_RXD / QSPI0_SS0
123	I/O	PD.1 / UART3_nRTS / UART4_TXD / QSPI0_CLK
124	I/O	PD.2 / UART3_RXD / QSPI0_MOSI0
125	I/O	PD.3 / UART3_TXD / QSPI0_MISO0
126	I/O	PD.4 / UART1_nCTS / UART2_RXD / I2C2_SDA / QSPI0_MOSI1
127	I/O	PD.5 / UART1_nRTS / UART2_TXD / I2C2_SCL / QSPI0_MISO1
128	P	V_{SS}
129	P	MV_{DD}
130	P	MV_{DD}
131	P	V_{DD_CORE}
132	P	$MV_{DD_DPHYPLL}$
133	P	MZQ_SDRAM
134	P	MZQ_DDRPHY
135	P	MV_{DD}
136	P	MV_{REF}
137	P	V_{DD_CORE}
138	P	MV_{DD}
139	P	$MV_{DD_DPHYPLL}$
140	P	V_{DD_CORE}
141	P	MV_{DD}
142	I/O	PC.0 / I2C4_SDA / SD0_CMD/eMMC0_CMD
143	I/O	PC.1 / I2C4_SCL / SD0_CLK/eMMC0_CLK
144	I/O	PC.2 / CAN0_RXD / SD0_DAT0/eMMC0_DAT0
145	I/O	PC.3 / CAN0_TXD / SD0_DAT1/eMMC0_DAT1

Pin	Type	MA35D16F787C / MA35D16F8(I)87C / MA35D16F9(J)87C (LQFP-EP 216-Pin, MCP with 128 / 256 / 512 MB DDR) Pin Function
146	I/O	PC.4 / I2C5_SDA / SD0_DAT2/eMMC0_DAT2
147	I/O	PC.5 / I2C5_SCL / SD0_DAT3/eMMC0_DAT3
148	P	V_{DDIO6}
149	I/O	PC.6 / CAN1_RXD / SD0_nCD **Note:** An external 10kΩ pull-down resistor is necessary on this pin when these PC.0~PC.5 pins (eMMC0_xxx) are used to connect with eMMC device and act as the booting source.
150	I/O	PC.7 / CAN1_TXD / SD0_WP
151	P	V_{DD_CORE}
152	I/O	PN.0 / I2C2_SDA / CCAP1_DATA0
153	I/O	PN.1 / I2C2_SCL / CCAP1_DATA1
154	I/O	PN.2 / CAN0_RXD / CCAP1_DATA2
155	I/O	PN.3 / CAN0_TXD / CCAP1_DATA3
156	I/O	PN.4 / I2C1_SDA / CCAP1_DATA4
157	I/O	PN.5 / I2C1_SCL / CCAP1_DATA5
158	P	V_{DDIO7}
159	I/O	PN.6 / CAN1_RXD / CCAP1_DATA6
160	I/O	PN.7 / CAN1_TXD / CCAP1_DATA7
161	I/O	PN.10 / CAN2_RXD / CCAP1_SCLK
162	I/O	PN.11 / CAN2_TXD / CCAP1_PIXCLK
163	I/O	PN.12 / UART6_nCTS / UART12_RXD / I2C5_SDA / CCAP1_HSYNC
164	I/O	PN.13 / UART6_nRTS / UART12_TXD / I2C5_SCL / CCAP1_VSYNC
165	I/O	PN.14 / UART6_RXD / CAN3_RXD / SPI1_SS1 / CCAP1_SFIELD / SPI1_I2SMCLK
166	I/O	PN.15 / EPWM2_CH4 / UART6_TXD / CAN3_TXD / I2S0_MCLK / SPI1_SS1 / SPI1_I2SMCLK / SC0_nCD / CLKO / TM6
167	I/O	PK.12 / EPWM2_CH0 / UART1_nCTS / UART13_RXD / I2C4_SDA / I2S0_LRCK / SPI1_SS0 / SC0_CLK / TM10 / INT2
168	P	AV_{DDL_PLL0}
169	I	XT1_IN
170	O	XT1_OUT
171	P	V_{DDIO10}
172	P	AV_{DDL_PLL1}
173	P	V_{DD_PLL1}
174	P	AV_{DDH_PLL2}
175	I/O	PE.0 / UART9_nCTS / UART8_RXD / CCAP1_DATA0 / RGMII0_MDC / RMII0_MDC
176	I/O	PE.1 / UART9_nRTS / UART8_TXD / CCAP1_DATA1 / RGMII0_MDIO / RMII0_MDIO
177	I/O	PE.2 / UART9_RXD / CCAP1_DATA2 / RGMII0_TXCTL / RMII0_TXEN

Pin	Type	MA35D16F787C / MA35D16F8(I)87C / MA35D16F9(J)87C (LQFP-EP 216-Pin, MCP with 128 / 256 / 512 MB DDR) Pin Function
178	I/O	PE.3 / UART9_TXD / CCAP1_DATA3 / RGMII0_TXD0 / RMII0_TXD0
179	P	V_{DDIO8}
180	P	V_{DD_CORE}
181	I/O	PE.4 / UART4_nCTS / UART3_RXD / CCAP1_DATA4 / RGMII0_TXD1 / RMII0_TXD1
182	I/O	PE.5 / UART4_nRTS / UART3_TXD / CCAP1_DATA5 / RGMII0_RXCLK / RMII0_REFCLK
183	I/O	PE.6 / UART4_RXD / CCAP1_DATA6 / RGMII0_RXCTL / RMII0_CRSDV
184	I/O	PE.7 / UART4_TXD / CCAP1_DATA7 / RGMII0_RXD0 / RMII0_RXD0
185	I/O	PE.8 / UART13_nCTS / UART12_RXD / CCAP1_SCLK / RGMII0_RXD1 / RMII0_RXD1
186	I/O	PE.9 / UART13_nRTS / UART12_TXD / CCAP1_PIXCLK / RGMII0_RXD2 / RMII0_RXERR
187	I/O	PE.10 / UART15_nCTS / UART14_RXD / SPI1_SS0 / CCAP1_HSYNC / RGMII0_RXD3
188	I/O	PE.11 / UART15_nRTS / UART14_TXD / SPI1_CLK / CCAP1_VSYNC / RGMII0_TXCLK
189	I/O	PE.12 / UART15_RXD / SPI1_MOSI / CCAP1_DATA8 / RGMII0_TXD2
190	I/O	PE.13 / UART15_TXD / SPI1_MISO / CCAP1_DATA9 / RGMII0_TXD3
191	P	V_{DDIO9}
192	I/O	PF.0 / UART2_nCTS / UART1_RXD / RGMII0_RXD3 / RMII1_MDC / KPI_COL0
193	I/O	PF.1 / UART2_nRTS / UART1_TXD / RGMII0_TXCLK / RMII1_MDIO / KPI_COL1
194	I/O	PF.2 / UART2_RXD / RGMII0_TXD2 / RMII1_TXEN / KPI_COL2
195	I/O	PF.3 / UART2_TXD / RGMII0_TXD3 / RMII1_TXD0 / KPI_COL3
196	I/O	PF.4 / UART11_nCTS / UART10_RXD / I2S0_LRCK / SPI1_SS0 / RMII1_TXD1 / CAN2_RXD / KPI_ROW0
197	I/O	PF.5 / UART11_nRTS / UART10_TXD / I2S0_BCLK / SPI1_CLK / RMII1_REFCLK / CAN2_TXD / KPI_ROW1
198	I/O	PF.6 / UART11_RXD / I2S0_DI / SPI1_MOSI / RMII1_CRSDV / I2C4_SDA / SC0_CLK / KPI_ROW2
199	I/O	PF.7 / UART11_TXD / I2S0_DO / SPI1_MISO / RMII1_RXD0 / I2C4_SCL / SC0_DAT / KPI_ROW3
200	I/O	PF.8 / UART13_RXD / I2C5_SDA / SPI0_SS0 / RMII1_RXD1 / SC0_RST / KPI_COL4
201	I/O	PF.9 / UART13_TXD / I2C5_SCL / SPI0_SS1 / RMII1_RXERR / SC0_PWR / KPI_COL5
202	P	V_{DD_CORE}
203	P	AV_{DD_EADC0}
204	I/O	PB.8 / EPWM2_BRAKE0 / UART2_nCTS / UART1_RXD / I2C2_SDA / SPI0_SS1 / SPI0_I2SMCLK / ADC0_CH0 / EBI_nCS0 / TM4 / QEI2_INDEX **Note:** When this pin be chosen as ADC0_CH0 pinout function, it has another function which is the reference voltage (V_{REF}) input of ADC0 by setting the control register REFSEL (ADC_CONF[7:6]) to 2'b00. In this case, it needs to provide a reference voltage for ADC0 on this pin and the ADC0 channel 0 analog input function will be disabled.
205	I/O	PB.9 / EPWM2_CH4 / UART2_nRTS / UART1_TXD / I2C2_SCL / SPI0_CLK / I2S0_MCLK / CCAP1_HSYNC / ADC0_CH1 / EBI_ALE / EBI_AD13 / TM0_EXT / I2S1_MCLK / SC0_nCD / QEI2_A
206	I/O	PB.10 / EPWM2_CH5 / UART2_RXD / CAN0_RXD / SPI0_MOSI / EBI_MCLK / CCAP1_VSYNC / ADC0_CH2 / EBI_ADR15 / EBI_AD14 / TM5 / I2C1_SDA / INT1 / QEI2_B

Pin	Type	MA35D16F787C / MA35D16F8(I)87C / MA35D16F9(J)87C (LQFP-EP 216-Pin, MCP with 128 / 256 / 512 MB DDR) Pin Function
207	I/O	PB.11 / EPWM2_BRAKE1 / UART2_TXD / CAN0_TXD / SPI0_MISO / I2S1_MCLK / CCAP1_SFIELD / ADC0_CH3 / EBI_nCS2 / EBI_ALE / TM5_EXT / I2C1_SCL / INT2 / QEI2_INDEX
208	P	AV_{DD_ADC0}
209	P	AV_{SS}
210	P	V_{DD_CORE}
211	I/O	PB.12 / EPWM2_CH0 / UART4_nCTS / UART3_RXD / I2C3_SDA / CAN2_RXD / I2S1_LRCK / ADC0_CH4 / EBI_ADR16 / ECAP2_IC0
212	I/O	PB.13 / EPWM2_CH1 / UART4_nRTS / UART3_TXD / I2C3_SCL / CAN2_TXD / I2S1_BCLK / ADC0_CH5 / EBI_ADR17 / ECAP2_IC1
213	I/O	PB.14 / EPWM2_CH2 / UART4_RXD / CAN1_RXD / I2C4_SDA / I2S1_DI / ADC0_CH6 / EBI_ADR18 / ECAP2_IC2
214	I/O	PB.15 / EPWM2_CH3 / UART4_TXD / CAN1_TXD / I2C4_SCL / I2S1_DO / ADC0_CH7 / EBI_ADR19
215	P	V_{DDIO0}
216	P	V_{DD_CORE}
EPAD	P	V_{SS} **Note:** When soldering SMD, please make sure the EPAD on the bottom of LQFP-EP 216-Pin package is tightly soldered to the ground of the PCB.

4.2.2 MA35D1 Series BGA 312-Ball Pinout Function Table

Pin	Type	MA35D16A8(I)87C / MA35D16AJ87C (BGA 312-Ball, MCP with 256 / 512 MB DDR3L) Pin Function
A1	P	V_{SS}
A2	I/O	PB.9 / EPWM2_CH4 / UART2_nRTS / UART1_TXD / I2C2_SCL / SPI0_CLK / I2S0_MCLK / CCAP1_HSYNC / ADC0_CH1 / EBI_ALE / EBI_AD13 / TM0_EXT / I2S1_MCLK / SC0_nCD / QEI2_A / KPI_ROW7
A3	A	EADC0_CH6
A4	A	EADC0_CH2
A5	I/O	PF.11 / UART13_nRTS / I2S0_BCLK / SPI1_CLK / RGMII1_TXCLK / SC0_DAT / KPI_COL7
A6	I/O	PF.5 / UART11_nRTS / UART10_TXD / I2S0_BCLK / SPI1_CLK / RGMII1_RXCLK / RMII1_REFCLK / CAN2_TXD / KPI_ROW1
A7	I/O	PF.6 / UART11_RXD / I2S0_DI / SPI1_MOSI / RGMII1_RXCTL / RMII1_CRSDV / I2C4_SDA / SC0_CLK / KPI_ROW2
A8	I/O	PF.1 / UART2_nRTS / UART1_TXD / RGMII0_TXCLK / RGMII1_MDIO / RMII1_MDIO / KPI_COL1
A9	I/O	PE.11 / UART15_nRTS / UART14_TXD / SPI1_CLK / CCAP1_VSYNC / RGMII0_TXCLK
A10	I/O	PE.8 / UART13_nCTS / UART12_RXD / CCAP1_SCLK / RGMII0_RXD1 / RMII0_RXD1
A11	I/O	PE.5 / UART4_nRTS / UART3_TXD / CCAP1_DATA5 / RGMII0_RXCLK / RMII0_REFCLK
A12	I/O	PE.2 / UART9_RXD / CCAP1_DATA2 / RGMII0_TXCTL / RMII0_TXEN
A13	I	XT1_IN
A14	I/O	PN.14 / UART6_RXD / CAN3_RXD / SPI1_SS1 / CCAP1_SFIELD / SPI1_I2SMCLK
A15	I/O	PN.12 / UART6_nCTS / UART12_RXD / I2C5_SDA / CCAP1_HSYNC
A16	I/O	PN.11 / CAN2_TXD / CCAP1_PIXCLK
A17	I/O	PN.5 / I2C1_SCL / CCAP1_DATA5
A18	P	V_{SS}
B1	I/O	PB.12 / EPWM2_CH0 / UART4_nCTS / UART3_RXD / I2C3_SDA / CAN2_RXD / I2S1_LRCK / ADC0_CH4 / EBI_ADR16 / ECAP2_IC0
B2	I/O	PB.11 / EPWM2_BRAKE1 / UART2_TXD / CAN0_TXD / SPI0_MISO / I2S1_MCLK / CCAP1_SFIELD / ADC0_CH3 / EBI_nCS2 / EBI_ALE / TM5_EXT / I2C1_SCL / INT2 / QEI2_INDEX
B3	I/O	PB.8 / EPWM2_BRAKE0 / UART2_nCTS / UART1_RXD / I2C2_SDA / SPI0_SS1 / SPI0_I2SMCLK / ADC0_CH0 / EBI_nCS0 / TM4 / QEI2_INDEX / KPI_ROW6 **Note:** When this pin be chosen as ADC0_CH0 pinout function, it has another function which is the reference voltage (V_{REF}) input of ADC0 by setting the control register REFSEL (ADC_CONF[7:6]) to 2'b00. In this case, it needs to provide a reference voltage for ADC0 on this pin and the ADC0 channel 0 analog input function will be disabled.
B4	A	EADC0_CH5
B5	A	EADC0_CH1
B6	I/O	PF.13 / I2S0_DO / SPI1_MISO / RGMII1_TXD3 / SC0_PWR / KPI_ROW5
B7	I/O	PF.7 / UART11_TXD / I2S0_DO / SPI1_MISO / RGMII1_RXD0 / RMII1_RXD0 / I2C4_SCL / SC0_DAT / KPI_ROW3
B8	I/O	PF.2 / UART2_RXD / RGMII0_TXD2 / RGMII1_TXCTL / RMII1_TXEN / KPI_COL2
B9	I/O	PE.12 / UART15_RXD / SPI1_MOSI / CCAP1_DATA8 / RGMII0_TXD2

Pin	Type	MA35D16A8(I)87C / MA35D16AJ87C (BGA 312-Ball, MCP with 256 / 512 MB DDR3L) Pin Function
B10	I/O	PE.9 / UART13_nRTS / UART12_TXD / CCAP1_PIXCLK / RGMII0_RXD2 / RMII0_RXERR
B11	I/O	PE.6 / UART4_RXD / CCAP1_DATA6 / RGMII0_RXCTL / RMII0_CRSDV
B12	I/O	PE.1 / UART9_nRTS / UART8_TXD / CCAP1_DATA1 / RGMII0_MDIO / RMII0_MDIO
B13	O	XT1_OUT
B14	I/O	PN.13 / UART6_nRTS / UART12_TXD / I2C5_SCL / CCAP1_VSYNC
B15	I/O	PN.10 / CAN2_RXD / CCAP1_SCLK
B16	I/O	PN.7 / CAN1_TXD / CCAP1_DATA7
B17	I/O	PN.0 / I2C2_SDA / CCAP1_DATA0
B18	I/O	PN.2 / CAN0_RXD / CCAP1_DATA2
C1	I/O	PB.15 / EPWM2_CH3 / UART4_TXD / CAN1_TXD / I2C4_SCL / I2S1_DO / ADC0_CH7 / EBI_ADR19
C2	I/O	PB.13 / EPWM2_CH1 / UART4_nRTS / UART3_TXD / I2C3_SCL / CAN2_TXD / I2S1_BCLK / ADC0_CH5 / EBI_ADR17 / ECAP2_IC1
C3	A	EADC0_CH7
C4	A	EADC0_CH4
C5	A	EADC0_CH0
C6	I/O	PF.12 / I2S0_DI / SPI1_MOSI / RGMII1_TXD2 / SC0_RST / KPI_ROW4
C7	I/O	PF.9 / UART13_TXD / I2C5_SCL / SPI0_SS1 / RGMII1_RXD2 / RMII1_RXERR / SC0_PWR / KPI_COL5
C8	I/O	PF.3 / UART2_TXD / RGMII0_TXD3 / RGMII1_TXD0 / RMII1_TXD0 / KPI_COL3
C9	I/O	PE.13 / UART15_TXD / SPI1_MISO / CCAP1_DATA9 / RGMII0_TXD3
C10	I/O	PE.7 / UART4_TXD / CCAP1_DATA7 / RGMII0_RXD0 / RMII0_RXD0
C11	I/O	PE.4 / UART4_nCTS / UART3_RXD / CCAP1_DATA4 / RGMII0_TXD1 / RMII0_TXD1
C12	I/O	PE.0 / UART9_nCTS / UART8_RXD / CCAP1_DATA0 / RGMII0_MDC / RMII0_MDC
C13	I/O	PK.12 / EPWM2_CH0 / UART1_nCTS / UART13_RXD / I2C4_SDA / I2S0_LRCK / SPI1_SS0 / SC0_CLK / TM10 / INT2
C14	I/O	PK.14 / EPWM2_CH2 / UART1_RXD / CAN3_RXD / I2S0_DI / SPI1_MOSI / SC0_RST / I2C5_SDA / TM11 / INT3
C15	I/O	PN.6 / CAN1_RXD / CCAP1_DATA6
C16	I/O	PN.4 / I2C1_SDA / CCAP1_DATA4
C17	I/O	PN.1 / I2C2_SCL / CCAP1_DATA1
C18	I/O	PN.3 / CAN0_TXD / CCAP1_DATA3
D1	I/O	PB.14 / EPWM2_CH2 / UART4_RXD / CAN1_RXD / I2C4_SDA / I2S1_DI / ADC0_CH6 / EBI_ADR18 / ECAP2_IC2
D2	I/O	PB.10 / EPWM2_CH5 / UART2_RXD / CAN0_RXD / SPI0_MOSI / EBI_MCLK / CCAP1_VSYNC / ADC0_CH2 / EBI_ADR15 / EBI_AD14 / TM5 / I2C1_SDA / INT1 / QEI2_B
D3	A	EADC0_CH3
D4	P	V_{SS}

Pin	Type	MA35D16A8(I)87C / MA35D16AJ87C (BGA 312-Ball, MCP with 256 / 512 MB DDR3L) Pin Function
D5	I/O	PF.14 / EPWM2_BRAKE0 / EADC0_ST / RGMII1_PPS / RMII1_PPS / SPI0_I2SMCLK / SPI1_I2SMCLK / CCAP1_SFIELD / RGMII0_PPS / RMII0_PPS / TM0 / INT0 / SPI1_SS1 / QEI2_INDEX / I2S0_MCLK
D6	I/O	PF.10 / UART13_nCTS / I2S0_LRCK / SPI1_SS0 / RGMII1_RXD3 / SC0_CLK / KPI_COL6
D7	I/O	PF.8 / UART13_RXD / I2C5_SDA / SPI0_SS0 / RGMII1_RXD1 / RMII1_RXD1 / SC0_RST / KPI_COL4
D8	I/O	PF.4 / UART11_nCTS / UART10_RXD / I2S0_LRCK / SPI1_SS0 / RGMII1_TXD1 / RMII1_TXD1 / CAN2_RXD / KPI_ROW0
D9	I/O	PF.0 / UART2_nCTS / UART1_RXD / RGMII0_RXD3 / RGMII1_MDC / RMII1_MDC / KPI_COL0
D10	I/O	PE.10 / UART15_nCTS / UART14_RXD / SPI1_SS0 / CCAP1_HSYNC / RGMII0_RXD3
D11	I/O	PE.3 / UART9_TXD / CCAP1_DATA3 / RGMII0_TXD0 / RMII0_TXD0
D12	I/O	PK.15 / EPWM2_CH3 / UART1_TXD / CAN3_TXD / I2S0_DO / SPI1_MISO / SC0_PWR / I2C5_SCL / TM11_EXT
D13	I/O	PN.15 / EPWM2_CH4 / UART6_TXD / CAN3_TXD / I2S0_MCLK / SPI1_SS1 / SPI1_I2SMCLK / SC0_nCD / EADC0_ST / CLKO / TM6
D14	I/O	PK.13 / EPWM2_CH1 / UART1_nRTS / UART13_TXD / I2C4_SCL / I2S0_BCLK / SPI1_CLK / SC0_DAT / TM10_EXT
D15	P	V_{SS}
D16	P	V_{SS}
D17	P	V_{SS}
D18	I/O	PC.1 / I2C4_SCL / SD0_CLK/eMMC0_CLK
E1	I/O	PL.12 / EPWM0_SYNC_IN / UART7_nCTS / ECAP1_IC0 / UART14_RXD / SPI0_SS0 / I2S1_LRCK / SC1_CLK / EBI_AD0 / HSUSBH_PWREN / I2C2_SDA / TM0 / EPWM0_CH2 / EBI_AD11 / RGMII0_PPS / RMII0_PPS
E2	I/O	PL.14 / EPWM0_CH2 / UART7_RXD / CAN1_RXD / SPI0_MOSI / I2S1_DI / SC1_RST / EBI_AD2 / TM2 / INT0 / EBI_AD13
E3	I/O	PL.13 / EPWM0_SYNC_OUT / UART7_nRTS / ECAP1_IC1 / UART14_TXD / SPI0_CLK / I2S1_BCLK / SC1_DAT / EBI_AD1 / HSUSBH_OVC / I2C2_SCL / TM0_EXT / EPWM0_CH3 / EBI_AD12 / RGMII1_PPS / RMII1_PPS
E4	I/O	PF.15 / HSUSB0_VBUSVLD
E7	O	RTC_RPWR
E8	I/O	PH.9 / CLK_32KOUT / TAMPER1
E9	I/O	PH.8 / TAMPER0
E10	P	V_{DDIO9}
E11	P	V_{DDIO8}
E12	P	AV_{DDL_PLL2}
E15	P	V_{SS}
E16	P	V_{SS}
E17	I/O	PC.7 / CAN1_TXD / SD0_WP
E18	I/O	PC.2 / CAN0_RXD / SD0_DAT0/eMMC0_DAT0
F1	P	V_{SS}

Pin	Type	MA35D16A8(I)87C / MA35D16AJ87C (BGA 312-Ball, MCP with 256 / 512 MB DDR3L) Pin Function
F2	P	V_{SS}
F3	I/O	PL.15 / EPWM0_CH1 / UART7_TXD / TRACE_CLK / CAN1_TXD / SPI0_MISO / I2S1_DO / SC1_PWR / EBI_AD3 / TM2_EXT / INT2 / EBI_AD14
F4	I/O	PA.3 / UART1_TXD / NAND_DATA3 / EBI_AD3 / EBI_ADR3
F6	P	V_{DD_HSUSB1}
F7	P	V_{BAT}
F8	P	AV_{DD}
F9	I	V_{REF_EADC0}
F10	P	AV_{DDH_PLL2}
F11	P	AV_{DDL_PLL1}
F12	P	V_{DD_PLL1}
F13	P	V_{DDIO7}
F15	I/O	PD.15 / EPWM0_SYNC_OUT / UART11_nRTS / CAN3_TXD / TRACE_DATA3 / EBI_ALE / EBI_AD7 / ECAP0_IC2 / TM6_EXT / I2S1_DO
F16	I/O	PI.6 / EPWM0_BRAKE0 / UART14_RXD / CAN1_RXD / I2S1_DI / EBI_ADR6 / QEI1_INDEX / INT2
F17	I/O	PC.6 / CAN1_RXD / SD0_nCD **Note:** An external 10kΩ pull-down resistor is necessary on this pin when these PC.0~PC.5 pins (eMMC0_xxx) are used to connect with eMMC device and act as the booting source.
F18	I/O	PC.0 / I2C4_SDA / SD0_CMD/eMMC0_CMD
G1	I	X32_IN
G2	O	X32_OUT
G3	I/O	PA.0 / UART1_nCTS / UART16_RXD / NAND_DATA0 / EBI_AD0 / EBI_ADR0
G4	I/O	PA.5 / UART3_nRTS / UART2_TXD / NAND_DATA5 / EBI_AD5 / EBI_ADR5
G5	I	RTC_nRWAKE
G6	P	V_{DD_OTP}
G7	P	V_{DD_HSUSB0}
G8	P	AV_{DD_ADC0}
G9	P	AV_{DD_EADC0}
G10	P	AV_{DDH_PLL1}
G11	P	AV_{DDL_PLL0}
G12	P	V_{DDIO10}
G13	P	V_{DDIO6}
G14	I/O	PD.13 / EPWM0_BRAKE1 / UART11_RXD / UART10_TXD / I2C4_SCL / TRACE_DATA1 / EBI_nCS2 / EBI_AD5 / ECAP0_IC0 / TM5_EXT / I2S1_BCLK
G15	I/O	PA.15 / EPWM0_CH2 / UART9_nCTS / UART6_RXD / I2C4_SDA / CAN2_RXD / EBI_ALE / QEI0_A / TM1 / RGMII0_PPS / RMII0_PPS **Note:** This pin is dedicated as PMIC_nIRQ input function from external PMIC by default.

Pin	Type	MA35D16A8(I)87C / MA35D16AJ87C (BGA 312-Ball, MCP with 256 / 512 MB DDR3L) Pin Function
G16	I/O	PK.1 / EPWM0_SYNC_OUT / UART16_nRTS / UART15_TXD / I2C4_SCL / EADC0_ST / EBI_ADR9 / TM7_EXT / ECAP0_IC2
G17	I/O	PC.5 / I2C5_SCL / SD0_DAT3/eMMC0_DAT3
G18	I/O	PC.3 / CAN0_TXD / SD0_DAT1/eMMC0_DAT1
H1	A	HSUSB1_D+
H2	A	HSUSB1_D-
H3	I/O	PA.1 / UART1_nRTS / UART16_TXD / NAND_DATA1 / EBI_AD1 / EBI_ADR1
H4	I/O	PA.7 / UART3_TXD / NAND_DATA7 / EBI_AD7 / EBI_ADR7
H5	I	nRESET / WDT_nRST
H6	P	V_{DD_CORE}
H7	P	AV_{DDL_ROSC}
H8	P	V_{DD_CORE}
H9	P	AV_{SS}
H10	P	AV_{SS_PLL}
H11	P	V_{SS}
H12	P	V_{DD_CORE}
H13	P	MV_{DD}
H14	P	MV_{REF}
H15	I/O	PI.1 / EPWM0_CH1 / UART12_nRTS / UART11_TXD / I2C2_SCL / SPI3_CLK / SC0_CLK / EBI_ADR1 / TM0_EXT / ECAP1_IC1
H16	I/O	PK.0 / EPWM0_SYNC_IN / UART16_nCTS / UART15_RXD / I2C4_SDA / I2S1_MCLK / EBI_ADR8 / TM7 / ECAP0_IC1
H17	I/O	PL.6 / EPWM0_CH0 / UART2_RXD / CAN0_RXD / QSPI1_MOSI1 / TRACE_CLK / EBI_AD5 / TM3 / ECAP1_IC0 / INT0
H18	I/O	PC.4 / I2C5_SDA / SD0_DAT2/eMMC0_DAT2
J1	A	HSUSB0_D+
J2	A	HSUSB0_D-
J3	I/O	PA.6 / UART3_RXD / NAND_DATA6 / EBI_AD6 / EBI_ADR6
J4	I/O	PA.8 / UART5_nCTS / UART4_RXD / NAND_RDY0 / EBI_AD8 / EBI_ADR8
J5	I	HSUSB0_ID
J6	P	V_{DD_CORE}
J7	P	V_{DD_CORE}
J8	P	V_{SS}
J9	P	V_{SS}
J10	P	V_{SS}
J11	P	V_{SS}

Pin	Type	MA35D16A8(I)87C / MA35D16AJ87C (BGA 312-Ball, MCP with 256 / 512 MB DDR3L) Pin Function
J12	P	V_{DD_CORE}
J13	P	MV_{DD}
J14	P	MZQ_DDRPHY
J15	I/O	PI.2 / EPWM0_CH2 / UART12_RXD / CAN0_RXD / SPI3_MOSI / SC0_DAT / EBI_ADR2 / TM1 / ECAP1_IC2
J16	I/O	PK.3 / EPWM1_CH1 / UART16_TXD / CAN2_TXD / SPI3_SS1 / SC1_nCD / EBI_ADR11 / QEI0_B
J17	I/O	PL.7 / EPWM0_CH1 / UART2_TXD / CAN0_TXD / QSPI1_MISO1 / EBI_AD6 / TM3_EXT / ECAP1_IC1 / INT1
J18	I/O	PL.5 / EPWM1_CH5 / UART2_nRTS / UART1_TXD / I2C4_SCL / SPI3_MISO / QSPI1_MISO0 / I2S1_MCLK / EBI_nWR / SC0_nCD / TM9_EXT / ECAP0_IC2
K1	I/O	PA.2 / UART1_RXD / NAND_DATA2 / EBI_AD2 / EBI_ADR2
K2	I/O	PA.4 / UART3_nCTS / UART2_RXD / NAND_DATA4 / EBI_AD4 / EBI_ADR4
K3	I/O	PA.9 / UART5_nRTS / UART4_TXD / NAND_nRE / EBI_AD9 / EBI_ADR9
K4	I/O	PA.12 / UART7_nCTS / UART8_RXD / NAND_ALE / EBI_AD12 / EBI_ADR12
K5	I/O	PA.10 / UART5_RXD / NAND_nWE / EBI_AD10 / EBI_ADR10
K6	P	V_{DD_CORE}
K7	P	V_{DD_CORE}
K8	P	V_{SS}
K9	P	V_{SS}
K10	P	V_{SS}
K11	P	V_{SS}
K12	P	V_{DD_CORE}
K13	P	MV_{DD}
K14	P	MZQ_SDRAM
K15	I/O	PI.0 / EPWM0_CH0 / UART12_nCTS / UART11_RXD / I2C2_SDA / SPI3_SS0 / SC0_nCD / EBI_ADR0 / TM0 / ECAP1_IC0
K16	I/O	PK.2 / EPWM1_CH0 / UART16_RXD / CAN2_RXD / SPI3_I2SMCLK / SC0_PWR / EBI_ADR10 / QEI0_A
K17	I/O	PL.8 / EPWM0_CH2 / UART14_nCTS / UART13_RXD / I2C5_SDA / SPI3_SS0 / EPWM0_CH4 / I2S1_LRCK / EBI_AD7 / SC0_CLK / TM4 / ECAP1_IC2 / INT2
K18	I/O	PL.4 / EPWM1_CH4 / UART2_nCTS / UART1_RXD / I2C4_SDA / SPI3_MOSI / QSPI1_MOSI0 / I2S0_MCLK / EBI_nRD / SC1_nCD / TM9 / ECAP0_IC1
L1	I/O	PA.13 / UART7_nRTS / UART8_TXD / NAND_nCS0 / EBI_AD13 / EBI_ADR13
L2	I/O	PA.11 / UART5_TXD / NAND_CLE / EBI_AD11 / EBI_ADR11
L3	I/O	PG.2 / EPWM0_CH4 / UART9_RXD / CAN0_RXD / SPI0_SS1 / EBI_ADR16 / EBI_nCS2 / QEI0_A / TM3 / INT1
L4	I/O	PG.4 / EPWM1_CH0 / UART5_nCTS / UART6_RXD / SPI3_SS0 / QEI1_INDEX / EBI_ADR18 / EBI_nCS0 / I2S1_DO / SC1_CLK / TM4 / INT2 / ECAP1_IC2
L5	I/O	PG.6 / EPWM1_CH2 / UART5_RXD / CAN1_RXD / SPI3_MOSI / ECAP0_IC1 / EBI_nRD / I2S1_BCLK

Pin	Type	MA35D16A8(I)87C / MA35D16AJ87C (BGA 312-Ball, MCP with 256 / 512 MB DDR3L) Pin Function
		/ SC1_RST / TM7 / INT3
L6	P	V_{DDIO0}
L7	P	V_{DD_CPU}
L8	P	V_{SS}
L9	P	V_{SS}
L10	P	V_{SS}
L11	P	V_{SS}
L12	P	V_{DD_CORE}
L13	P	MV_{DD}
L14	I/O	PD.14 / EPWM0_SYNC_IN / UART11_nCTS / CAN3_RXD / TRACE_DATA2 / EBI_MCLK / EBI_AD6 / ECAP0_IC1 / TM6 / I2S1_DI / INT3
L15	I/O	PI.3 / EPWM0_CH3 / UART12_TXD / CAN0_TXD / SPI3_MISO / SC0_RST / EBI_ADR3 / TM1_EXT
L16	I/O	PJ.12 / EPWM1_CH2 / UART2_nCTS / UART1_RXD / I2C5_SDA / SPI3_SS0 / SC1_CLK / EBI_ADR12 / TM2 / QEI0_INDEX
L17	I/O	PL.9 / EPWM0_CH3 / UART14_nRTS / UART13_TXD / I2C5_SCL / SPI3_CLK / EPWM1_CH4 / I2S1_BCLK / EBI_AD8 / SC0_DAT / TM4_EXT / QEI0_A / INT3
L18	I/O	PL.3 / EPWM1_CH3 / UART11_TXD / CAN3_TXD / SPI2_CLK / QSPI1_CLK / I2S0_DO / EBI_AD14 / SC1_PWR / TM7_EXT / ECAP0_IC0
M1	I/O	PG.1 / EPWM0_CH3 / UART9_nRTS / UART6_TXD / I2C4_SCL / CAN2_TXD / EBI_nCS0 / QEI0_B / TM1_EXT / RGMII1_PPS / RMII1_PPS
M2	I/O	PG.0 / EPWM0_CH0 / UART7_TXD / CAN3_TXD / SPI0_SS0 / EADC0_ST / EBI_AD15 / I2S1_MCLK / QEI0_INDEX / TM1 / CLKO / INT0 / EBI_ADR15
M3	I/O	PG.5 / EPWM1_CH1 / UART5_nRTS / UART6_TXD / SPI3_CLK / ECAP0_IC0 / EBI_ADR19 / EBI_ALE / I2S1_DI / SC1_DAT / TM4_EXT
M4	I/O	PG.3 / EPWM0_CH5 / UART9_TXD / CAN0_TXD / SPI0_I2SMCLK / EBI_ADR17 / EBI_nCS1 / EBI_MCLK / QEI0_B / TM3_EXT / I2S1_MCLK
M5	I/O	PA.14 / UART7_RXD / CAN3_RXD / NAND_nWP / EBI_AD14 / EBI_ADR14
M6	P	V_{DDIO1}
M7	P	V_{DD_CPU}
M8	P	V_{SS}
M9	P	V_{SS}
M10	P	V_{SS}
M11	P	V_{SS}
M12	P	$MV_{DD_DPHYPLL}$
M13	P	V_{DDIO5}
M14	I/O	PD.12 / EPWM0_BRAKE0 / UART11_TXD / UART10_RXD / I2C4_SDA / TRACE_DATA0 / EBI_nCS1 / EBI_AD4 / QEI0_INDEX / TM5 / I2S1_LRCK / INT1
M15	I/O	PI.5 / EPWM0_CH5 / UART14_nRTS / UART13_TXD / I2C3_SCL / I2S1_BCLK / EBI_ADR5 / INT1
M16	I/O	PJ.13 / EPWM1_CH3 / UART2_nRTS / UART1_TXD / I2C5_SCL / SPI3_MOSI / SC1_DAT / EBI_ADR13 / TM2_EXT

Pin	Type	MA35D16A8(I)87C / MA35D16AJ87C (BGA 312-Ball, MCP with 256 / 512 MB DDR3L) Pin Function
M17	I/O	PL.10 / EPWM0_CH4 / UART14_RXD / CAN3_RXD / SPI3_MOSI / EPWM0_CH5 / I2S1_DI / EBI_AD9 / SC0_RST / EBI_nWRH / QEI0_B
M18	I/O	PL.0 / EPWM1_CH0 / UART11_nCTS / UART10_RXD / I2C3_SDA / SPI2_MOSI / QSPI1_MOSI1 / I2S0_LRCK / EBI_AD11 / SC1_CLK / TM5 / QEI1_A
N1	I/O	PM.13 / EPWM1_CH5 / UART10_nRTS / TRACE_DATA1 / UART11_TXD / I2C2_SCL / EBI_AD9 / ECAP1_IC0 / TM8_EXT
N2	I/O	PM.14 / EPWM1_BRAKE0 / UART10_RXD / TRACE_DATA2 / CAN2_RXD / I2C3_SDA / EBI_AD10 / ECAP1_IC1 / TM10 / INT1
N3	I/O	PM.12 / EPWM1_CH4 / UART10_nCTS / TRACE_DATA0 / UART11_RXD / I2C2_SDA / SC1_nCD / EBI_AD8 / I2S1_MCLK / TM8
N4	I/O	PG.7 / EPWM1_CH3 / UART5_TXD / CAN1_TXD / SPI3_MISO / ECAP0_IC2 / EBI_nWR / I2S1_LRCK / SC1_PWR / TM7_EXT
N6	P	V_{DDIO2}
N7	P	V_{DD_CPU}
N8	P	V_{DD_CPU}
N9	P	V_{DD_CORE}
N10	P	V_{DD_CORE}
N11	P	V_{DD_CORE}
N12	P	V_{DDIO3}
N13	P	V_{DDIO4}
N15	I/O	PI.4 / EPWM0_CH4 / UART14_nCTS / UART13_RXD / I2C3_SDA / SPI2_SS1 / I2S1_LRCK / EBI_ADR4 / INT0
N16	I/O	PJ.15 / EPWM1_CH5 / UART2_TXD / CAN3_TXD / SPI3_CLK / EADC0_ST / SC1_PWR / EBI_ADR15 / TM3_EXT / INT1
N17	I/O	PL.11 / EPWM0_CH5 / UART14_TXD / CAN3_TXD / SPI3_MISO / EPWM1_CH5 / I2S1_DO / EBI_AD10 / SC0_PWR / EBI_nWRL / QEI0_INDEX
N18	I/O	PL.1 / EPWM1_CH1 / UART11_nRTS / UART10_TXD / I2C3_SCL / SPI2_MISO / QSPI1_MISO1 / I2S0_BCLK / EBI_AD12 / SC1_DAT / TM5_EXT / QEI1_B
P1	I/O	PM.15 / EPWM1_BRAKE1 / UART10_TXD / TRACE_DATA3 / CAN2_TXD / I2C3_SCL / EBI_AD11 / ECAP1_IC2 / TM10_EXT / INT2
P2	I/O	PG.12 / JTAG_TCK/SW_CLK / I2S0_LRCK / EBI_nWRL / EBI_AD1 **Note:** When the chip enters debug mode, this pin will be automatically forced to JTAG/SWD functionality by hardware, and other pinout functions of this pin will be disabled.
P3	I/O	PG.15 / JTAG_nTRST / I2S0_DO / EBI_nCS0 / EBI_AD4 **Note:** When the chip enters debug mode, this pin will be automatically forced to JTAG functionality by hardware, and other pinout functions of this pin will be disabled.
P4	I/O	PG.11 / JTAG_TDO / I2S0_MCLK / NAND_RDY1 / EBI_nWRH / EBI_nCS1 / EBI_AD0 **Note:** When the chip enters debug mode, this pin will be automatically forced to JTAG functionality by hardware, and other pinout functions of this pin will be disabled.
P7	I/O	PM.1 / I2C4_SCL / SPI3_I2SMCLK / CCAP0_SFIELD / EBI_AD4 / EBI_ADR4
P8	I/O	PJ.5 / I2C3_SCL / SD1_nCD **Note:** An external 10kΩ pull-down resistor is necessary on this pin when these PJ.0~PJ.3 and PJ.6~PJ.11 pins (eMMC1_xxx) are used to connect with eMMC device and act as the booting source.

Pin	Type	MA35D16A8(I)87C / MA35D16AJ87C (BGA 312-Ball, MCP with 256 / 512 MB DDR3L) Pin Function
P9	I/O	PJ.4 / I2C3_SDA / SD1_WP
P10	I/O	PK.5 / EPWM1_CH1 / UART12_nRTS / UART13_TXD / I2C4_SCL / SPI2_CLK / I2S1_DI / SC0_DAT / EADC0_ST / TM8_EXT / INT1
P11	I/O	PK.6 / EPWM1_CH2 / UART12_RXD / CAN0_RXD / SPI2_MOSI / I2S1_BCLK / SC0_RST / TM6 / INT2
P12	I/O	PK.7 / EPWM1_CH3 / UART12_TXD / CAN0_TXD / SPI2_MISO / I2S1_LRCK / SC0_PWR / CLKO / TM6_EXT / INT3
P15	I/O	PI.7 / EPWM0_BRAKE1 / UART14_TXD / CAN1_TXD / I2S1_DO / EBI_ADR7 / ECAP0_IC0 / INT3
P16	I/O	PJ.14 / EPWM1_CH4 / UART2_RXD / CAN3_RXD / SPI3_MISO / SC1_RST / EBI_ADR14 / TM3
P17	I/O	PK.8 / EPWM1_CH0 / I2C3_SDA / SPI3_CLK / EADC0_ST / EBI_AD15 / EBI_MCLK / EBI_ADR15 / TM8 / QEI1_INDEX
P18	I/O	PL.2 / EPWM1_CH2 / UART11_RXD / CAN3_RXD / SPI2_SS0 / QSPI1_SS1 / I2S0_DI / EBI_AD13 / SC1_RST / TM7 / QEI1_INDEX
R1	I/O	PG.14 / JTAG_TDI / I2S0_DI / NAND_nCS1 / EBI_ALE / EBI_AD3 **Note:** When the chip enters debug mode, this pin will be automatically forced to JTAG functionality by hardware, and other pinout functions of this pin will be disabled.
R2	I/O	PG.13 / JTAG_TMS/SW_DIO / I2S0_BCLK / EBI_MCLK / EBI_AD2 **Note:** When the chip enters debug mode, this pin will be automatically forced to JTAG/SWD functionality by hardware, and other pinout functions of this pin will be disabled.
R3	I/O	PD.7 / EPWM0_SYNC_OUT / UART1_TXD / QSPI1_MISO1 / I2C0_SCL / I2S1_MCLK / EPWM0_CH1 / EBI_AD6 / SC1_nCD / EADC0_ST **Note:** This pin is dedicated as I2C0_SCL function for controlling external PMIC by default.
R4	P	V_{SS}
R5	I/O	PM.11 / EPWM1_CH3 / CAN2_TXD / SPI3_SS1 / CCAP0_DATA9 / SPI2_SS1 / EBI_AD14 / EBI_ADR14
R6	I/O	PM.9 / I2C0_SCL / CCAP0_DATA7 / EBI_AD12 / EBI_ADR12
R7	I/O	PJ.1 / EPWM1_BRAKE1 / UART8_nRTS / UART7_TXD / I2C2_SCL / SPI2_CLK / eMMC1_DAT5 / I2S0_BCLK / SC0_DAT / EBI_AD12 / EBI_ADR17 / EBI_nCS1 / EBI_AD8
R8	I/O	PJ.3 / EPWM1_CH5 / UART8_TXD / CAN1_TXD / SPI2_MISO / eMMC1_DAT7 / I2S0_DO / SC0_PWR / EBI_AD14 / EBI_ADR19 / EBI_nWRL / EBI_AD10
R9	I/O	PJ.9 / I2C4_SCL / SD1_DAT1/eMMC1_DAT1
R10	I/O	PK.4 / UART12_nCTS / UART13_RXD / SPI2_MISO / LCM_DEN/LCM_MPU_RS / EBI_AD10 / EBI_nWRL
R11	I/O	PI.9 / UART4_nRTS / UART3_TXD / LCM_DATA1/LCM_MPU_D1 / EBI_AD12
R12	I/O	PI.14 / UART6_RXD / LCM_DATA6/LCM_MPU_D6
R13	I/O	PH.3 / UART8_TXD / LCM_DATA11/LCM_MPU_D11
R14	I/O	PH.7 / UART10_TXD / LCM_DATA15/LCM_MPU_D15
R15	P	V_{SS}
R16	I/O	PC.14 / UART12_RXD / LCM_DATA18/LCM_MPU_CS
R17	I/O	PH.12 / UART14_nCTS / UART13_RXD / LCM_DATA20
R18	I/O	PH.14 / UART14_RXD / LCM_DATA22

Pin	Type	MA35D16A8(I)87C / MA35D16AJ87C (BGA 312-Ball, MCP with 256 / 512 MB DDR3L) Pin Function
T1	I/O	PE.14 / UART0_TXD
T2	I/O	PE.15 / UART0_RXD **Note:** This GPIO PE.15 (UART0_RXD) pin must be pulled to high level through an external 10kΩ resistor or an internal pull-up resistor in the external device (such as a transceiver or MCU).
T3	I/O	PD.11 / EPWM1_BRAKE1 / UART16_TXD / QSPI1_MISO0 / I2S1_DO / EPWM0_CH5 / EBI_AD10 / SC1_PWR / TM2_EXT
T4	I/O	PK.11 / CAN1_TXD / CCAP0_HSYNC / EBI_AD2 / EBI_ADR2
T5	I/O	PM.3 / CAN3_TXD / CCAP0_DATA1 / EBI_AD6 / EBI_ADR6
T6	I/O	PM.7 / CAN0_TXD / CCAP0_DATA5 / EBI_AD10 / EBI_ADR10
T7	I/O	PM.8 / I2C0_SDA / CCAP0_DATA6 / EBI_AD11 / EBI_ADR11
T8	I/O	PJ.2 / EPWM1_CH4 / UART8_RXD / CAN1_RXD / SPI2_MOSI / eMMC1_DAT6 / I2S0_DI / SC0_RST / EBI_AD13 / EBI_ADR18 / EBI_nWRH / EBI_AD9
T9	I/O	PJ.6 / CAN3_RXD / SD1_CMD/eMMC1_CMD
T10	I/O	PG.8 / EPWM1_CH4 / UART12_RXD / CAN3_RXD / SPI2_SS0 / LCM_VSYNC/LCM_MPU_RD/EN / I2C3_SDA / EBI_AD7 / EBI_nCS0
T11	I/O	PI.8 / UART4_nCTS / UART3_RXD / LCM_DATA0/LCM_MPU_D0 / EBI_AD11
T12	I/O	PI.13 / UART6_nRTS / UART5_TXD / LCM_DATA5/LCM_MPU_D5
T13	I/O	PH.2 / UART8_RXD / LCM_DATA10/LCM_MPU_D10
T14	I/O	PH.6 / UART10_RXD / LCM_DATA14/LCM_MPU_D14
T15	I/O	PD.5 / UART1_nRTS / UART2_TXD / I2C2_SCL / QSPI0_MISO1
T16	I/O	PC.12 / UART12_nCTS / UART11_RXD / LCM_DATA16/LCM_MPU_D16
T17	I/O	PC.13 / UART12_nRTS / UART11_TXD / LCM_DATA17/LCM_MPU_D17
T18	I/O	PH.15 / UART14_TXD / LCM_DATA23
U1	I/O	PD.6 / EPWM0_SYNC_IN / UART1_RXD / QSPI1_MOSI1 / I2C0_SDA / I2S0_MCLK / EPWM0_CH0 / EBI_AD5 / SPI3_SS1 / TRACE_CLK **Note:** This pin is dedicated as I2C0_SDA function for controlling external PMIC by default.
U2	I/O	PD.8 / EPWM0_BRAKE0 / UART16_nCTS / UART15_RXD / QSPI1_SS0 / I2S1_LRCK / EPWM0_CH2 / EBI_AD7 / SC1_CLK / TM0
U3	I/O	PD.10 / EPWM1_BRAKE0 / UART16_RXD / QSPI1_MOSI0 / I2S1_DI / EPWM0_CH4 / EBI_AD9 / SC1_RST / TM2
U4	I/O	PM.0 / I2C4_SDA / CCAP0_VSYNC / EBI_AD3 / EBI_ADR3
U5	I/O	PM.4 / I2C5_SDA / CCAP0_DATA2 / EBI_AD7 / EBI_ADR7
U6	I/O	PM.5 / I2C5_SCL / CCAP0_DATA3 / EBI_AD8 / EBI_ADR8
U7	I/O	PM.10 / EPWM1_CH2 / CAN2_RXD / SPI3_SS0 / CCAP0_DATA8 / SPI2_I2SMCLK / EBI_AD13 / EBI_ADR13
U8	I/O	PJ.11 / CAN0_TXD / SD1_DAT3/eMMC1_DAT3
U9	I/O	PJ.8 / I2C4_SDA / SD1_DAT0/eMMC1_DAT0
U10	I/O	PG.9 / EPWM1_CH5 / UART12_TXD / CAN3_TXD / SPI2_CLK / LCM_HSYNC/LCM_MPU_WR/RW / I2C3_SCL / EBI_AD8 / EBI_nCS1

Pin	Type	MA35D16A8(I)87C / MA35D16AJ87C (BGA 312-Ball, MCP with 256 / 512 MB DDR3L) Pin Function
U11	I/O	PI.15 / UART6_TXD / LCM_DATA7/LCM_MPU_D7
U12	I/O	PI.12 / UART6_nCTS / UART5_RXD / LCM_DATA4/LCM_MPU_D4
U13	I/O	PH.1 / UART8_nRTS / UART7_TXD / LCM_DATA9/LCM_MPU_D9
U14	I/O	PH.5 / UART10_nRTS / UART9_TXD / LCM_DATA13/LCM_MPU_D13
U15	I/O	PD.3 / UART3_TXD / QSPI0_MISO0
U16	I/O	PD.0 / UART3_nCTS / UART4_RXD / QSPI0_SS0
U17	I/O	PC.15 / UART12_TXD / LCM_DATA19 / LCM_MPU_TE / LCM_MPU_VSYNC
U18	I/O	PH.13 / UART14_nRTS / UART13_TXD / LCM_DATA21
V1	P	V_{SS}
V2	I/O	PD.9 / EPWM0_BRAKE1 / UART16_nRTS / UART15_TXD / QSPI1_CLK / I2S1_BCLK / EPWM0_CH3 / EBI_AD8 / SC1_DAT / TM0_EXT
V3	I/O	PK.9 / I2C3_SCL / CCAP0_SCLK / EBI_AD0 / EBI_ADR0
V4	I/O	PK.10 / CAN1_RXD / CCAP0_PIXCLK / EBI_AD1 / EBI_ADR1
V5	I/O	PM.2 / CAN3_RXD / CCAP0_DATA0 / EBI_AD5 / EBI_ADR5
V6	I/O	PM.6 / CAN0_RXD / CCAP0_DATA4 / EBI_AD9 / EBI_ADR9
V7	I/O	PJ.0 / EPWM1_BRAKE0 / UART8_nCTS / UART7_RXD / I2C2_SDA / SPI2_SS0 / eMMC1_DAT4 / I2S0_LRCK / SC0_CLK / EBI_AD11 / EBI_ADR16 / EBI_nCS0 / EBI_AD7
V8	I/O	PJ.7 / CAN3_TXD / SD1_CLK/eMMC1_CLK
V9	I/O	PJ.10 / CAN0_RXD / SD1_DAT2/eMMC1_DAT2
V10	I/O	PG.10 / UART12_nRTS / UART13_TXD / SPI2_MOSI / LCM_CLK / EBI_AD9 / EBI_nWRH
V11	I/O	PI.10 / UART4_RXD / LCM_DATA2/LCM_MPU_D2 / EBI_AD13
V12	I/O	PI.11 / UART4_TXD / LCM_DATA3/LCM_MPU_D3 / EBI_AD14
V13	I/O	PH.0 / UART8_nCTS / UART7_RXD / LCM_DATA8/LCM_MPU_D8
V14	I/O	PH.4 / UART10_nCTS / UART9_RXD / LCM_DATA12/LCM_MPU_D12
V15	I/O	PD.4 / UART1_nCTS / UART2_RXD / I2C2_SDA / QSPI0_MOSI1
V16	I/O	PD.2 / UART3_RXD / QSPI0_MOSI0
V17	I/O	PD.1 / UART3_nRTS / UART4_TXD / QSPI0_CLK
V18	P	V_{SS}

4.2.3 MA35D1 Series BGA 364-Ball Pinout Function Table

Pin	Type	MA35D16A087C (BGA 364-Ball, non-MCP) Pin Function
A1	P	V_{SS}
A2	A	EADC0_CH3
A3	I/O	PF.11 / UART13_nRTS / I2S0_BCLK / SPI1_CLK / RGMII1_TXCLK / SC0_DAT / KPI_COL7
A4	I/O	PF.10 / UART13_nCTS / I2S0_LRCK / SPI1_SS0 / RGMII1_RXD3 / SC0_CLK / KPI_COL6
A5	I/O	PF.5 / UART11_nRTS / UART10_TXD / I2S0_BCLK / SPI1_CLK / RGMII1_RXCLK / RMII1_REFCLK / CAN2_TXD / KPI_ROW1
A6	I/O	PF.3 / UART2_TXD / RGMII0_TXD3 / RGMII1_TXD0 / RMII1_TXD0 / KPI_COL3
A7	I/O	PF.1 / UART2_nRTS / UART1_TXD / RGMII0_TXCLK / RGMII1_MDIO / RMII1_MDIO / KPI_COL1
A8	I/O	PE.11 / UART15_nRTS / UART14_TXD / SPI1_CLK / CCAP1_VSYNC / RGMII0_TXCLK
A9	I/O	PE.9 / UART13_nRTS / UART12_TXD / CCAP1_PIXCLK / RGMII0_RXD2 / RMII0_RXERR
A10	I/O	PE.5 / UART4_nRTS / UART3_TXD / CCAP1_DATA5 / RGMII0_RXCLK / RMII0_REFCLK
A11	I/O	PE.3 / UART9_TXD / CCAP1_DATA3 / RGMII0_TXD0 / RMII0_TXD0
A12	I/O	PN.7 / CAN1_TXD / CCAP1_DATA7
A13	I/O	PN.1 / I2C2_SCL / CCAP1_DATA1
A14	I/O	PN.11 / CAN2_TXD / CCAP1_PIXCLK
A15	I	XT1_IN
A16	I/O	PN.6 / CAN1_RXD / CCAP1_DATA6
A17	I/O	PN.2 / CAN0_RXD / CCAP1_DATA2
A18	I/O	PN.13 / UART6_nRTS / UART12_TXD / I2C5_SCL / CCAP1_VSYNC
A19	I/O	PC.1 / I2C4_SCL / SD0_CLK/eMMC0_CLK
A20	P	V_{SS}
B1	A	EADC0_CH7
B2	A	EADC0_CH6
B3	A	EADC0_CH2
B4	I/O	PF.13 / I2S0_DO / SPI1_MISO / RGMII1_TXD3 / SC0_PWR / KPI_ROW5
B5	I/O	PF.14 / EPWM2_BRAKE0 / EADC0_ST / RGMII1_PPS / RMII1_PPS / SPI0_I2SMCLK / SPI1_I2SMCLK / CCAP1_SFIELD / RGMII0_PPS / RMII0_PPS / TM0 / INT0 / SPI1_SS1 / QEI2_INDEX / I2S0_MCLK
B6	I/O	PF.12 / I2S0_DI / SPI1_MOSI / RGMII1_TXD2 / SC0_RST / KPI_ROW4
B7	I/O	PF.9 / UART13_TXD / I2C5_SCL / SPI0_SS1 / RGMII1_RXD2 / RMII1_RXERR / SC0_PWR / KPI_COL5
B8	I/O	PF.7 / UART11_TXD / I2S0_DO / SPI1_MISO / RGMII1_RXD0 / RMII1_RXD0 / I2C4_SCL / SC0_DAT / KPI_ROW3
B9	I/O	PE.13 / UART15_TXD / SPI1_MISO / CCAP1_DATA9 / RGMII0_TXD3
B10	I/O	PE.7 / UART4_TXD / CCAP1_DATA7 / RGMII0_RXD0 / RMII0_RXD0
B11	I/O	PE.8 / UART13_nCTS / UART12_RXD / CCAP1_SCLK / RGMII0_RXD1 / RMII0_RXD1

Pin	Type	MA35D16A087C (BGA 364-Ball, non-MCP) Pin Function
B12	I/O	PE.1 / UART9_nRTS / UART8_TXD / CCAP1_DATA1 / RGMII0_MDIO / RMII0_MDIO
B13	I/O	PK.15 / EPWM2_CH3 / UART1_TXD / CAN3_TXD / I2S0_DO / SPI1_MISO / SC0_PWR / I2C5_SCL / TM11_EXT
B14	I/O	PN.12 / UART6_nCTS / UART12_RXD / I2C5_SDA / CCAP1_HSYNC
B15	O	XT1_OUT
B16	I/O	PN.4 / I2C1_SDA / CCAP1_DATA4
B17	I/O	PN.0 / I2C2_SDA / CCAP1_DATA0
B18	I/O	PC.2 / CAN0_RXD / SD0_DAT0/eMMC0_DAT0
B19	I/O	PC.5 / I2C5_SCL / SD0_DAT3/eMMC0_DAT3
B20	I/O	PC.3 / CAN0_TXD / SD0_DAT1/eMMC0_DAT1
C1	I/O	PB.15 / EPWM2_CH3 / UART4_TXD / CAN1_TXD / I2C4_SCL / I2S1_DO / ADC0_CH7 / EBI_ADR19
C2	I/O	PB.12 / EPWM2_CH0 / UART4_nCTS / UART3_RXD / I2C3_SDA / CAN2_RXD / I2S1_LRCK / ADC0_CH4 / EBI_ADR16 / ECAP2_IC0
C3	I/O	PB.9 / EPWM2_CH4 / UART2_nRTS / UART1_TXD / I2C2_SCL / SPI0_CLK / I2S0_MCLK / CCAP1_HSYNC / ADC0_CH1 / EBI_ALE / EBI_AD13 / TM0_EXT / I2S1_MCLK / SC0_nCD / QEI2_A / KPI_ROW7
C4	I/O	PB.11 / EPWM2_BRAKE1 / UART2_TXD / CAN0_TXD / SPI0_MISO / I2S1_MCLK / CCAP1_SFIELD / ADC0_CH3 / EBI_nCS2 / EBI_ALE / TM5_EXT / I2C1_SCL / INT2 / QEI2_INDEX
C5	A	EADC0_CH5
C6	A	EADC0_CH1
C7	I/O	PF.8 / UART13_RXD / I2C5_SDA / SPI0_SS0 / RGMII1_RXD1 / RMII1_RXD1 / SC0_RST / KPI_COL4
C8	I/O	PF.6 / UART11_RXD / I2S0_DI / SPI1_MOSI / RGMII1_RXCTL / RMII1_CRSDV / I2C4_SDA / SC0_CLK / KPI_ROW2
C9	I/O	PF.4 / UART11_nCTS / UART10_RXD / I2S0_LRCK / SPI1_SS0 / RGMII1_TXD1 / RMII1_TXD1 / CAN2_RXD / KPI_ROW0
C10	I/O	PF.2 / UART2_RXD / RGMII0_TXD2 / RGMII1_TXCTL / RMII1_TXEN / KPI_COL2
C11	I/O	PE.10 / UART15_nCTS / UART14_RXD / SPI1_SS0 / CCAP1_HSYNC / RGMII0_RXD3
C12	I/O	PE.2 / UART9_RXD / CCAP1_DATA2 / RGMII0_TXCTL / RMII0_TXEN
C13	I/O	PK.14 / EPWM2_CH2 / UART1_RXD / CAN3_RXD / I2S0_DI / SPI1_MOSI / SC0_RST / I2C5_SDA / TM11 / INT3
C14	I/O	PN.10 / CAN2_RXD / CCAP1_SCLK
C15	I/O	PN.5 / I2C1_SCL / CCAP1_DATA5
C16	P	V_{SS}
C17	P	V_{SS}
C18	I/O	PC.0 / I2C4_SDA / SD0_CMD/eMMC0_CMD
C19	P	V_{SS}
C20	I/O	MD14
D1	P	V_{SS}

Pin	Type	MA35D16A087C (BGA 364-Ball, non-MCP) Pin Function
D2	P	V_{SS}
D3	I/O	PL.12 / EPWM0_SYNC_IN / UART7_nCTS / ECAP1_IC0 / UART14_RXD / SPI0_SS0 / I2S1_LRCK / SC1_CLK / EBI_AD0 / HSUSBH_PWREN / I2C2_SDA / TM0 / EPWM0_CH2 / EBI_AD11 / RGMII0_PPS / RMII0_PPS
D4	I/O	PB.14 / EPWM2_CH2 / UART4_RXD / CAN1_RXD / I2C4_SDA / I2S1_DI / ADC0_CH6 / EBI_ADR18 / ECAP2_IC2
D5	I/O	PB.13 / EPWM2_CH1 / UART4_nRTS / UART3_TXD / I2C3_SCL / CAN2_TXD / I2S1_BCLK / ADC0_CH5 / EBI_ADR17 / ECAP2_IC1
D6	I/O	PF.15 / HSUSB0_VBUSVLD
D7	I/O	PL.15 / EPWM0_CH1 / UART7_TXD / TRACE_CLK / CAN1_TXD / SPI0_MISO / I2S1_DO / SC1_PWR / EBI_AD3 / TM2_EXT / INT2 / EBI_AD14
D8	I/O	PF.0 / UART2_nCTS / UART1_RXD / RGMII0_RXD3 / RGMII1_MDC / RMII1_MDC / KPI_COL0
D9	I/O	PE.6 / UART4_RXD / CCAP1_DATA6 / RGMII0_RXCTL / RMII0_CRSDV
D10	I/O	PE.12 / UART15_RXD / SPI1_MOSI / CCAP1_DATA8 / RGMII0_TXD2
D11	I/O	PE.4 / UART4_nCTS / UART3_RXD / CCAP1_DATA4 / RGMII0_TXD1 / RMII0_TXD1
D12	I/O	PE.0 / UART9_nCTS / UART8_RXD / CCAP1_DATA0 / RGMII0_MDC / RMII0_MDC
D13	P	V_{SS}
D14	I/O	PN.3 / CAN0_TXD / CCAP1_DATA3
D15	I/O	PC.4 / I2C5_SDA / SD0_DAT2/eMMC0_DAT2
D16	P	V_{SS}
D17	P	V_{SS}
D18	P	V_{SS}
D19	I/O	MD15
D20	I/O	MD12
E1	I	X32_IN
E2	O	X32_OUT
E3	O	RTC_RPWR
E4	I/O	PL.14 / EPWM0_CH2 / UART7_RXD / CAN1_RXD / SPI0_MOSI / I2S1_DI / SC1_RST / EBI_AD2 / TM2 / INT0 / EBI_AD13
E5	I/O	PH.9 / CLK_32KOUT / TAMPER1
E6	I/O	PL.13 / EPWM0_SYNC_OUT / UART7_nRTS / ECAP1_IC1 / UART14_TXD / SPI0_CLK / I2S1_BCLK / SC1_DAT / EBI_AD1 / HSUSBH_OVC / I2C2_SCL / TM0_EXT / EPWM0_CH3 / EBI_AD12 / RGMII1_PPS / RMII1_PPS
E7	I	nRESET / WDT_nRST
E8	I/O	PB.10 / EPWM2_CH5 / UART2_RXD / CAN0_RXD / SPI0_MOSI / EBI_MCLK / CCAP1_VSYNC / ADC0_CH2 / EBI_ADR15 / EBI_AD14 / TM5 / I2C1_SDA / INT1 / QEI2_B
E9	I/O	PB.8 / EPWM2_BRAKE0 / UART2_nCTS / UART1_RXD / I2C2_SDA / SPI0_SS1 / SPI0_I2SMCLK / ADC0_CH0 / EBI_nCS0 / TM4 / QEI2_INDEX / KPI_ROW6 **Note:** When this pin be chosen as ADC0_CH0 pinout function, it has another function which is the reference voltage (V_{REF}) input of ADC0 by setting the control register REFSEL (ADC_CONF[7:6]) to 2'b00. In this case, it needs to provide a reference voltage for ADC0 on this pin and the ADC0 channel 0

Pin	Type	MA35D16A087C (BGA 364-Ball, non-MCP) Pin Function
		analog input function will be disabled.
E10	A	EADC0_CH4
E11	A	EADC0_CH0
E12	I/O	PN.14 / UART6_RXD / CAN3_RXD / SPI1_SS1 / CCAP1_SFIELD / SPI1_I2SMCLK
E13	I/O	PK.13 / EPWM2_CH1 / UART1_nRTS / UART13_TXD / I2C4_SCL / I2S0_BCLK / SPI1_CLK / SC0_DAT / TM10_EXT
E14	I/O	PN.15 / EPWM2_CH4 / UART6_TXD / CAN3_TXD / I2S0_MCLK / SPI1_SS1 / SPI1_I2SMCLK / SC0_nCD / EADC0_ST / CLKO / TM6
E15	I/O	PK.12 / EPWM2_CH0 / UART1_nCTS / UART13_RXD / I2C4_SDA / I2S0_LRCK / SPI1_SS0 / SC0_CLK / TM10 / INT2
E16	I/O	PC.7 / CAN1_TXD / SD0_WP
E17	O	MBA1
E18	O	MBA2
E19	I/O	MD13
E20	O	MDMask1
F1	A	HSUSB1_D+
F2	A	HSUSB1_D-
F3	I	HSUSB0_ID
F4	I	RTC_nRWAKE
F5	I/O	PH.8 / TAMPER0
F16	I/O	PC.6 / CAN1_RXD / SD0_nCD **Note:** An external 10kΩ pull-down resistor is necessary on this pin when these PC.0~PC.5 pins (eMMC0_xxx) are used to connect with eMMC device and act as the booting source.
F17	O	MA10
F18	O	MA5
F19	I/O	MDQS1_P
F20	I/O	MDQS1_N
G1	A	HSUSB0_D+
G2	A	HSUSB0_D-
G3	I/O	PA.0 / UART1_nCTS / UART16_RXD / NAND_DATA0 / EBI_AD0 / EBI_ADR0
G4	I/O	PA.5 / UART3_nRTS / UART2_TXD / NAND_DATA5 / EBI_AD5 / EBI_ADR5
G5	I/O	PA.1 / UART1_nRTS / UART16_TXD / NAND_DATA1 / EBI_AD1 / EBI_ADR1
G7	P	V_{DDIO8}
G8	P	V_{DD_HSUSB1}
G9	P	V_{BAT}
G10	I	V_{REF_EADC0}
G11	P	AV_{DDH_PLL2}

Pin	Type	MA35D16A087C (BGA 364-Ball, non-MCP) Pin Function
G12	P	AV_{DDL_PLL2}
G13	P	V_{DDIO7}
G14	P	V_{DDIO6}
G16	P	V_{SS}
G17	O	MA3
G18	O	MA12
G19	I/O	MD9
G20	I/O	MD10
H1	I/O	PA.2 / UART1_RXD / NAND_DATA2 / EBI_AD2 / EBI_ADR2
H2	I/O	PA.4 / UART3_nCTS / UART2_RXD / NAND_DATA4 / EBI_AD4 / EBI_ADR4
H3	I/O	PA.9 / UART5_nRTS / UART4_TXD / NAND_nRE / EBI_AD9 / EBI_ADR9
H4	I/O	PA.6 / UART3_RXD / NAND_DATA6 / EBI_AD6 / EBI_ADR6
H5	I/O	PA.3 / UART1_TXD / NAND_DATA3 / EBI_AD3 / EBI_ADR3
H7	P	V_{DDIO9}
H8	P	V_{DD_HSUSB0}
H9	P	AV_{DD}
H10	P	AV_{DD_EADC0}
H11	P	AV_{DDH_PLL1}
H12	P	AV_{DDL_PLL1}
H13	P	AV_{DDL_PLL0}
H14	P	V_{DD_PLL1}
H16	O	MCS1n
H17	O	MA0
H18	O	MA1
H19	I/O	MD11
H20	I/O	MD8
J1	I/O	PA.11 / UART5_TXD / NAND_CLE / EBI_AD11 / EBI_ADR11
J2	I/O	PA.7 / UART3_TXD / NAND_DATA7 / EBI_AD7 / EBI_ADR7
J3	I/O	PA.13 / UART7_nRTS / UART8_TXD / NAND_nCS0 / EBI_AD13 / EBI_ADR13
J4	I/O	PD.13 / EPWM0_BRAKE1 / UART11_RXD / UART10_TXD / I2C4_SCL / TRACE_DATA1 / EBI_nCS2 / EBI_AD5 / ECAP0_IC0 / TM5_EXT / I2S1_BCLK
J5	I/O	PA.8 / UART5_nCTS / UART4_RXD / NAND_RDY0 / EBI_AD8 / EBI_ADR8
J7	P	V_{DD_OTP}
J8	P	V_{DD_CORE}
J9	P	AV_{DD_ADC0}

Pin	Type	MA35D16A087C (BGA 364-Ball, non-MCP) Pin Function
J10	P	AV_{SS}
J11	P	AV_{SS_PLL}
J12	P	V_{SS}
J13	P	V_{DD_CORE}
J14	P	V_{DDIO10}
J16	O	MCKE1
J17	O	MBA0
J18	O	MCASn
J19	I/O	MDQS0_P
J20	I/O	MDQS0_N
K1	I/O	PG.0 / EPWM0_CH0 / UART7_TXD / CAN3_TXD / SPI0_SS0 / EADC0_ST / EBI_AD15 / I2S1_MCLK / QEI0_INDEX / TM1 / CLKO / INT0 / EBI_ADR15
K2	I/O	PD.15 / EPWM0_SYNC_OUT / UART11_nRTS / CAN3_TXD / TRACE_DATA3 / EBI_ALE / EBI_AD7 / ECAP0_IC2 / TM6_EXT / I2S1_DO
K3	I/O	PD.12 / EPWM0_BRAKE0 / UART11_TXD / UART10_RXD / I2C4_SDA / TRACE_DATA0 / EBI_nCS1 / EBI_AD4 / QEI0_INDEX / TM5 / I2S1_LRCK / INT1
K4	I/O	PA.12 / UART7_nCTS / UART8_RXD / NAND_ALE / EBI_AD12 / EBI_ADR12
K5	I/O	PA.10 / UART5_RXD / NAND_nWE / EBI_AD10 / EBI_ADR10
K7	P	V_{DD_CORE}
K8	P	AV_{DDL_ROSC}
K9	P	V_{SS}
K10	P	V_{SS}
K11	P	V_{SS}
K12	P	V_{SS}
K13	P	MV_{DD}
K14	P	$MV_{DD_DPHYPLL}$
K16	O	MCKE0
K17	O	MRASn
K18	O	MWEn
K19	I/O	MD2
K20	I/O	MD3
L1	I/O	PG.1 / EPWM0_CH3 / UART9_nRTS / UART6_TXD / I2C4_SCL / CAN2_TXD / EBI_nCS0 / QEI0_B / TM1_EXT / RGMII1_PPS / RMII1_PPS
L2	I/O	PD.14 / EPWM0_SYNC_IN / UART11_nCTS / CAN3_RXD / TRACE_DATA2 / EBI_MCLK / EBI_AD6 / ECAP0_IC1 / TM6 / I2S1_DI / INT3
L3	I/O	PG.4 / EPWM1_CH0 / UART5_nCTS / UART6_RXD / SPI3_SS0 / QEI1_INDEX / EBI_ADR18 / EBI_nCS0 / I2S1_DO / SC1_CLK / TM4 / INT2 / ECAP1_IC2

Pin	Type	MA35D16A087C (BGA 364-Ball, non-MCP) Pin Function
L4	I/O	PG.2 / EPWM0_CH4 / UART9_RXD / CAN0_RXD / SPI0_SS1 / EBI_ADR16 / EBI_nCS2 / QEI0_A / TM3 / INT1
L5	I/O	PA.14 / UART7_RXD / CAN3_RXD / NAND_nWP / EBI_AD14 / EBI_ADR14
L7	P	V_{DD_CORE}
L8	P	V_{DD_CORE}
L9	P	V_{SS}
L10	P	V_{SS}
L11	P	V_{SS}
L12	P	V_{SS}
L13	P	MV_{DD}
L14	P	MV_{REF_DQ}
L16	O	MODT1
L17	O	MA2
L18	O	MA6
L19	I/O	MD1
L20	I/O	MD0
M1	I/O	PG.3 / EPWM0_CH5 / UART9_TXD / CAN0_TXD / SPI0_I2SMCLK / EBI_ADR17 / EBI_nCS1 / EBI_MCLK / QEI0_B / TM3_EXT / I2S1_MCLK
M2	I/O	PG.7 / EPWM1_CH3 / UART5_TXD / CAN1_TXD / SPI3_MISO / ECAP0_IC2 / EBI_nWR / I2S1_LRCK / SC1_PWR / TM7_EXT
M3	I/O	PM.12 / EPWM1_CH4 / UART10_nCTS / TRACE_DATA0 / UART11_RXD / I2C2_SDA / SC1_nCD / EBI_AD8 / I2S1_MCLK / TM8
M4	I/O	PG.6 / EPWM1_CH2 / UART5_RXD / CAN1_RXD / SPI3_MOSI / ECAP0_IC1 / EBI_nRD / I2S1_BCLK / SC1_RST / TM7 / INT3
M5	I/O	PA.15 / EPWM0_CH2 / UART9_nCTS / UART6_RXD / I2C4_SDA / CAN2_RXD / EBI_ALE / QEI0_A / TM1 / RGMII0_PPS / RMII0_PPS **Note:** This pin is dedicated as PMIC_nIRQ input function from external PMIC by default.
M7	P	V_{DD_CORE}
M8	P	V_{DD_CPU}
M9	P	V_{SS}
M10	P	V_{SS}
M11	P	V_{SS}
M12	P	V_{SS}
M13	P	MV_{DD}
M14	P	MV_{REF_CA}
M16	O	MODT0
M17	O	MA11
M18	O	MA4

Pin	Type	MA35D16A087C (BGA 364-Ball, non-MCP) Pin Function
M19	O	MCK_N
M20	O	MCK_P
N1	I/O	PG.5 / EPWM1_CH1 / UART5_nRTS / UART6_TXD / SPI3_CLK / ECAP0_IC0 / EBI_ADR19 / EBI_ALE / I2S1_DI / SC1_DAT / TM4_EXT
N2	I/O	PM.13 / EPWM1_CH5 / UART10_nRTS / TRACE_DATA1 / UART11_TXD / I2C2_SCL / EBI_AD9 / ECAP1_IC0 / TM8_EXT
N3	I/O	PG.13 / JTAG_TMS/SW_DIO / I2S0_BCLK / EBI_MCLK / EBI_AD2 **Note:** When the chip enters debug mode, this pin will be automatically forced to JTAG/SWD functionality by hardware, and other pinout functions of this pin will be disabled.
N4	I/O	PG.11 / JTAG_TDO / I2S0_MCLK / NAND_RDY1 / EBI_nWRH / EBI_nCS1 / EBI_AD0 **Note:** When the chip enters debug mode, this pin will be automatically forced to JTAG functionality by hardware, and other pinout functions of this pin will be disabled.
N5	I/O	PM.14 / EPWM1_BRAKE0 / UART10_RXD / TRACE_DATA2 / CAN2_RXD / I2C3_SDA / EBI_AD10 / ECAP1_IC1 / TM10 / INT1
N7	P	V_{DDIO0}
N8	P	V_{DD_CPU}
N9	P	V_{DD_CPU}
N10	P	V_{DD_CPU}
N11	P	V_{DD_CORE}
N12	P	V_{DD_CORE}
N13	P	MV_{DD}
N14	P	MZQ_DDRPHY
N16	O	MRESETn
N17	O	MA7
N18	O	MA9
N19	O	MDMask0
N20	I/O	MD6
P1	I/O	PM.15 / EPWM1_BRAKE1 / UART10_TXD / TRACE_DATA3 / CAN2_TXD / I2C3_SCL / EBI_AD11 / ECAP1_IC2 / TM10_EXT / INT2
P2	I/O	PG.14 / JTAG_TDI / I2S0_DI / NAND_nCS1 / EBI_ALE / EBI_AD3 **Note:** When the chip enters debug mode, this pin will be automatically forced to JTAG functionality by hardware, and other pinout functions of this pin will be disabled.
P3	I/O	PI.3 / EPWM0_CH3 / UART12_TXD / CAN0_TXD / SPI3_MISO / SC0_RST / EBI_ADR3 / TM1_EXT
P4	I/O	PI.1 / EPWM0_CH1 / UART12_nRTS / UART11_TXD / I2C2_SCL / SPI3_CLK / SC0_CLK / EBI_ADR1 / TM0_EXT / ECAP1_IC1
P5	I/O	PG.15 / JTAG_nTRST / I2S0_DO / EBI_nCS0 / EBI_AD4 **Note:** When the chip enters debug mode, this pin will be automatically forced to JTAG functionality by hardware, and other pinout functions of this pin will be disabled.
P7	P	V_{DDIO1}
P8	P	V_{DDIO2}

Pin	Type	MA35D16A087C (BGA 364-Ball, non-MCP) Pin Function
P9	P	V_{DD_CORE}
P10	P	V_{DD_CORE}
P11	P	V_{DD_CORE}
P12	P	V_{DDIO3}
P13	P	V_{DDIO4}
P14	P	V_{DDIO5}
P16	O	MCS0n
P17	O	MA15
P18	O	MA14
P19	I/O	MD7
P20	I/O	MD4
R1	I/O	PG.12 / JTAG_TCK/SW_CLK / I2S0_LRCK / EBI_nWRL / EBI_AD1 **Note:** When the chip enters debug mode, this pin will be automatically forced to JTAG/SWD functionality by hardware, and other pinout functions of this pin will be disabled.
R2	I/O	PI.4 / EPWM0_CH4 / UART14_nCTS / UART13_RXD / I2C3_SDA / SPI2_SS1 / I2S1_LRCK / EBI_ADR4 / INT0
R3	I/O	PI.0 / EPWM0_CH0 / UART12_nCTS / UART11_RXD / I2C2_SDA / SPI3_SS0 / SC0_nCD / EBI_ADR0 / TM0 / ECAP1_IC0
R4	I/O	PK.1 / EPWM0_SYNC_OUT / UART16_nRTS / UART15_TXD / I2C4_SCL / EADC0_ST / EBI_ADR9 / TM7_EXT / ECAP0_IC2
R5	I/O	PK.3 / EPWM1_CH1 / UART16_TXD / CAN2_TXD / SPI3_SS1 / SC1_nCD / EBI_ADR11 / QEI0_B
R16	P	V_{SS}
R17	O	MA13
R18	O	MA8
R19	P	V_{SS}
R20	I/O	MD5
T1	I/O	PI.2 / EPWM0_CH2 / UART12_RXD / CAN0_RXD / SPI3_MOSI / SC0_DAT / EBI_ADR2 / TM1 / ECAP1_IC2
T2	I/O	PJ.14 / EPWM1_CH4 / UART2_RXD / CAN3_RXD / SPI3_MISO / SC1_RST / EBI_ADR14 / TM3
T3	I/O	PI.5 / EPWM0_CH5 / UART14_nRTS / UART13_TXD / I2C3_SCL / I2S1_BCLK / EBI_ADR5 / INT1
T4	I/O	PJ.15 / EPWM1_CH5 / UART2_TXD / CAN3_TXD / SPI3_CLK / EADC0_ST / SC1_PWR / EBI_ADR15 / TM3_EXT / INT1
T5	I/O	PJ.13 / EPWM1_CH3 / UART2_nRTS / UART1_TXD / I2C5_SCL / SPI3_MOSI / SC1_DAT / EBI_ADR13 / TM2_EXT
T6	I/O	PL.7 / EPWM0_CH1 / UART2_TXD / CAN0_TXD / QSPI1_MISO1 / EBI_AD6 / TM3_EXT / ECAP1_IC1 / INT1
T7	I/O	PD.8 / EPWM0_BRAKE0 / UART16_nCTS / UART15_RXD / QSPI1_SS0 / I2S1_LRCK / EPWM0_CH2 / EBI_AD7 / SC1_CLK / TM0
T8	I/O	PL.1 / EPWM1_CH1 / UART11_nRTS / UART10_TXD / I2C3_SCL / SPI2_MISO / QSPI1_MISO1 / I2S0_BCLK / EBI_AD12 / SC1_DAT / TM5_EXT / QEI1_B

Pin	Type	MA35D16A087C (BGA 364-Ball, non-MCP) Pin Function
T9	I/O	PM.1 / I2C4_SCL / SPI3_I2SMCLK / CCAP0_SFIELD / EBI_AD4 / EBI_ADR4
T10	I/O	PL.0 / EPWM1_CH0 / UART11_nCTS / UART10_RXD / I2C3_SDA / SPI2_MOSI / QSPI1_MOSI1 / I2S0_LRCK / EBI_AD11 / SC1_CLK / TM5 / QEI1_A
T11	I/O	PL.5 / EPWM1_CH5 / UART2_nRTS / UART1_TXD / I2C4_SCL / SPI3_MISO / QSPI1_MISO0 / I2S1_MCLK / EBI_nWR / SC0_nCD / TM9_EXT / ECAP0_IC2
T12	I/O	PJ.4 / I2C3_SDA / SD1_WP
T13	I/O	PJ.5 / I2C3_SCL / SD1_nCD **Note:** An external 10kΩ pull-down resistor is necessary on this pin when these PJ.0~PJ.3 and PJ.6~PJ.11 pins (eMMC1_xxx) are used to connect with eMMC device and act as the booting source.
T14	I/O	PK.6 / EPWM1_CH2 / UART12_RXD / CAN0_RXD / SPI2_MOSI / I2S1_BCLK / SC0_RST / TM6 / INT2
T15	I/O	PK.5 / EPWM1_CH1 / UART12_nRTS / UART13_TXD / I2C4_SCL / SPI2_CLK / I2S1_DI / SC0_DAT / EADC0_ST / TM8_EXT / INT1
T16	I/O	PK.7 / EPWM1_CH3 / UART12_TXD / CAN0_TXD / SPI2_MISO / I2S1_LRCK / SC0_PWR / CLKO / TM6_EXT / INT3
T17	P	V_{SS}
T18	P	V_{SS}
T19	I/O	PD.0 / UART3_nCTS / UART4_RXD / QSPI0_SS0
T20	P	V_{SS}
U1	I/O	PI.6 / EPWM0_BRAKE0 / UART14_RXD / CAN1_RXD / I2S1_DI / EBI_ADR6 / QEI1_INDEX / INT2
U2	I/O	PJ.12 / EPWM1_CH2 / UART2_nCTS / UART1_RXD / I2C5_SDA / SPI3_SS0 / SC1_CLK / EBI_ADR12 / TM2 / QEI0_INDEX
U3	I/O	PI.7 / EPWM0_BRAKE1 / UART14_TXD / CAN1_TXD / I2S1_DO / EBI_ADR7 / ECAP0_IC0 / INT3
U4	I/O	PL.9 / EPWM0_CH3 / UART14_nRTS / UART13_TXD / I2C5_SCL / SPI3_CLK / EPWM1_CH4 / I2S1_BCLK / EBI_AD8 / SC0_DAT / TM4_EXT / QEI0_A / INT3
U5	I/O	PL.11 / EPWM0_CH5 / UART14_TXD / CAN3_TXD / SPI3_MISO / EPWM1_CH5 / I2S1_DO / EBI_AD10 / SC0_PWR / EBI_nWRL / QEI0_INDEX
U6	I/O	PE.15 / UART0_RXD **Note:** This GPIO PE.15 (UART0_RXD) pin must be pulled to high level through an external 10kΩ resistor or an internal pull-up resistor in the external device (such as a transceiver or MCU).
U7	I/O	PM.11 / EPWM1_CH3 / CAN2_TXD / SPI3_SS1 / CCAP0_DATA9 / SPI2_SS1 / EBI_AD14 / EBI_ADR14
U8	I/O	PK.11 / CAN1_TXD / CCAP0_HSYNC / EBI_AD2 / EBI_ADR2
U9	I/O	PM.3 / CAN3_TXD / CCAP0_DATA1 / EBI_AD6 / EBI_ADR6
U10	I/O	PM.5 / I2C5_SCL / CCAP0_DATA3 / EBI_AD8 / EBI_ADR8
U11	I/O	PM.9 / I2C0_SCL / CCAP0_DATA7 / EBI_AD12 / EBI_ADR12
U12	I/O	PJ.9 / I2C4_SCL / SD1_DAT1/eMMC1_DAT1
U13	I/O	PG.9 / EPWM1_CH5 / UART12_TXD / CAN3_TXD / SPI2_CLK / LCM_HSYNC/LCM_MPU_WR/RW / I2C3_SCL / EBI_AD8 / EBI_nCS1
U14	I/O	PI.10 / UART4_RXD / LCM_DATA2/LCM_MPU_D2 / EBI_AD13
U15	I/O	PH.0 / UART8_nCTS / UART7_RXD / LCM_DATA8/LCM_MPU_D8

Pin	Type	MA35D16A087C (BGA 364-Ball, non-MCP) Pin Function
U16	I/O	PH.7 / UART10_TXD / LCM_DATA15/LCM_MPU_D15
U17	I/O	PC.12 / UART12_nCTS / UART11_RXD / LCM_DATA16/LCM_MPU_D16
U18	I/O	PH.13 / UART14_nRTS / UART13_TXD / LCM_DATA21
U19	I/O	PD.5 / UART1_nRTS / UART2_TXD / I2C2_SCL / QSPI0_MISO1
U20	I/O	PD.3 / UART3_TXD / QSPI0_MISO0
V1	I/O	PK.0 / EPWM0_SYNC_IN / UART16_nCTS / UART15_RXD / I2C4_SDA / I2S1_MCLK / EBI_ADR8 / TM7 / ECAP0_IC1
V2	I/O	PK.2 / EPWM1_CH0 / UART16_RXD / CAN2_RXD / SPI3_I2SMCLK / SC0_PWR / EBI_ADR10 / QEI0_A
V3	I/O	PD.9 / EPWM0_BRAKE1 / UART16_nRTS / UART15_TXD / QSPI1_CLK / I2S1_BCLK / EPWM0_CH3 / EBI_AD8 / SC1_DAT / TM0_EXT
V4	I/O	PD.11 / EPWM1_BRAKE1 / UART16_TXD / QSPI1_MISO0 / I2S1_DO / EPWM0_CH5 / EBI_AD10 / SC1_PWR / TM2_EXT
V5	I/O	PL.3 / EPWM1_CH3 / UART11_TXD / CAN3_TXD / SPI2_CLK / QSPI1_CLK / I2S0_DO / EBI_AD14 / SC1_PWR / TM7_EXT / ECAP0_IC0
V6	I/O	PK.9 / I2C3_SCL / CCAP0_SCLK / EBI_AD0 / EBI_ADR0
V7	I/O	PM.7 / CAN0_TXD / CCAP0_DATA5 / EBI_AD10 / EBI_ADR10
V8	I/O	PM.2 / CAN3_RXD / CCAP0_DATA0 / EBI_AD5 / EBI_ADR5
V9	I/O	PM.4 / I2C5_SDA / CCAP0_DATA2 / EBI_AD7 / EBI_ADR7
V10	I/O	PM.6 / CAN0_RXD / CCAP0_DATA4 / EBI_AD9 / EBI_ADR9
V11	I/O	PJ.7 / CAN3_TXD / SD1_CLK/eMMC1_CLK
V12	I/O	PI.8 / UART4_nCTS / UART3_RXD / LCM_DATA0/LCM_MPU_D0 / EBI_AD11
V13	I/O	PK.4 / UART12_nCTS / UART13_RXD / SPI2_MISO / LCM_DEN/LCM_MPU_RS / EBI_AD10 / EBI_nWRL
V14	I/O	PI.12 / UART6_nCTS / UART5_RXD / LCM_DATA4/LCM_MPU_D4
V15	I/O	PI.14 / UART6_RXD / LCM_DATA6/LCM_MPU_D6
V16	I/O	PH.5 / UART10_nRTS / UART9_TXD / LCM_DATA13/LCM_MPU_D13
V17	I/O	PC.15 / UART12_TXD / LCM_DATA19 / LCM_MPU_TE / LCM_MPU_VSYNC
V18	I/O	PD.2 / UART3_RXD / QSPI0_MOSI0
V19	I/O	PH.12 / UART14_nCTS / UART13_RXD / LCM_DATA20
V20	I/O	PD.1 / UART3_nRTS / UART4_TXD / QSPI0_CLK
W1	I/O	PE.14 / UART0_TXD
W2	I/O	PL.6 / EPWM0_CH0 / UART2_RXD / CAN0_RXD / QSPI1_MOSI1 / TRACE_CLK / EBI_AD5 / TM3 / ECAP1_IC0 / INT0
W3	I/O	PL.8 / EPWM0_CH2 / UART14_nCTS / UART13_RXD / I2C5_SDA / SPI3_SS0 / EPWM0_CH4 / I2S1_LRCK / EBI_AD7 / SC0_CLK / TM4 / ECAP1_IC2 / INT2
W4	I/O	PD.6 / EPWM0_SYNC_IN / UART1_RXD / QSPI1_MOSI1 / I2C0_SDA / I2S0_MCLK / EPWM0_CH0 / EBI_AD5 / SPI3_SS1 / TRACE_CLK **Note:** This pin is dedicated as I2C0_SDA function for controlling external PMIC by default.

Pin	Type	MA35D16A087C (BGA 364-Ball, non-MCP) Pin Function
W5	I/O	PL.2 / EPWM1_CH2 / UART11_RXD / CAN3_RXD / SPI2_SS0 / QSPI1_SS1 / I2S0_DI / EBI_AD13 / SC1_RST / TM7 / QEI1_INDEX
W6	I/O	PL.4 / EPWM1_CH4 / UART2_nCTS / UART1_RXD / I2C4_SDA / SPI3_MOSI / QSPI1_MOSI0 / I2S0_MCLK / EBI_nRD / SC1_nCD / TM9 / ECAP0_IC1
W7	I/O	PM.0 / I2C4_SDA / CCAP0_VSYNC / EBI_AD3 / EBI_ADR3
W8	I/O	PM.10 / EPWM1_CH2 / CAN2_RXD / SPI3_SS0 / CCAP0_DATA8 / SPI2_I2SMCLK / EBI_AD13 / EBI_ADR13
W9	I/O	PJ.1 / EPWM1_BRAKE1 / UART8_nRTS / UART7_TXD / I2C2_SCL / SPI2_CLK / eMMC1_DAT5 / I2S0_BCLK / SC0_DAT / EBI_AD12 / EBI_ADR17 / EBI_nCS1 / EBI_AD8
W10	I/O	PJ.11 / CAN0_TXD / SD1_DAT3/eMMC1_DAT3
W11	I/O	PJ.6 / CAN3_RXD / SD1_CMD/eMMC1_CMD
W12	I/O	PJ.10 / CAN0_RXD / SD1_DAT2/eMMC1_DAT2
W13	I/O	PG.10 / UART12_nRTS / UART13_TXD / SPI2_MOSI / LCM_CLK / EBI_AD9 / EBI_nWRH
W14	I/O	PC.14 / UART12_RXD / LCM_DATA18/LCM_MPU_CS
W15	I/O	PI.13 / UART6_nRTS / UART5_TXD / LCM_DATA5/LCM_MPU_D5
W16	I/O	PH.1 / UART8_nRTS / UART7_TXD / LCM_DATA9/LCM_MPU_D9
W17	I/O	PH.6 / UART10_RXD / LCM_DATA14/LCM_MPU_D14
W18	I/O	PC.13 / UART12_nRTS / UART11_TXD / LCM_DATA17/LCM_MPU_D17
W19	I/O	PH.15 / UART14_TXD / LCM_DATA23
W20	I/O	PH.14 / UART14_RXD / LCM_DATA22
Y1	P	V_{SS}
Y2	I/O	PL.10 / EPWM0_CH4 / UART14_RXD / CAN3_RXD / SPI3_MOSI / EPWM0_CH5 / I2S1_DI / EBI_AD9 / SC0_RST / EBI_nWRH / QEI0_B
Y3	I/O	PD.7 / EPWM0_SYNC_OUT / UART1_TXD / QSPI1_MISO1 / I2C0_SCL / I2S1_MCLK / EPWM0_CH1 / EBI_AD6 / SC1_nCD / EADC0_ST **Note:** This pin is dedicated as I2C0_SCL function for controlling external PMIC by default.
Y4	I/O	PD.10 / EPWM1_BRAKE0 / UART16_RXD / QSPI1_MOSI0 / I2S1_DI / EPWM0_CH4 / EBI_AD9 / SC1_RST / TM2
Y5	I/O	PK.8 / EPWM1_CH0 / I2C3_SDA / SPI3_CLK / EADC0_ST / EBI_AD15 / EBI_MCLK / EBI_ADR15 / TM8 / QEI1_INDEX
Y6	I/O	PK.10 / CAN1_RXD / CCAP0_PIXCLK / EBI_AD1 / EBI_ADR1
Y7	I/O	PM.8 / I2C0_SDA / CCAP0_DATA6 / EBI_AD11 / EBI_ADR11
Y8	I/O	PJ.3 / EPWM1_CH5 / UART8_TXD / CAN1_TXD / SPI2_MISO / eMMC1_DAT7 / I2S0_DO / SC0_PWR / EBI_AD14 / EBI_ADR19 / EBI_nWRL / EBI_AD10
Y9	I/O	PJ.0 / EPWM1_BRAKE0 / UART8_nCTS / UART7_RXD / I2C2_SDA / SPI2_SS0 / eMMC1_DAT4 / I2S0_LRCK / SC0_CLK / EBI_AD11 / EBI_ADR16 / EBI_nCS0 / EBI_AD7
Y10	I/O	PJ.2 / EPWM1_CH4 / UART8_RXD / CAN1_RXD / SPI2_MOSI / eMMC1_DAT6 / I2S0_DI / SC0_RST / EBI_AD13 / EBI_ADR18 / EBI_nWRH / EBI_AD9
Y11	I/O	PJ.8 / I2C4_SDA / SD1_DAT0/eMMC1_DAT0
Y12	I/O	PG.8 / EPWM1_CH4 / UART12_RXD / CAN3_RXD / SPI2_SS0 / LCM_VSYNC/LCM_MPU_RD/EN / I2C3_SDA / EBI_AD7 / EBI_nCS0

Pin	Type	MA35D16A087C (BGA 364-Ball, non-MCP) Pin Function
Y13	I/O	PH.2 / UART8_RXD / LCM_DATA10/LCM_MPU_D10
Y14	I/O	PI.9 / UART4_nRTS / UART3_TXD / LCM_DATA1/LCM_MPU_D1 / EBI_AD12
Y15	I/O	PI.11 / UART4_TXD / LCM_DATA3/LCM_MPU_D3 / EBI_AD14
Y16	I/O	PI.15 / UART6_TXD / LCM_DATA7/LCM_MPU_D7
Y17	I/O	PH.3 / UART8_TXD / LCM_DATA11/LCM_MPU_D11
Y18	I/O	PH.4 / UART10_nCTS / UART9_RXD / LCM_DATA12/LCM_MPU_D12
Y19	I/O	PD.4 / UART1_nCTS / UART2_RXD / I2C2_SDA / QSPI0_MOSI1
Y20	P	V_{SS}

4.2.4 MA35D1 Series Multi-function Description Table

Group	Pin Name	Type	Description
ADC0	ADC0_CH0	A	ADC0 channel 0 analog input. (VREF) **Note:** This ADC0_CH0 (PB.8 pin) has another function, which is that it can be used as the reference voltage (V_{REF}) input of ADC0 by setting the control register REFSEL (ADC_CONF[7:6]) to 2'b00. In this case, it needs to provide a reference voltage for ADC0 on this pin and the ADC0 channel 0 analog input function will be disabled.
	ADC0_CH1	A	ADC0 channel 1 analog input. (A1)
	ADC0_CH2	A	ADC0 channel 2 analog input. (A2)
	ADC0_CH3	A	ADC0 channel 3 analog input. (VSENSE)
	ADC0_CH4	A	ADC0 channel 4 analog input. (YM)
	ADC0_CH5	A	ADC0 channel 5 analog input. (YP)
	ADC0_CH6	A	ADC0 channel 6 analog input. (XM)
	ADC0_CH7	A	ADC0 channel 7 analog input. (XP)
CAN0	CAN0_RXD	I	CAN0 bus receiver input.
	CAN0_TXD	O	CAN0 bus transmitter output.
CAN1	CAN1_RXD	I	CAN1 bus receiver input.
	CAN1_TXD	O	CAN1 bus transmitter output.
CAN2	CAN2_RXD	I	CAN2 bus receiver input.
	CAN2_TXD	O	CAN2 bus transmitter output.
CAN3	CAN3_RXD	I	CAN3 bus receiver input.
	CAN3_TXD	O	CAN3 bus transmitter output.
CCAP0	CCAP0_DATA0	I	Video Image/Camera capture 0 data input bus bit 0. (YUV DATA [0])
	CCAP0_DATA1	I	Video Image/Camera capture 0 data input bus bit 1. (YUV DATA [1])
	CCAP0_DATA2	I	Video Image/Camera capture 0 data input bus bit 2. (YUV DATA [2])
	CCAP0_DATA3	I	Video Image/Camera capture 0 data input bus bit 3. (YUV DATA [3])
	CCAP0_DATA4	I	Video Image/Camera capture 0 data input bus bit 4. (YUV DATA [4])
	CCAP0_DATA5	I	Video Image/Camera capture 0 data input bus bit 5. (YUV DATA [5])
	CCAP0_DATA6	I	Video Image/Camera capture 0 data input bus bit 6. (YUV DATA [6])
	CCAP0_DATA7	I	Video Image/Camera capture 0 data input bus bit 7. (YUV DATA [7])
	CCAP0_DATA8	I	Video Image/Camera capture 0 data input bus bit 8.
	CCAP0_DATA9	I	Video Image/Camera capture 0 data input bus bit 9.
	CCAP0_HSYNC	I	Video Image/Camera capture 0 interface HSYNC input pin.
	CCAP0_PIXCLK	I	Video Image/Camera capture 0 interface pixel clock input pin.
	CCAP0_SCLK	O	Video Image/Camera capture 0 interface sensor clock output pin.
	CCAP0_SFIELD	I	Video Image/Camera capture 0 interface SFIELD input pin.
	CCAP0_VSYNC	I	Video Image/Camera capture 0 interface VSYNC input pin.

Group	Pin Name	Type	Description
CCAP1	CCAP1_DATA0	I	Video Image/Camera capture 1 data input bus bit 0. (YUV DATA [0])
	CCAP1_DATA1	I	Video Image/Camera capture 1 data input bus bit 1. (YUV DATA [1])
	CCAP1_DATA2	I	Video Image/Camera capture 1 data input bus bit 2. (YUV DATA [2])
	CCAP1_DATA3	I	Video Image/Camera capture 1 data input bus bit 3. (YUV DATA [3])
	CCAP1_DATA4	I	Video Image/Camera capture 1 data input bus bit 4. (YUV DATA [4])
	CCAP1_DATA5	I	Video Image/Camera capture 1 data input bus bit 5. (YUV DATA [5])
	CCAP1_DATA6	I	Video Image/Camera capture 1 data input bus bit 6. (YUV DATA [6])
	CCAP1_DATA7	I	Video Image/Camera capture 1 data input bus bit 7. (YUV DATA[7])
	CCAP1_DATA8	I	Video Image/Camera capture 1 data input bus bit 8.
	CCAP1_DATA9	I	Video Image/Camera capture 1 data input bus bit 9.
	CCAP1_HSYNC	I	Video Image/Camera capture 1 interface HSYNC input pin.
	CCAP1_PIXCLK	I	Video Image/Camera capture 1 interface pixel clock input pin.
	CCAP1_SCLK	O	Video Image/Camera capture 1 interface sensor clock output pin.
	CCAP1_SFIELD	I	Video Image/Camera capture 1 interface SFIELD input pin.
	CCAP1_VSYNC	I	Video Image/Camera capture 1 interface VSYNC input pin.
CLKO	CLKO	O	Clock output pin.
CLK	CLK_32KOUT	O	32 kHz clock output pin.
EADC0	EADC0_CH0	A	EADC0 channel 0 analog input.
	EADC0_CH1	A	EADC0 channel 1 analog input.
	EADC0_CH2	A	EADC0 channel 2 analog input.
	EADC0_CH3	A	EADC0 channel 3 analog input.
	EADC0_CH4	A	EADC0 channel 4 analog input.
	EADC0_CH5	A	EADC0 channel 5 analog input.
	EADC0_CH6	A	EADC0 channel 6 analog input.
	EADC0_CH7	A	EADC0 channel 7 analog input.
	EADC0_ST	I	EADC0 external trigger input.
EBI	EBI_AD0	I/O	EBI address/data bus bit 0.
	EBI_AD1	I/O	EBI address/data bus bit 1.
	EBI_AD2	I/O	EBI address/data bus bit 2.
	EBI_AD3	I/O	EBI address/data bus bit 3.
	EBI_AD4	I/O	EBI address/data bus bit 4.
	EBI_AD5	I/O	EBI address/data bus bit 5.
	EBI_AD6	I/O	EBI address/data bus bit 6.
	EBI_AD7	I/O	EBI address/data bus bit 7.

Group	Pin Name	Type	Description
	EBI_AD8	I/O	EBI address/data bus bit 8.
	EBI_AD9	I/O	EBI address/data bus bit 9.
	EBI_AD10	I/O	EBI address/data bus bit 10.
	EBI_AD11	I/O	EBI address/data bus bit 11.
	EBI_AD12	I/O	EBI address/data bus bit 12.
	EBI_AD13	I/O	EBI address/data bus bit 13.
	EBI_AD14	I/O	EBI address/data bus bit 14.
	EBI_AD15	I/O	EBI address/data bus bit 15.
	EBI_ADR0	O	EBI address bus bit 0.
	EBI_ADR1	O	EBI address bus bit 1.
	EBI_ADR2	O	EBI address bus bit 2.
	EBI_ADR3	O	EBI address bus bit 3.
	EBI_ADR4	O	EBI address bus bit 4.
	EBI_ADR5	O	EBI address bus bit 5.
	EBI_ADR6	O	EBI address bus bit 6.
	EBI_ADR7	O	EBI address bus bit 7.
	EBI_ADR8	O	EBI address bus bit 8.
	EBI_ADR9	O	EBI address bus bit 9.
	EBI_ADR10	O	EBI address bus bit 10.
	EBI_ADR11	O	EBI address bus bit 11.
	EBI_ADR12	O	EBI address bus bit 12.
	EBI_ADR13	O	EBI address bus bit 13.
	EBI_ADR14	O	EBI address bus bit 14.
	EBI_ADR15	O	EBI address bus bit 15.
	EBI_ADR16	O	EBI address bus bit 16.
	EBI_ADR17	O	EBI address bus bit 17.
	EBI_ADR18	O	EBI address bus bit 18.
	EBI_ADR19	O	EBI address bus bit 19.
	EBI_ALE	O	EBI address latch enable output pin.
	EBI_MCLK	O	EBI external clock output pin.
	EBI_nCS0	O	EBI chip select 0 output pin.
	EBI_nCS1	O	EBI chip select 1 output pin.
	EBI_nCS2	O	EBI chip select 2 output pin.
	EBI_nRD	O	EBI read enable output pin.

Group	Pin Name	Type	Description
	EBI_nWR	O	EBI write enable output pin.
	EBI_nWRH	O	EBI high byte write enable output pin.
	EBI_nWRL	O	EBI low byte write enable output pin.
ECAP0	ECAP0_IC0	I	Enhanced capture unit 0 input 0 pin.
	ECAP0_IC1	I	Enhanced capture unit 0 input 1 pin.
	ECAP0_IC2	I	Enhanced capture unit 0 input 2 pin.
ECAP1	ECAP1_IC0	I	Enhanced capture unit 1 input 0 pin.
	ECAP1_IC1	I	Enhanced capture unit 1 input 1 pin.
	ECAP1_IC2	I	Enhanced capture unit 1 input 2 pin.
ECAP2	ECAP2_IC0	I	Enhanced capture unit 2 input 0 pin.
	ECAP2_IC1	I	Enhanced capture unit 2 input 1 pin.
	ECAP2_IC2	I	Enhanced capture unit 2 input 2 pin.
EPWM0	EPWM0_BRAKE0	I	EPWM0 Brake 0 input pin.
	EPWM0_BRAKE1	I	EPWM0 Brake 1 input pin.
	EPWM0_CH0	I/O	EPWM0 channel 0 output/capture input.
	EPWM0_CH1	I/O	EPWM0 channel 1 output/capture input.
	EPWM0_CH2	I/O	EPWM0 channel 2 output/capture input.
	EPWM0_CH3	I/O	EPWM0 channel 3 output/capture input.
	EPWM0_CH4	I/O	EPWM0 channel 4 output/capture input.
	EPWM0_CH5	I/O	EPWM0 channel 5 output/capture input.
	EPWM0_SYNC_IN	I	EPWM0 counter synchronous trigger input pin.
	EPWM0_SYNC_OUT	O	EPWM0 counter synchronous trigger output pin.
EPWM1	EPWM1_BRAKE0	I	EPWM1 Brake 0 input pin.
	EPWM1_BRAKE1	I	EPWM1 Brake 1 input pin.
	EPWM1_CH0	I/O	EPWM1 channel 0 output/capture input.
	EPWM1_CH1	I/O	EPWM1 channel 1 output/capture input.
	EPWM1_CH2	I/O	EPWM1 channel 2 output/capture input.
	EPWM1_CH3	I/O	EPWM1 channel 3 output/capture input.
	EPWM1_CH4	I/O	EPWM1 channel 4 output/capture input.
	EPWM1_CH5	I/O	EPWM1 channel 5 output/capture input.
EPWM2	EPWM2_BRAKE0	I	EPWM2 Brake 0 input pin.
	EPWM2_BRAKE1	I	EPWM2 Brake 1 input pin.
	EPWM2_CH0	I/O	EPWM2 channel 0 output/capture input.
	EPWM2_CH1	I/O	EPWM2 channel 1 output/capture input.

Group	Pin Name	Type	Description
	EPWM2_CH2	I/O	EPWM2 channel 2 output/capture input.
	EPWM2_CH3	I/O	EPWM2 channel 3 output/capture input.
	EPWM2_CH4	I/O	EPWM2 channel 4 output/capture input.
	EPWM2_CH5	I/O	EPWM2 channel 5 output/capture input.
HSUSB0	HSUSB0_VBUSVLD	I	HSUSB0 external V_{BUS} status pin.
HSUSBH	HSUSBH_OVC	I	HSUSB host bus power overcurrent detector pin.
	HSUSBH_PWREN	O	HSUSB host external V_{BUS} regulator enable pin.
I2C0	I2C0_SCL	I/O	I2C0 clock pin.
	I2C0_SDA	I/O	I2C0 data input/output pin.
I2C1	I2C1_SCL	I/O	I2C1 clock pin.
	I2C1_SDA	I/O	I2C1 data input/output pin.
I2C2	I2C2_SCL	I/O	I2C2 clock pin.
	I2C2_SDA	I/O	I2C2 data input/output pin.
I2C3	I2C3_SCL	I/O	I2C3 clock pin.
	I2C3_SDA	I/O	I2C3 data input/output pin.
I2C4	I2C4_SCL	I/O	I2C4 clock pin.
	I2C4_SDA	I/O	I2C4 data input/output pin.
I2C5	I2C5_SCL	I/O	I2C5 clock pin.
	I2C5_SDA	I/O	I2C5 data input/output pin.
I2S0	I2S0_BCLK	O	I2S0 bit clock output pin.
	I2S0_DI	I	I2S0 data input pin.
	I2S0_DO	O	I2S0 data output pin.
	I2S0_LRCK	O	I2S0 left right channel clock output pin.
	I2S0_MCLK	O	I2S0 master clock output pin.
I2S1	I2S1_BCLK	O	I2S1 bit clock output pin.
	I2S1_DI	I	I2S1 data input pin.
	I2S1_DO	O	I2S1 data output pin.
	I2S1_LRCK	O	I2S1 left right channel clock output pin.
	I2S1_MCLK	O	I2S1 master clock output pin.
INT0	INT0	I	External interrupt 0 input pin.
INT1	INT1	I	External interrupt 1 input pin.
INT2	INT2	I	External interrupt 2 input pin.
INT3	INT3	I	External interrupt 3 input pin.
JTAG	JTAG_TCK/SW_CLK	I	JTAG clock input pin. Serial wire clock input pin.

Group	Pin Name	Type	Description
	JTAG_TDI	I	JTAG data input pin.
	JTAG_TDO	O	JTAG data output pin.
	JTAG_TMS/SW_DIO	I I/O	JTAG test mode selection input pin. Serial wire data pin.
	JTAG_nTRST	I	JTAG reset input pin.
KPI	KPI_COL0	I	Keypad Interface Column 0 input pin.
	KPI_COL1	I	Keypad Interface Column 1 input pin.
	KPI_COL2	I	Keypad Interface Column 2 input pin.
	KPI_COL3	I	Keypad Interface Column 3 input pin.
	KPI_COL4	I	Keypad Interface Column 4 input pin.
	KPI_COL5	I	Keypad Interface Column 5 input pin.
	KPI_COL6	I	Keypad Interface Column 6 input pin.
	KPI_COL7	I	Keypad Interface Column 7 input pin.
	KPI_ROW0	O	Keypad Interface Row 0 output pin.
	KPI_ROW1	O	Keypad Interface Row 1 output pin.
	KPI_ROW2	O	Keypad Interface Row 2 output pin.
	KPI_ROW3	O	Keypad Interface Row 3 output pin.
	KPI_ROW4	O	Keypad Interface Row 4 output pin.
	KPI_ROW5	O	Keypad Interface Row 5 output pin.
	KPI_ROW6	O	Keypad Interface Row 6 output pin.
	KPI_ROW7	O	Keypad Interface Row 7 output pin.
LCM	LCM_CLK	O	TFT LCD Module Pixel Clock output pin in Sync-type mode.
	LCM_DATA0/LCM_MPU_D0	I/O	TFT LCD Module Pixel Data output bit 0 in Sync-type mode. TFT LCD Module Command/Data input/output bit 0 in MPU-type mode.
	LCM_DATA1/LCM_MPU_D1	I/O	TFT LCD Module Pixel Data output bit 1 in Sync-type mode. TFT LCD Module Command/Data input/output bit 1 in MPU-type mode.
	LCM_DATA10/LCM_MPU_D10	I/O	TFT LCD Module Pixel Data output bit 10 in Sync-type mode. TFT LCD Module Command/Data input/output bit 10 in MPU-type mode.
	LCM_DATA11/LCM_MPU_D11	I/O	TFT LCD Module Pixel Data output bit 11 in Sync-type mode. TFT LCD Module Command/Data input/output bit 11 in MPU-type mode.
	LCM_DATA12/LCM_MPU_D12	I/O	TFT LCD Module Pixel Data output bit 12 in Sync-type mode. TFT LCD Module Command/Data input/output bit 12 in MPU-type mode.
	LCM_DATA13/LCM_MPU_D13	I/O	TFT LCD Module Pixel Data output bit 13 in Sync-type mode. TFT LCD Module Command/Data input/output bit 13 in MPU-type mode.
	LCM_DATA14/LCM_MPU_D14	I/O	TFT LCD Module Pixel Data output bit 14 in Sync-type mode. TFT LCD Module Command/Data input/output bit 14 in MPU-type mode.
	LCM_DATA15/LCM_M	I/O	TFT LCD Module Pixel Data output bit 15 in Sync-type mode.

Group	Pin Name	Type	Description
	PU_D15		TFT LCD Module Command/Data input/output bit 15 in MPU-type mode.
	LCM_DATA16/LCM_MPU_D16	I/O	TFT LCD Module Pixel Data output bit 16 in Sync-type mode. TFT LCD Module Command/Data input/output bit 16 in MPU-type mode.
	LCM_DATA17/LCM_MPU_D17	I/O	TFT LCD Module Pixel Data output bit 17 in Sync-type mode. TFT LCD Module Command/Data input/output bit 17 in MPU-type mode.
	LCM_DATA18/LCM_MPU_CS	O	TFT LCD Module Pixel Data output bit 18 in Sync-type mode. TFT LCD Module Chip Select output pin in MPU-type mode.
	LCM_DATA19	O	TFT LCD Module Pixel Data output bit 19 in Sync-type mode.
	LCM_DATA2/LCM_MPU_D2	I/O	TFT LCD Module Pixel Data output bit 2 in Sync-type mode. TFT LCD Module Command/Data input/output bit 2 in MPU-type mode.
	LCM_DATA20	O	TFT LCD Module Pixel Data output bit 20 in Sync-type mode.
	LCM_DATA21	O	TFT LCD Module Pixel Data output bit 21 in Sync-type mode.
	LCM_DATA22	O	TFT LCD Module Pixel Data output bit 22 in Sync-type mode.
	LCM_DATA23	O	TFT LCD Module Pixel Data output bit 23 in Sync-type mode.
	LCM_DATA3/LCM_MPU_D3	I/O	TFT LCD Module Pixel Data output bit 3 in Sync-type mode. TFT LCD Module Command/Data input/output bit 3 in MPU-type mode.
	LCM_DATA4/LCM_MPU_D4	I/O	TFT LCD Module Pixel Data output bit 4 in Sync-type mode. TFT LCD Module Command/Data input/output bit 4 in MPU-type mode.
	LCM_DATA5/LCM_MPU_D5	I/O	TFT LCD Module Pixel Data output bit 5 in Sync-type mode. TFT LCD Module Command/Data input/output bit 5 in MPU-type mode.
	LCM_DATA6/LCM_MPU_D6	I/O	TFT LCD Module Pixel Data output bit 6 in Sync-type mode. TFT LCD Module Command/Data input/output bit 6 in MPU-type mode.
	LCM_DATA7/LCM_MPU_D7	I/O	TFT LCD Module Pixel Data output bit 7 in Sync-type mode. TFT LCD Module Command/Data input/output bit 7 in MPU-type mode.
	LCM_DATA8/LCM_MPU_D8	I/O	TFT LCD Module Pixel Data output bit 8 in Sync-type mode. TFT LCD Module Command/Data input/output bit 8 in MPU-type mode.
	LCM_DATA9/LCM_MPU_D9	I/O	TFT LCD Module Pixel Data output bit 9 in Sync-type mode. TFT LCD Module Command/Data input/output bit 9 in MPU-type mode.
	LCM_DEN/LCM_MPU_RS	O	TFT LCD Module Data Enable/Display Control Signal output pin in Sync-type mode. TFT LCD Module Register Select (RS) output pin in MPU-type mode.
	LCM_HSYNC/LCM_MPU_WR/RW	O	TFT LCD Module Horizontal/Line sync. output in Sync-type mode. TFT LCD Module Write(WR)/ReadWrite(RW) output pin in MPU-type mode.
	LCM_MPU_TE	I	TFT LCD Module TE input pin in MPU-type mode.
	LCM_MPU_VSYNC	O	TFT LCD Module VSYNC output pin in MPU-type mode.
	LCM_VSYNC/LCM_MPU_RD/EN	O	TFT LCD Module Vertical/Frame sync. output pin in Sync-type mode. TFT LCD Module Read(RD)/Enable(EN) output pin in MPU-type mode.
NAND	NAND_ALE	O	NAND Flash address latch enable output pin.

Group	Pin Name	Type	Description
	NAND_CLE	O	NAND Flash command latch enable output pin.
	NAND_DATA0	I/O	NAND Flash date bus bit 0.
	NAND_DATA1	I/O	NAND Flash date bus bit 1.
	NAND_DATA2	I/O	NAND Flash date bus bit 2.
	NAND_DATA3	I/O	NAND Flash date bus bit 3.
	NAND_DATA4	I/O	NAND Flash date bus bit 4.
	NAND_DATA5	I/O	NAND Flash date bus bit 5.
	NAND_DATA6	I/O	NAND Flash date bus bit 6.
	NAND_DATA7	I/O	NAND Flash date bus bit 7.
	NAND_RDY0	I	NAND Flash ready/busy 0 input pin.
	NAND_RDY1	I	NAND Flash ready/busy 1 input pin.
	NAND_nCS0	O	NAND Flash chip select 0 pin.
	NAND_nCS1	O	NAND Flash chip select 1 pin.
	NAND_nRE	O	NAND Flash read enable output pin.
	NAND_nWE	O	NAND Flash write enable output pin.
	NAND_nWP	I	NAND Flash write protect input pin.
QEI0	QEI0_A	I	Quadrature encoder 0 phase A input.
	QEI0_B	I	Quadrature encoder 0 phase B input.
	QEI0_INDEX	I	Quadrature encoder 0 index input.
QEI1	QEI1_A	I	Quadrature encoder 1 phase A input.
	QEI1_B	I	Quadrature encoder 1 phase B input.
	QEI1_INDEX	I	Quadrature encoder 1 index input.
QEI2	QEI2_A	I	Quadrature encoder 2 phase A input.
	QEI2_B	I	Quadrature encoder 2 phase B input.
	QEI2_INDEX	I	Quadrature encoder 2 index input.
QSPI0	QSPI0_CLK	I/O	Quad SPI0 serial clock pin.
	QSPI0_MISO0	I/O	Quad SPI0 MISO0 (Master In, Slave Out) pin.
	QSPI0_MISO1	I/O	Quad SPI0 MISO1 (Master In, Slave Out) pin.
	QSPI0_MOSI0	I/O	Quad SPI0 MOSI0 (Master Out, Slave In) pin.
	QSPI0_MOSI1	I/O	Quad SPI0 MOSI1 (Master Out, Slave In) pin.
	QSPI0_SS0	I/O	Quad SPI0 slave select 0 pin.
QSPI1	QSPI1_CLK	I/O	Quad SPI1 serial clock pin.
	QSPI1_MISO0	I/O	Quad SPI1 MISO0 (Master In, Slave Out) pin.
	QSPI1_MISO1	I/O	Quad SPI1 MISO1 (Master In, Slave Out) pin.

Group	Pin Name	Type	Description
	QSPI1_MOSI0	I/O	Quad SPI1 MOSI0 (Master Out, Slave In) pin.
	QSPI1_MOSI1	I/O	Quad SPI1 MOSI1 (Master Out, Slave In) pin.
	QSPI1_SS0	I/O	Quad SPI1 slave select 0 pin.
	QSPI1_SS1	I/O	Quad SPI1 slave select 1 pin.
RGMII0	RGMII0_MDC	O	RGMII0 PHY Management Clock output pin.
	RGMII0_MDIO	I/O	RGMII0 PHY Management Data pin.
	RGMII0_PPS	O	RGMII0 Pulse Per Second output pin.
	RGMII0_RXCLK	I	RGMII0 Mode RX Clock input pin.
	RGMII0_RXCTL	I	RGMII0 Receive Control input pin.
	RGMII0_RXD0	I	RGMII0 Receive Data bus bit 0.
	RGMII0_RXD1	I	RGMII0 Receive Data bus bit 1.
	RGMII0_RXD2	I	RGMII0 Receive Data bus bit 2.
	RGMII0_RXD3	I	RGMII0 Receive Data bus bit 3.
	RGMII0_TXCLK	O	RGMII0 Mode TX Clock output pin.
	RGMII0_TXCTL	O	RGMII0 Transmit Control output pin.
	RGMII0_TXD0	O	RGMII0 Transmit Data bus bit 0.
	RGMII0_TXD1	O	RGMII0 Transmit Data bus bit 1.
	RGMII0_TXD2	O	RGMII0 Transmit Data bus bit 2.
	RGMII0_TXD3	O	RGMII0 Transmit Data bus bit 3.
RGMII1	RGMII1_MDC	O	RGMII1 PHY Management Clock output pin.
	RGMII1_MDIO	I/O	RGMII1 PHY Management Data pin.
	RGMII1_PPS	O	RGMII1 Pulse Per Second output pin.
	RGMII1_RXCLK	I	RGMII1 Mode RX Clock input pin.
	RGMII1_RXCTL	I	RGMII1 Receive Control input pin.
	RGMII1_RXD0	I	RGMII1 Receive Data bus bit 0.
	RGMII1_RXD1	I	RGMII1 Receive Data bus bit 1.
	RGMII1_RXD2	I	RGMII1 Receive Data bus bit 2.
	RGMII1_RXD3	I	RGMII1 Receive Data bus bit 3.
	RGMII1_TXCLK	O	RGMII1 Mode TX Clock output pin.
	RGMII1_TXCTL	O	RGMII1 Transmit Control output pin.
	RGMII1_TXD0	O	RGMII1 Transmit Data bus bit 0.
	RGMII1_TXD1	O	RGMII1 Transmit Data bus bit 1.
	RGMII1_TXD2	O	RGMII1 Transmit Data bus bit 2.
	RGMII1_TXD3	O	RGMII1 Transmit Data bus bit 3.

Group	Pin Name	Type	Description
RMII0	RMII0_CRSDV	I	RMII0 Carrier Sense/Receive Data input pin.
	RMII0_MDC	O	RMII0 PHY Management Clock output pin.
	RMII0_MDIO	I/O	RMII0 PHY Management Data pin.
	RMII0_PPS	O	RMII0 Pulse Per Second output pin.
	RMII0_REFCLK	I	RMII0 Reference Clock input pin.
	RMII0_RXD0	I	RMII0 Receive Data bus bit 0.
	RMII0_RXD1	I	RMII0 Receive Data bus bit 1.
	RMII0_RXERR	I	RMII0 Receive Data Error input pin.
	RMII0_TXD0	O	RMII0 Transmit Data bus bit 0.
	RMII0_TXD1	O	RMII0 Transmit Data bus bit 1.
	RMII0_TXEN	O	RMII0 Transmit Enable output pin.
RMII1	RMII1_CRSDV	I	RMII1 Carrier Sense/Receive Data input pin.
	RMII1_MDC	O	RMII1 PHY Management Clock output pin.
	RMII1_MDIO	I/O	RMII1 PHY Management Data pin.
	RMII1_PPS	O	RMII1 Pulse Per Second output pin.
	RMII1_REFCLK	I	RMII1 Reference Clock input pin.
	RMII1_RXD0	I	RMII1 Receive Data bus bit 0.
	RMII1_RXD1	I	RMII1 Receive Data bus bit 1.
	RMII1_RXERR	I	RMII1 Receive Data Error input pin.
	RMII1_TXD0	O	RMII1 Transmit Data bus bit 0.
	RMII1_TXD1	O	RMII1 Transmit Data bus bit 1.
	RMII1_TXEN	O	RMII1 Transmit Enable output pin.
SC0	SC0_CLK	O	Smart Card 0 clock pin.
	SC0_DAT	I/O	Smart Card 0 data pin.
	SC0_PWR	O	Smart Card 0 power pin.
	SC0_RST	O	Smart Card 0 reset pin.
	SC0_nCD	I	Smart Card 0 card detect pin.
SC1	SC1_CLK	O	Smart Card 1 clock pin.
	SC1_DAT	I/O	Smart Card 1 data pin.
	SC1_PWR	O	Smart Card 1 power pin.
	SC1_RST	O	Smart Card 1 reset pin.
	SC1_nCD	I	Smart Card 1 card detect pin.
SD0	SD0_CLK/eMMC0_CLK	O	SD/SDIO0 clock output pin. eMMC0 clock output pin.
	SD0_CMD/eMMC0_C	I/O	SD/SDIO0 command/response pin.

Group	Pin Name	Type	Description
	MD		eMMC0 command/response pin.
	SD0_DAT0/eMMC0_DAT0	I/O	SD/SDIO0 data line bit 0. eMMC0 data line bit 0.
	SD0_DAT1/eMMC0_DAT1	I/O	SD/SDIO0 data line bit 1. eMMC0 data line bit 1.
	SD0_DAT2/eMMC0_DAT2	I/O	SD/SDIO0 data line bit 2. eMMC0 data line bit 2.
	SD0_DAT3/eMMC0_DAT3	I/O	SD/SDIO0 data line bit 3. eMMC0 data line bit 3.
	SD0_WP	I	SD/SDIO0 write protect input.
	SD0_nCD	I	SD/SDIO0 card detect input pin. **Note:** An external 10kΩ pull-down resistor is necessary on this pin when these eMMC0_xxx pins are used to connect with eMMC device and act as the booting source.
SD1	SD1_CLK/eMMC1_CLK	O	SD/SDIO1 clock output pin. eMMC1 clock output pin.
	SD1_CMD/eMMC1_CMD	I/O	SD/SDIO1 command/response pin. eMMC1 command/response pin.
	SD1_DAT0/eMMC1_DAT0	I/O	SD/SDIO1 data line bit 0. eMMC1 data line bit 0.
	SD1_DAT1/eMMC1_DAT1	I/O	SD/SDIO1 data line bit 1. eMMC1 data line bit 1.
	SD1_DAT2/eMMC1_DAT2	I/O	SD/SDIO1 data line bit 2. eMMC1 data line bit 2.
	SD1_DAT3/eMMC1_DAT3	I/O	SD/SDIO1 data line bit 3. eMMC1 data line bit 3.
	SD1_WP	I	SD/SDIO1 write protect input.
	SD1_nCD	I	SD/SDIO1 card detect input pin. **Note:** An external 10kΩ pull-down resistor is necessary on this pin when these eMMC1_xxx pins are used to connect with eMMC device and act as the booting source.
SPI0	SPI0_CLK	I/O	SPI0 serial clock pin.
	SPI0_I2SMCLK	I/O	SPI0 I^2S master clock output pin.
	SPI0_MISO	I/O	SPI0 MISO (Master In, Slave Out) pin.
	SPI0_MOSI	I/O	SPI0 MOSI (Master Out, Slave In) pin.
	SPI0_SS0	I/O	SPI0 slave select 0 pin.
	SPI0_SS1	I/O	SPI0 slave select 1 pin.
SPI1	SPI1_CLK	I/O	SPI1 serial clock pin.
	SPI1_I2SMCLK	I/O	SPI1 I^2S master clock output pin.
	SPI1_MISO	I/O	SPI1 MISO (Master In, Slave Out) pin.

Group	Pin Name	Type	Description
	SPI1_MOSI	I/O	SPI1 MOSI (Master Out, Slave In) pin.
	SPI1_SS0	I/O	SPI1 slave select 0 pin.
	SPI1_SS1	I/O	SPI1 slave select 1 pin.
SPI2	SPI2_CLK	I/O	SPI2 serial clock pin.
	SPI2_I2SMCLK	I/O	SPI2 I^2S master clock output pin.
	SPI2_MISO	I/O	SPI2 MISO (Master In, Slave Out) pin.
	SPI2_MOSI	I/O	SPI2 MOSI (Master Out, Slave In) pin.
	SPI2_SS0	I/O	SPI2 slave select 0 pin.
	SPI2_SS1	I/O	SPI2 slave select 1 pin.
SPI3	SPI3_CLK	I/O	SPI3 serial clock pin.
	SPI3_I2SMCLK	I/O	SPI3 I^2S master clock output pin.
	SPI3_MISO	I/O	SPI3 MISO (Master In, Slave Out) pin.
	SPI3_MOSI	I/O	SPI3 MOSI (Master Out, Slave In) pin.
	SPI3_SS0	I/O	SPI3 slave select 0 pin.
	SPI3_SS1	I/O	SPI3 slave select 1 pin.
TAMPER0	TAMPER0	I/O	TAMPER detector loop pin 0.
TAMPER1	TAMPER1	I/O	TAMPER detector loop pin 1.
TM0	TM0	I/O	Timer0 event counter input/toggle output pin.
	TM0_EXT	I/O	Timer0 external capture input/toggle output pin.
TM1	TM1	I/O	Timer1 event counter input/toggle output pin.
	TM1_EXT	I/O	Timer1 external capture input/toggle output pin.
TM2	TM2	I/O	Timer2 event counter input/toggle output pin.
	TM2_EXT	I/O	Timer2 external capture input/toggle output pin.
TM3	TM3	I/O	Timer3 event counter input/toggle output pin.
	TM3_EXT	I/O	Timer3 external capture input/toggle output pin.
TM4	TM4	I/O	Timer4 event counter input/toggle output pin.
	TM4_EXT	I/O	Timer4 external capture input/toggle output pin.
TM5	TM5	I/O	Timer5 event counter input/toggle output pin.
	TM5_EXT	I/O	Timer5 external capture input/toggle output pin.
TM6	TM6	I/O	Timer6 event counter input/toggle output pin.
	TM6_EXT	I/O	Timer6 external capture input/toggle output pin.
TM7	TM7	I/O	Timer7 event counter input/toggle output pin.
	TM7_EXT	I/O	Timer7 external capture input/toggle output pin.
TM8	TM8	I/O	Timer8 event counter input/toggle output pin.

Group	Pin Name	Type	Description
	TM8_EXT	I/O	Timer8 external capture input/toggle output pin.
TM9	TM9	I/O	Timer9 event counter input/toggle output pin.
	TM9_EXT	I/O	Timer9 external capture input/toggle output pin.
TM10	TM10	I/O	Timer10 event counter input/toggle output pin.
	TM10_EXT	I/O	Timer10 external capture input/toggle output pin.
TM11	TM11	I/O	Timer11 event counter input/toggle output pin.
	TM11_EXT	I/O	Timer11 external capture input/toggle output pin.
TRACE	TRACE_CLK	O	ETM Trace Clock output pin.
	TRACE_DATA0	O	ETM Trace Data 0 output pin.
	TRACE_DATA1	O	ETM Trace Data 1 output pin.
	TRACE_DATA2	O	ETM Trace Data 2 output pin.
	TRACE_DATA3	O	ETM Trace Data 3 output pin.
UART0	UART0_RXD	I	UART0 data receiver input pin. **Note:** This UART0_RXD (GPIO PE.15) pin must be pulled to high level through an external 10kΩ resistor or an internal pull-up resistor in the external device (such as a transceiver or MCU).
	UART0_TXD	O	UART0 data transmitter output pin.
UART1	UART1_RXD	I	UART1 data receiver input pin.
	UART1_TXD	O	UART1 data transmitter output pin.
	UART1_nCTS	I	UART1 clear to Send input pin.
	UART1_nRTS	O	UART1 request to Send output pin.
UART2	UART2_RXD	I	UART2 data receiver input pin.
	UART2_TXD	O	UART2 data transmitter output pin.
	UART2_nCTS	I	UART2 clear to Send input pin.
	UART2_nRTS	O	UART2 request to Send output pin.
UART3	UART3_RXD	I	UART3 data receiver input pin.
	UART3_TXD	O	UART3 data transmitter output pin.
	UART3_nCTS	I	UART3 clear to Send input pin.
	UART3_nRTS	O	UART3 request to Send output pin.
UART4	UART4_RXD	I	UART4 data receiver input pin.
	UART4_TXD	O	UART4 data transmitter output pin.
	UART4_nCTS	I	UART4 clear to Send input pin.
	UART4_nRTS	O	UART4 request to Send output pin.
UART5	UART5_RXD	I	UART5 data receiver input pin.
	UART5_TXD	O	UART5 data transmitter output pin.
	UART5_nCTS	I	UART5 clear to Send input pin.

Group	Pin Name	Type	Description
	UART5_nRTS	O	UART5 request to Send output pin.
UART6	UART6_RXD	I	UART6 data receiver input pin.
	UART6_TXD	O	UART6 data transmitter output pin.
	UART6_nCTS	I	UART6 clear to Send input pin.
	UART6_nRTS	O	UART6 request to Send output pin.
UART7	UART7_RXD	I	UART7 data receiver input pin.
	UART7_TXD	O	UART7 data transmitter output pin.
	UART7_nCTS	I	UART7 clear to Send input pin.
	UART7_nRTS	O	UART7 request to Send output pin.
UART8	UART8_RXD	I	UART8 data receiver input pin.
	UART8_TXD	O	UART8 data transmitter output pin.
	UART8_nCTS	I	UART8 clear to Send input pin.
	UART8_nRTS	O	UART8 request to Send output pin.
UART9	UART9_RXD	I	UART9 data receiver input pin.
	UART9_TXD	O	UART9 data transmitter output pin.
	UART9_nCTS	I	UART9 clear to Send input pin.
	UART9_nRTS	O	UART9 request to Send output pin.
UART10	UART10_RXD	I	UART10 data receiver input pin.
	UART10_TXD	O	UART10 data transmitter output pin.
	UART10_nCTS	I	UART10 clear to Send input pin.
	UART10_nRTS	O	UART10 request to Send output pin.
UART11	UART11_RXD	I	UART11 data receiver input pin.
	UART11_TXD	O	UART11 data transmitter output pin.
	UART11_nCTS	I	UART11 clear to Send input pin.
	UART11_nRTS	O	UART11 request to Send output pin.
UART12	UART12_RXD	I	UART12 data receiver input pin.
	UART12_TXD	O	UART12 data transmitter output pin.
	UART12_nCTS	I	UART12 clear to Send input pin.
	UART12_nRTS	O	UART12 request to Send output pin.
UART13	UART13_RXD	I	UART13 data receiver input pin.
	UART13_TXD	O	UART13 data transmitter output pin.
	UART13_nCTS	I	UART13 clear to Send input pin.
	UART13_nRTS	O	UART13 request to Send output pin.
UART14	UART14_RXD	I	UART14 data receiver input pin.

Group	Pin Name	Type	Description
	UART14_TXD	O	UART14 data transmitter output pin.
	UART14_nCTS	I	UART14 clear to Send input pin.
	UART14_nRTS	O	UART14 request to Send output pin.
UART15	UART15_RXD	I	UART15 data receiver input pin.
	UART15_TXD	O	UART15 data transmitter output pin.
	UART15_nCTS	I	UART15 clear to Send input pin.
	UART15_nRTS	O	UART15 request to Send output pin.
UART16	UART16_RXD	I	UART16 data receiver input pin.
	UART16_TXD	O	UART16 data transmitter output pin.
	UART16_nCTS	I	UART16 clear to Send input pin.
	UART16_nRTS	O	UART16 request to Send output pin.
WDT	WDT_nRST	O	Watchdog Timer reset trigger output pin.
eMMC1	eMMC1_DAT4	I/O	eMMC1 data line bit 4.
	eMMC1_DAT5	I/O	eMMC1 data line bit 5.
	eMMC1_DAT6	I/O	eMMC1 data line bit 6.
	eMMC1_DAT7	I/O	eMMC1 data line bit 7.
USB	HSUSB0_D+	A	HSUSB0 differential signal D+.
	HSUSB0_D-	A	HSUSB0 differential signal D-.
	HSUSB0_ID	I	HSUSB0 identification pin.
	HSUSB1_D+	A	HSUSB1 differential signal D+.
	HSUSB1_D-	A	HSUSB1 differential signal D-.
	V_{DD_HSUSB0}	P	3.3V power supply for high speed USB 2.0 port 0 PHY.
	V_{DD_HSUSB1}	P	3.3V power supply for high speed USB 2.0 port 1 PHY.
Analog	AV_{DD}	P	3.3V analog power supply for internal POR33, LVD, LVR and temperature sensor. **Note:** This pin is only for BGA 312-Ball and BGA 364-Ball package types.
	AV_{DDH_PLL1}	P	3.3V analog power supply for PLL group 1.
	AV_{DDH_PLL2}	P	3.3V analog power supply for PLL group 2.
	AV_{DDL_PLL0}	P	Low analog power supply for PLL group 0.
	AV_{DDL_PLL1}	P	Low analog power supply for PLL group 1.
	AV_{DDL_PLL2}	P	Low analog power supply for PLL group 2.
	AV_{DDL_ROSC}	P	Low analog power supply for internal 12 MHz High Speed RC Oscillator (HIRC).
	AV_{DD_ADC0}	P	3.3V analog power supply for ADC0. **Note:** PB.8~PB.15 are belong to this power domain.
	AV_{DD_EADC0}	P	3.3V analog power supply for EADC0.

Group	Pin Name	Type	Description
			Note: EADC0_CH0~EADC0_CH7 are belong to this power domain.
	AV_{SS}	P	Ground pin for analog circuit.
	AV_{SS_PLL}	P	Ground pin for PLL.
	V_{REF_EADC0}	I	Analog EADC0 reference voltage input. **Note:** This pin needs to be connected with 0.1uF and 2.2uF capacitors.
DDR	MA0	O	External DDR SDRAM address bus bit 0.
	MA1	O	External DDR SDRAM address bus bit 1.
	MA10	O	External DDR SDRAM address bus bit 10.
	MA11	O	External DDR SDRAM address bus bit 11.
	MA12	O	External DDR SDRAM address bus bit 12.
	MA13	O	External DDR SDRAM address bus bit 13.
	MA14	O	External DDR SDRAM address bus bit 14.
	MA15	O	External DDR SDRAM address bus bit 15.
	MA2	O	External DDR SDRAM address bus bit 2.
	MA3	O	External DDR SDRAM address bus bit 3.
	MA4	O	External DDR SDRAM address bus bit 4.
	MA5	O	External DDR SDRAM address bus bit 5.
	MA6	O	External DDR SDRAM address bus bit 6.
	MA7	O	External DDR SDRAM address bus bit 7.
	MA8	O	External DDR SDRAM address bus bit 8.
	MA9	O	External DDR SDRAM address bus bit 9.
	MBA0	O	External DDR SDRAM bank address 0 output.
	MBA1	O	External DDR SDRAM bank address 1 output.
	MBA2	O	External DDR SDRAM bank address 2 output.
	MCASn	O	External DDR SDRAM column command output.
	MCKE0	O	External DDR SDRAM clock enable output 0.
	MCKE1	O	External DDR SDRAM clock enable output 1.
	MCK_N	O	External DDR SDRAM negative clock output.
	MCK_P	O	External DDR SDRAM positive clock output.
	MCS0n	O	External DDR SDRAM chip select output 0.
	MCS1n	O	External DDR SDRAM chip select output 1.
	MD0	I/O	External DDR SDRAM data bus bit 0.
	MD1	I/O	External DDR SDRAM data bus bit 1.
	MD10	I/O	External DDR SDRAM data bus bit 10.
	MD11	I/O	External DDR SDRAM data bus bit 11.

Group	Pin Name	Type	Description
	MD12	I/O	External DDR SDRAM data bus bit 12.
	MD13	I/O	External DDR SDRAM data bus bit 13.
	MD14	I/O	External DDR SDRAM data bus bit 14.
	MD15	I/O	External DDR SDRAM data bus bit 15.
	MD2	I/O	External DDR SDRAM data bus bit 2.
	MD3	I/O	External DDR SDRAM data bus bit 3.
	MD4	I/O	External DDR SDRAM data bus bit 4.
	MD5	I/O	External DDR SDRAM data bus bit 5.
	MD6	I/O	External DDR SDRAM data bus bit 6.
	MD7	I/O	External DDR SDRAM data bus bit 7.
	MD8	I/O	External DDR SDRAM data bus bit 8.
	MD9	I/O	External DDR SDRAM data bus bit 9.
	MDMask0	O	External DDR SDRAM write data mask 0 output.
	MDMask1	O	External DDR SDRAM write data mask 1 output.
	MDQS0_N	I/O	External DDR SDRAM data strobe 0 negative.
	MDQS0_P	I/O	External DDR SDRAM data strobe 0 positive.
	MDQS1_N	I/O	External DDR SDRAM data strobe 1 negative.
	MDQS1_P	I/O	External DDR SDRAM data strobe 1 positive.
	MODT0	O	External DDR SDRAM on-die termination control 0.
	MODT1	O	External DDR SDRAM on-die termination control 1.
	MRASn	O	External DDR SDRAM row command output.
	MRESETn	O	External DDR SDRAM reset signal output.
	MV_{DD}	P	Power supply for internal DDR2/DDR3L-type SDRAM or external DDR3/DDR3L-type SDRAM. **Note 1:** Please refer to the part number from MA35D1 selection guide to supply the correct voltage on this pin. **Note 2:** 1.8V for internal DDR2-type SDRAM or 1.35V for internal DDR3L-type SDRAM. **Note 3:** 1.5V for external DDR3-type SDRAM or 1.35V for external DDR3L-type SDRAM.
	$MV_{DD_DPHYPLL}$	P	2.5V power supply for PLL of DDR PHY.
	MV_{REF}	P	DDR SDRAM reference voltage. (MV_{REF} = 1/2MV_{DD})
	MV_{REF_CA}	P	DDR SDRAM reference voltage for control, command and address. (MV_{REF_CA} = 1/2MV_{DD})
	MV_{REF_DQ}	P	DDR SDRAM reference voltage for DQ data. (MV_{REF_DQ} = 1/2MV_{DD})
	MWEn	O	External DDR SDRAM write command output.
	MZQ_DDRPHY	P	An external 240Ω (1%) pull-down resistor on this pin for DDR PHY.
	MZQ_SDRAM	P	An external 240Ω (1%) pull-down resistor on this pin for internal SDRAM.

Group	Pin Name	Type	Description
RTC	RTC_RPWR	O	RTC wake-up output pin for external Power IC enable pin control.
	RTC_nRWAKE	I	RTC wake-up interrupt Input with internal pull-high. **Note:** Please pull this pin to low if the RTC wake-up interrupt function is unused. In this condition, user also needs to disable the PWRST bit (RTC_PWRCTL[6]) of the RTC Power Control Register by clearing it for saving the RTC power consumption.
	V_{BAT}	P	3.3V power supply by batteries for RTC. **Note:** PH.8~PH.9, RTC_RPWR, RTC_nRWAKE, X32_IN and X32_OUT are belong to this power domain.
	X32_IN	I	External 32.768 kHz crystal input pin.
	X32_OUT	O	External 32.768 kHz crystal output pin.
V_{DDIO}	V_{DDIO0}	P	3.3V digital power supply for I/O group 0. **Note:** nRESET, HSUSB0_ID, PF.15 and PL.12~PL.15 are belong to this power domain. **Note:** The AV_{DD} analog power pad is bonded with this V_{DDIO0} pin in LQFP-EP 216-Pin package type.
	V_{DDIO1}	P	3.3V digital power supply for I/O group 1. **Note:** PA.0~PA.15, PD.6~PD.15, PE.14~PE.15, PG.0~PG.7, PG11~PG15, PI0~PI.7, PJ.12~PJ.15, PK0~PK.3, PL.0~PL.11 and PM.12~PM.15 are belong to this power domain.
	V_{DDIO2}	P	1.8V ~ 3.3V digital power supply for I/O group 2. **Note:** PK.8~PK.11 and PM.0~PM.11 are belong to this power domain.
	V_{DDIO3}	P	1.8V ~ 3.3V digital power supply for I/O group 3. **Note:** PJ.0~PJ.11 are belong to this power domain.
	V_{DDIO4}	P	1.8V ~ 3.3V digital power supply for I/O group 4. **Note:** PC.12~PC.15, PG.8~PG.10, PH.0~PH.7, PH.12~PH.15, PI.8~PI.15 and PK.4~PK.7 are belong to this power domain.
	V_{DDIO5}	P	1.8V ~ 3.3V digital power supply for I/O group 5. **Note:** PD.0~PD.5 are belong to this power domain.
	V_{DDIO6}	P	3.3V digital power supply for I/O group 6. **Note:** PC.0~PC.7 are belong to this power domain.
	V_{DDIO7}	P	1.8V ~ 3.3V digital power supply for I/O group 7. **Note:** PK.12~PK.15, PN.0~PN.7 and PN.10~PN.15 are belong to this power domain.
	V_{DDIO8}	P	1.8V ~ 3.3V digital power supply for I/O group 8. **Note:** PE.0~PE.13 are belong to this power domain.
	V_{DDIO9}	P	1.8V ~ 3.3V digital power supply for I/O group 9. **Note:** PF.0~PF.14 are belong to this power domain.
	V_{DDIO10}	P	3.3V digital power supply for I/O group 10. **Note:** XT1_IN and XT1_OUT are belong to this power domain.
Core	V_{DD_CORE}	P	Digital power supply for digital core logic circuit. **Note:** Please refer to the part number from MA35D1 selection guide to supply the correct voltage on this pin. For the actual input voltage range, please refer to the Minimum/Typical/Maximum values in section 8.2.1 of Datasheet document.

Group	Pin Name	Type	Description
CPU	V_{DD_CPU}	P	Digital power supply for Cortex-A35 CPU. **Note:** Please refer to the part number from MA35D1 selection guide to supply the correct voltage on this pin. For the actual input voltage range, please refer to the Minimum/Typical/Maximum values in section 8.2.1 of Datasheet document. **Note:** There are not any V_{DD_CPU} pins in the LQFP-EP 216-Pin package (MA35D16F787C/MA35D16F8(I)87C/MA35D16F9(J)87C/MA35D14F884/ MA35D14F984) that is because of these V_{DD_CPU} and V_{DD_CORE} power pads of MA35D1 die are bonded together inside the package.
OTP	V_{DD_OTP}	P	2.5V digital power supply for OTP.
PLL	V_{DD_PLL1}	P	Low digital power supply for PLL group 1.
Ground	V_{SS}	P	Ground pin for digital circuit.
	EPAD	P	Ground pad for digital circuit. **Note:** When soldering SMD, please make sure the EPAD on the bottom of LQFP-EP 216-Pin package is tightly soldered to the ground of the PCB.
Crystal	XT1_IN	I	External 24 MHz (high speed) crystal input pin.
	XT1_OUT	O	External 24 MHz (high speed) crystal output pin.
Reset	nRESET	I	External reset input: active LOW, with an internal pull-up. Set this pin low reset to initial state.

5 BLOCK DIAGRAM

5.1 MA35D1 Series Block Diagram

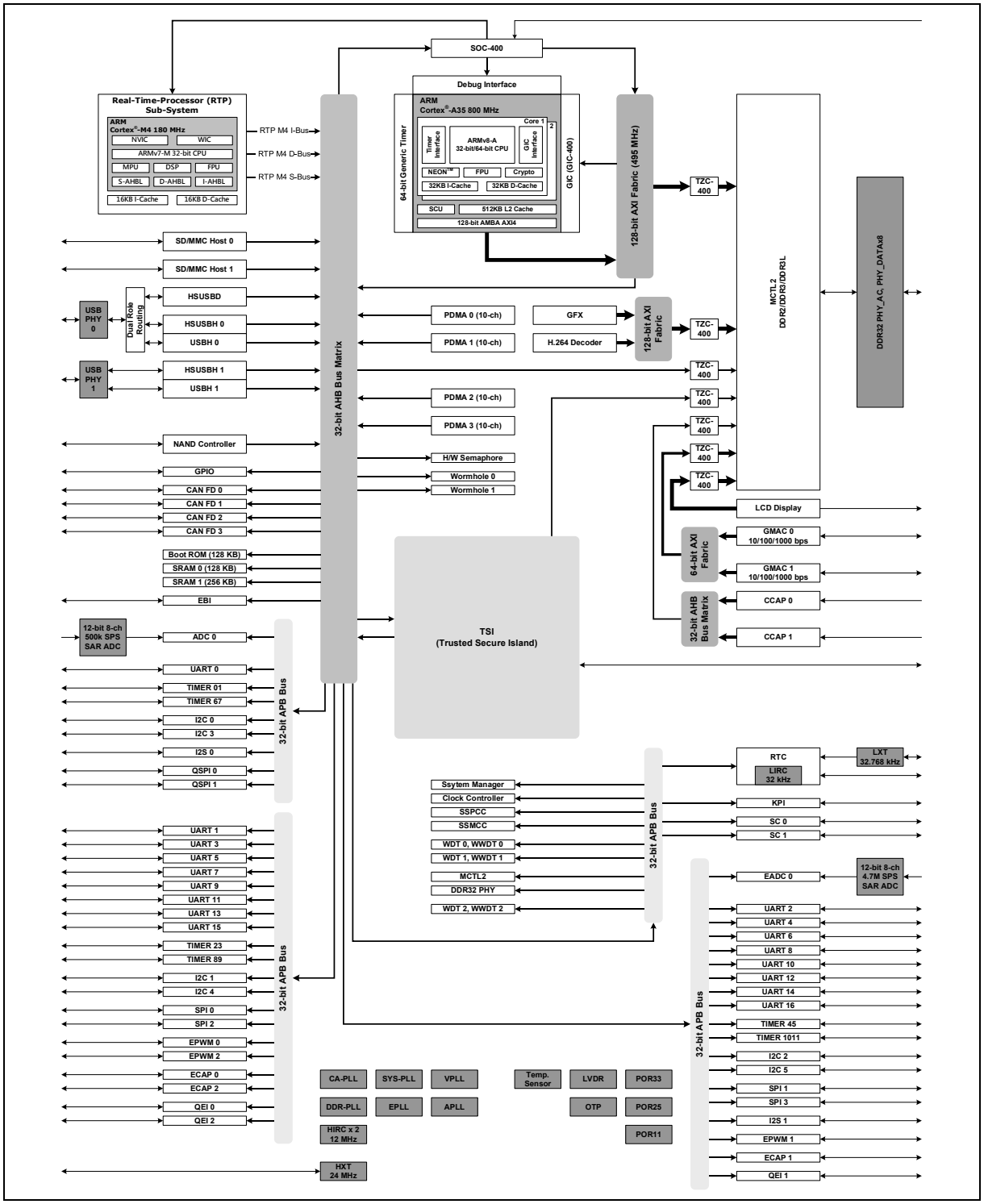


Figure 5.1-1 MA35D1 Series Block Diagram

6 FUNCTIONAL DESCRIPTION

6.1 Arm Cortex-A35 Core

6.1.1 Overview

The Cortex-A35 core is designed to give mid-range instruction execution performance with low power consumption.

In MA35D1 series, the Cortex-A35 core cluster consists of two ARMv8-A compliant cores with 32 KB instruction and 32 KB data L1 cache each, a shared 512 KB L2 cache with Snoop Control Unit (SCU)-L2 cache protection and a 128-bit AXI system bus interface.

For more detailed information, please refer to the "*ARM® Cortex-A35 Processor Technical Reference Manual*" and "*ARM® Architecture Reference Manual ARMv8, for ARMv8-A architecture profile*".

6.1.2 Features

In MA35D1 series, the Cortex-A35 core includes the following features:

- Full implementation of the ARMv8-A A64, A32, and T32 instruction sets.
- Support both the AArch32 and AArch64 execution states at all Exception levels (EL0 to EL3).
- Support in-order pipeline with direct and indirect branch prediction.
- Separate Level 1 (L1) data and instruction side memory systems with a Memory Management Unit (MMU).
  - The MA35D1 series configured with 32 KB instruction and 32 KB data cache.
- Level 2 (L2) memory system with Snoop Control Unit (SCU)-L2 cache protection that provides cluster memory coherency.
  - The MA35D1 series configured with 512 KB shared L2 cache.
- TrustZone.
- Data engine that implements the NEON SIMD and FPU support.
- Cryptographic Extension.
- ARMv8 debug logic.
- Performance Monitoring Unit (PMU).
- Embedded Trace Macrocell (ETM) that supports instruction trace only.
- Generic Interrupt Controller (GIC) CPU interface to connect to an external distributor.
- Generic Timers supporting 64-bit count input from an external system counter.

6.1.3 Block Diagram

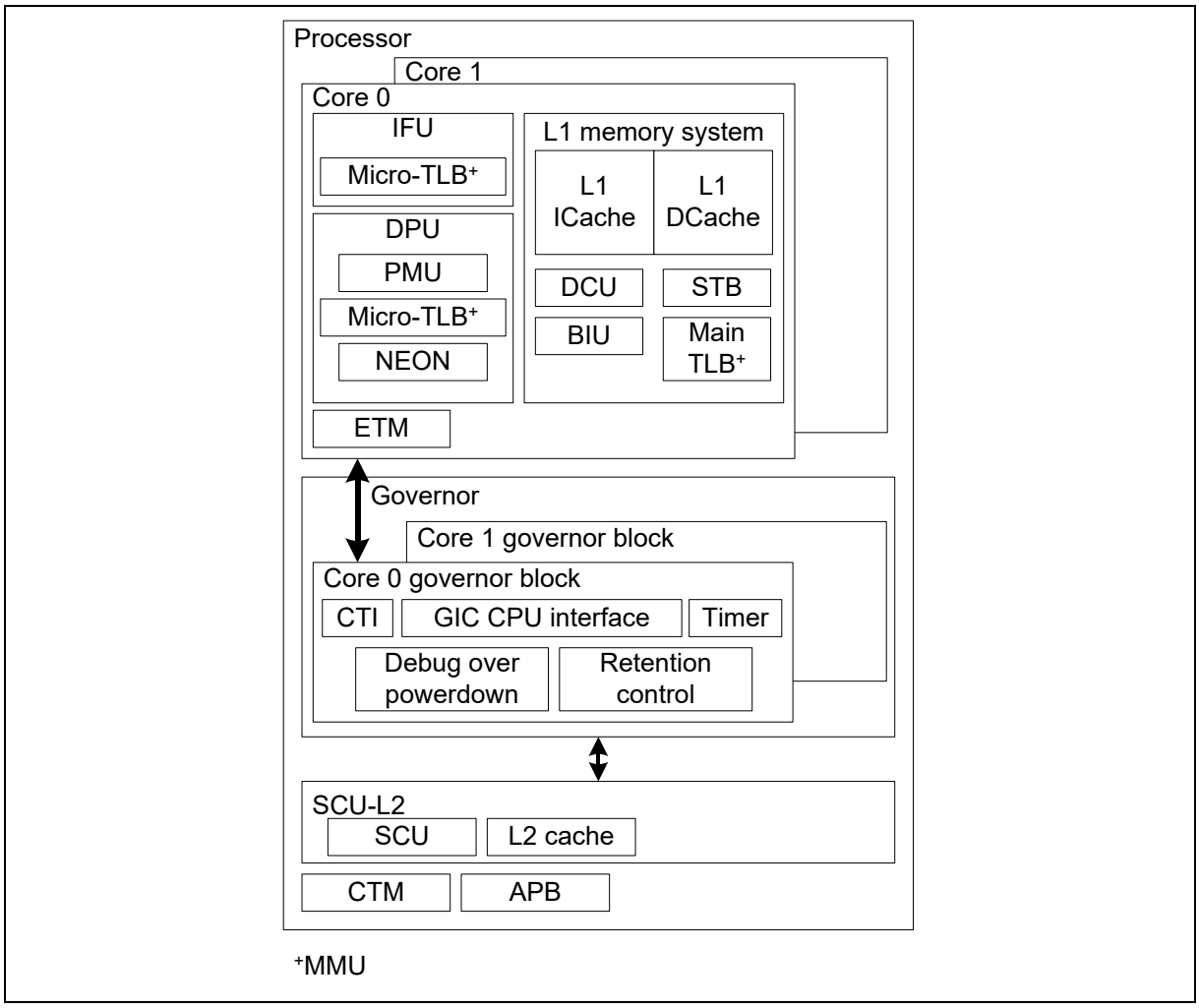


Figure 6.1-1 Arm Cortex-A35 Processor Block Diagram

6.1.4 Functional Description

6.1.4.1 Instruction Fetch Unit (IFU)

The IFU obtains instructions from the instruction cache or from external memory and predicts the outcome of branches in the instruction stream. It passes the instructions to the Data Processing Unit (DPU) for processing.

In implementations with CPU cache protection, parity bits protect the L1 Instruction cache data and tag RAMs by enabling the detection of any single-bit error. If an error is detected, the line is invalidated and fetched again.

6.1.4.2 Data Processing Unit (DPU)

The DPU decodes and executes instructions. It executes instructions that require data transfer to or from the memory system by interfacing to the Data Cache Unit (DCU). The DPU includes the Performance Monitor Unit (PMU), the Advanced SIMD and floating-point support, and the Cryptographic Extension.

6.1.4.3 Memory Management Unit (MMU)

The MMU provides fine-grained memory system control through a set of virtual-to-physical address mappings and memory attributes that are held in translation tables. These are loaded into the Translation Lookaside Buffer (TLB) when a location is accessed. The TLB entries include global and application specific identifiers to prevent context switch TLB flushes. They also include Virtual Machine Identifiers (VMIDs) to prevent TLB flushes on virtual machine switches by the hypervisor.

6.1.4.4 L1 data-side memory system

The L1 data-side memory system includes the Data Cache Unit (DCU), the Store Buffer (STB), and the Bus Interface Unit.

6.1.4.5 L2 Memory System

The L2 memory system contains the L2 cache pipeline and all the logic that maintains memory coherence between the cores in the cluster.

6.1.5 Generic Timer

The Generic Timer has an interface to a 64-bit incrementing system counter. It provides a consistent view of time, which can be used to schedule events and trigger interrupts.

The Generic Timer

- Provide a system counter, that measures the passing of time in real-time
- Support virtual counters that measure the passing of virtual-time. That is, a virtual counter can measure the passing of time on a particular virtual machine.
- Support to trigger events after a period of time has passed.
  - Used as count-up or as count-down timers.
  - operate in real-time or in virtual-time.

A set of Generic Timer registers allocated within each Cortex-A35 core. The Generic Timer registers are either 32-bits wide or 64-bits wide and accessible in the AArch32 and AArch64 Execution states.

For more detailed information, please refer to the "*ARM® Cortex-A35 Processor Technical Reference Manual*" and "*ARM® Architecture Reference Manual ARMv8, for ARMv8-A architecture profile*".

6.1.6 Generic Interrupt Controller (GIC)

The Generic Interrupt Controller (GIC) is a centralized resource for supporting and managing interrupts on this chip. The GIC implements

- GIC Security Extensions, which supports
  - 2 groups, group 0 and group 1, for secure interrupts and Non-secure interrupts.
  - Signal the FIQ interrupt request to a connected processor.
- The GIC Virtualization Extensions, that provide hardware support for managing virtualized interrupts.

For more detailed information, please refer to the manual "*Arm Generic Interrupt Controller - Architecture Specification v2.0*".

The integrated GIC can configure interrupts with following operations, Interrupts can be

- Edge-trigger or level-sensitive.
- Enabled or disabled.
- Assigning to secure group or Non-secure group.
- Prioritizing and masking.
- Trigger by software.

There are 4 types of interrupts:

- 16 software generated interrupts(SGIs)
- 4 external Private Peripheral timer Interrupts (PPIs) for each processor.
- 1 internal PPI for each processor.
- 130 Shared Peripheral Interrupt (SPIs).

6.1.6.1 System Interrupt Map

GIC SPI Number	Interrupt Name	Interrupt Description
0 ~ 15	SGI	Software Generated Interrupt
16 ~ 24	Reserved	Reserved
25	VMAINT_INT	Virtual Maintenance Interrupt
26	HVTMR_INT	Hypervisor Timer Interrupt
27	VTMR_INT	Virtual Timer Interrupt
28	Reserved	Reserved
29	SPTMR_INT	Secure Physical Timer Interrupt
30	NSPTMR_INT	Non-secure Physical Timer Interrupt
31	Reserved	Reserved
32	LVD_INT	Low Voltage detection Interrupt
33	A35PMU_INT	A35 PMU Interrupt
34	HWSEM0_INT	Hardware Semaphore 0 Interrupt (CA35, CM4)
35	CKFAIL_INT	Clock failed Interrupt
36	WRHO0_INT	Wormhole 0 Interrupt (CA35, CM4)
37	RTC_INT	Real Time Clock Interrupt
38	TAMPER_INT	Tamper detection Interrupt
39	WDT0_INT	Watchdog timer 0 Interrupt
40	WWDT0_INT	Window Watchdog timer 0 Interrupt
41	EINT0	External Input 0 Interrupt
42	EINT1	External Input 1 Interrupt
43	EINT2	External Input 2 Interrupt
44	EINT3	External Input 3 Interrupt
45	I2C0_INT	I^2C 0 Interrupt
46	GPA_INT	GPIO Port A Interrupt
47	GPB_INT	GPIO Port B Interrupt
48	GPC_INT	GPIO Port C Interrupt
49	GPD_INT	GPIO Port D Interrupt
50	PDMA0_INT	Peripheral DMA 0 Interrupt
51	PDMA1_INT	Peripheral DMA 1 Interrupt
52	DISP_INT	Display Controller (DCUltra) Interrupt
53	CCAP0_INT	CCAP 0 Interrupt
54	CCAP1_INT	CCAP 1 Interrupt
55	GMAC0_INT	GMAC0 Interrupt

56	GMAC1_INT	GMAC1 Interrupt
57	SSMCC_INT	SSMCC Interrupt
58	SSPCC_INT	SSPCC Interrupt
59	GFX_INT	GFX GC520L Interrupt (Graphic Engine)
60	VDE_INT	Video Decoder (VC8000) Interrupt
61	WRHO1_INT	Wormhole 1 Interrupt (CA35, TSI)
62	SDH0_INT	SDH 0 Interrupt
63	SDH1_INT	SDH 1 Interrupt
64	HSUSBD_INT	USB 2.0 High-Speed Device Interrupt
65	HSUSBH0_INT	USB 2.0 High-Speed Host 0 (EHCI) Interrupt
66	HSUSBH1_INT	USB 2.0 High-Speed Host 1 (EHCI) Interrupt
67	USBH0_INT	USB 1.1 Host (OHCI) 0 Interrupt
68	USBH1_INT	USB 1.1 Host (OHCI) 1 Interrupt
69	Reserved	Reserved
70	NAND_INT	NAND Controller Interrupt
71	Reserved	Reserved
72	Reserved	Reserved
73	Reserved	Reserved
74	Reserved	Reserved
75	WDT1_INT	Watchdog timer 1 Interrupt
76	WWDT1_INT	Window Watchdog timer 1 Interrupt
77	PDMA2_INT	Peripheral DMA 2 Interrupt
78	PDMA3_INT	Peripheral DMA 3 Interrupt
79	TMR0_INT	Timer 0 Interrupt
80	TMR1_INT	Timer 1 Interrupt
81	TMR2_INT	Timer 2 Interrupt
82	TMR3_INT	Timer 3 Interrupt
83	BRAKE0_INT	BRAKE0 Interrupt
84	EPWM0P0_INT	EPWM0P0 Interrupt
85	EPWM0P1_INT	EPWM0P1 Interrupt
86	EPWM0P2_INT	EPWM0P2 Interrupt
87	QEI0_INT	QEI0 Interrupt
88	ECAP0_INT	ECAP0 Interrupt
89	QSPI0_INT	QSPI0 Interrupt
90	QSPI1_INT	QSPI1 Interrupt

91	UART0_INT	UART 0 Interrupt
92	UART1_INT	UART 1 Interrupt
93	UART2_INT	UART 2 Interrupt
94	UART3_INT	UART 3 Interrupt
95	UART4_INT	UART 4 Interrupt
96	UART5_INT	UART 5 Interrupt
97	EADC00_INT	EADC00 Interrupt
98	EADC01_INT	EADC01 Interrupt
99	EADC02_INT	EADC02 Interrupt
100	EADC03_INT	EADC03 Interrupt
101	I2C1_INT	I^2C 1 Interrupt
102	I2S0_INT	I^2S 0 Interrupt
103	MCAN00_INT	MCAN 00 Interrupt
104	SC0_INT	Smart Card 0 Interrupt
105	GPE_INT	GPIO Port E Interrupt
106	GPF_INT	GPIO Port F Interrupt
107	GPG_INT	GPIO Port G Interrupt
108	GPH_INT	GPIO Port H Interrupt
109	GPI_INT	GPIO Port I Interrupt
110	GPJ_INT	GPIO Port J Interrupt
111	KPI_INT	KPI Interrupt
112	ADC0_INT	ADC 0 (Touch Panel ADC) Interrupt
113	TMR4_INT	Timer 4 Interrupt
114	TMR5_INT	Timer 5 Interrupt
115	BRAKE1_INT	BRAKE1 Interrupt
116	EPWM1P0_INT	EPWM1P0 Interrupt
117	EPWM1P1_INT	EPWM1P1 Interrupt
118	EPWM1P2_INT	EPWM1P2 Interrupt
119	QEI1_INT	QEI1 Interrupt
120	ECAP1_INT	ECAP1 Interrupt
121	SPI0_INT	SPI0 Interrupt
122	SPI1_INT	SPI1 Interrupt
123	UART6_INT	UART 6 Interrupt
124	UART7_INT	UART 7 Interrupt
125	UART8_INT	UART 8 Interrupt

126	UART9_INT	UART 9 Interrupt
127	UART10_INT	UART 10 Interrupt
128	UART11_INT	UART 11 Interrupt
129	I2C2_INT	I^2C 2 Interrupt
130	I2C3_INT	I^2C 3 Interrupt
131	I2S1_INT	I^2S 1 Interrupt
132	MCAN10_INT	MCAN 10 Interrupt
133	SC1_INT	Smart Card 1 Interrupt
134	GPK_INT	GPIO Port K Interrupt
135	GPL_INT	GPIO Port L Interrupt
136	GPM_INT	GPIO Port M Interrupt
137	GPN_INT	GPIO Port N Interrupt
138	TMR6_INT	Timer 6 Interrupt
139	TMR7_INT	Timer 7 Interrupt
140	TMR8_INT	Timer 8 Interrupt
141	TMR9_INT	Timer 9 Interrupt
142	BRAKE2_INT	BRAKE2 Interrupt
143	EPWM2P0_INT	EPWM2P0 Interrupt
144	EPWM2P1_INT	EPWM2P1 Interrupt
145	EPWM2P2_INT	EPWM2P2 Interrupt
146	QEI2_INT	QEI2 Interrupt
147	ECAP2_INT	ECAP2 Interrupt
148	SPI2_INT	SPI2 Interrupt
149	SPI3_INT	SPI3 Interrupt
150	UART12_INT	UART 12 Interrupt
151	UART13_INT	UART 13 Interrupt
152	UART14_INT	UART 14 Interrupt
153	UART15_INT	UART 15 Interrupt
154	UART16_INT	UART 16 Interrupt
155	I2C4_INT	I^2C 4 Interrupt
156	I2C5_INT	I^2C 5 Interrupt
157	MCAN20_INT	MCAN 20 Interrupt
158	MCAN30_INT	MCAN 30 Interrupt
159	TMR10_INT	Timer 10 Interrupt
160	TMR11_INT	Timer 11 Interrupt

161	Reserved	Reserved
162	Reserved	Reserved
163	MCAN01_INT	MCAN 01 Interrupt
164	MCAN11_INT	MCAN 11 Interrupt
165	MCAN21_INT	MCAN 21 Interrupt
166	MCAN31_INT	MCAN 31 Interrupt
167	PWRWU_INT	Power Down Wake-up interrupt
168	DDRPOISION_INT	DDR Out of range Interrupt
169	Reserved	Reserved

Table 6.1-1 GIC Interrupt Number Table

6.1.6.2 Register Map

R: read only, W: write only, R/W: both read and write, C: Only value 0 can be written

Register	Offset	R/W	Description	Reset Value
GIC Base Address: **GICD_BA = 0x5080_1000** **GICC_BA = 0x5080_2000** **GICH_BA = 0x5080_4000** **GICV_BA = 0x5080_6000**				
GICD_CTLR	GICD_BA+0x000	R/W	Distributor Control Register	0x00000000
GICD_TYPER	GICD_BA+0x004	R	Interrupt Controller Type Register	0x0000FC25
GICD_IIDR	GICD_BA+0x008	R	Distributor Implementer Identification Register	0x0200143B
GICD_IGROUPRn	GICD_BA+0x080-0x0BC	R/W	Interrupt Group Registers	0x00000000
GICD_ISENABLER0	GICD_BA+0x100	R/W	Interrupt Set-Enable Registers	0x0000FFFF
GICD_ISENABLERn	GICD_BA+0x104-0x13C	R/W	Interrupt Set-Enable Registers	0x00000000
GICD_ICENABLER0	GICD_BA+0x180	R/W	Interrupt Clear-Enable Registers	0x0000FFFF
GICD_ICENABLERn	GICD_BA+0x184-0x1BC	R/W	Interrupt Clear-Enable Registers	0x00000000
GICD_ISPENDRn	GICD_BA+0x200-0x23C	R/W	Interrupt Set-Pending Registers	0x00000000
GICD_ICPENDRn	GICD_BA+0x280-0x2BC	R/W	Interrupt Clear-Pending Registers	0x00000000
GICD_ISACTIVERn	GICD_BA+0x300-0x33C	R/W	Interrupt Set-Active Registers	0x00000000
GICD_ICACTIVERn	GICD_BA+0x380-0x3BC	R/W	Interrupt Clear-Active Registers	0x00000000
GICD_IPRIORITYRn	GICD_BA+0x400-0x5FC	R/W	Interrupt Priority Registers	0x00000000
GICD_ITARGETSR0	GICD_BA+0x800	R	Interrupt Processor Targets Registers	0x01010101
GICD_ITARGETSR1	GICD_BA+0x804	R	Interrupt Processor Targets Registers	0x01010101
GICD_ITARGETSR2	GICD_BA+0x808	R	Interrupt Processor Targets Registers	0x01010101
GICD_ITARGETSR3	GICD_BA+0x80C	R	Interrupt Processor Targets Registers	0x01010101
GICD_ITARGETSR4	GICD_BA+0x810	R	Interrupt Processor Targets Registers	0x00000000
GICD_ITARGETSR5	GICD_BA+0x814	R	Interrupt Processor Targets Registers	0x00000000
GICD_ITARGETSR6	GICD_BA+0x818	R	Interrupt Processor Targets Registers	0x01010100
GICD_ITARGETSR7	GICD_BA+0x81C	R	Interrupt Processor Targets Registers	0x01010101
GICD_ITARGETSRn	GICD_BA+0x820-0x9FC	R/W	Interrupt Processor Targets Registers	0x00000000
GICD_ICFGR0	GICD_BA+0xC00	R	Interrupt Configuration Registers	0xAAAAAAAA
GICD_ICFGR1	GICD_BA+0xC04	R	Interrupt Configuration Registers	0x55540000
GICD_ICFGRn	GICD_BA+0xC08-0xC3C	R/W	Interrupt Configuration Registers	0x55555555

GICD_PPISR	GICD_BA+0xD00	R	Private Peripheral Interrupt Status Register	0x00000000
GICD_SPISRn	GICD_BA+0xD04-0xD3C	R	Shared Peripheral Interrupt Status Registers	0x00000000
GICD_SGIR	GICD_BA+0xF00	W	Software Generated Interrupt Register	0xXXXXXXXX
GICD_CPENDSGIRn	GICD_BA+0xF10-0xF1C	R/W	SGI Clear-Pending Registers	0x00000000
GICD_SPENDSGIRn	GICD_BA+0xF20-0xF2C	R/W	SGI Set-Pending Registers	0x00000000
GICD_PIDR4	GICD_BA+0xFD0	R	Peripheral ID 4 Register	0x00000004
GICD_PIDR5	GICD_BA+0xFD4	R	Peripheral ID 5 Register	0x00000000
GICD_PIDR6	GICD_BA+0xFD8	R	Peripheral ID 6 Register	0x00000000
GICD_PIDR7	GICD_BA+0xFDC	R	Peripheral ID 7 Register	0x00000000
GICD_PIDR0	GICD_BA+0xFE0	R	Peripheral ID 0 Register	0x00000090
GICD_PIDR1	GICD_BA+0xFE4	R	Peripheral ID 1 Register	0x000000B4
GICD_PIDR2	GICD_BA+0xFE8	R	Peripheral ID 2 Register	0x0000002B
GICD_PIDR3	GICD_BA+0xFEC	R	Peripheral ID 3 Register	0x00000000
GICD_CIDR0	GICD_BA+0xFF0	R	Component ID 0 Register	0x0000000D
GICD_CIDR1	GICD_BA+0xFF4	R	Component ID 1 Register	0x000000F0
GICD_CIDR2	GICD_BA+0xFF8	R	Component ID 2 Register	0x00000005
GICD_CIDR3	GICD_BA+0xFFC	R	Component ID 3 Register	0x000000B1
GICC_CTLR	GICC_BA+0x0000	R/W	CPU Interface Control Register	0x00000000
GICC_PMR	GICC_BA+0x0004	R/W	Interrupt Priority Mask Register	0x00000000
GICC_BPR	GICC_BA+0x0008	R/W	Binary Point Register	0x00000002
GICC_IAR	GICC_BA+0x000C	R	Interrupt Acknowledge Register	0x000003FF
GICC_EOIR	GICC_BA+0x0010	W	End of Interrupt Register	0xXXXXXXXX
GICC_RPR	GICC_BA+0x0014	R	Running Priority Register	0x000000FF
GICC_HPPIR	GICC_BA+0x0018	R	Highest Priority Pending Interrupt Register	0x000003FF
GICC_ABPR	GICC_BA+0x001C	R/W	Aliased Binary Point Register	0x00000003
GICC_AIAR	GICC_BA+0x0020	R	Aliased Interrupt Acknowledge Register	0x000003FF
GICC_AEOIR	GICC_BA+0x0024	W	Aliased End of Interrupt Register	0xXXXXXXXX
GICC_AHPPIR	GICC_BA+0x0028	R	Aliased Highest Priority Pending Interrupt Registered	0x000003FF
GICC_APR0	GICC_BA+0x00D0	R/W	Active Priority Register	0x00000000
GICC_NSAPR0	GICC_BA+0x00E0	R/W	Non-Secure Active Priority Register	0x00000000
GICC_IIDR	GICC_BA+0x00FC	R	CPU Interface Identification Register	0x0202143B

GICC_DIR	GICC_BA+0x1000	W	Deactivate Interrupt Register	0xXXXXXXXX
GICH_HCR	GICH_BA+0x000	R/W	Hypervisor Control Register	0x00000000
GICH_VTR	GICH_BA+0x004	R	VGIC Type Register	0x90000003
GICH_VMCR	GICH_BA+0x008	R/W	Virtual Machine Control Register	0x004C0000
GICH_MISR	GICH_BA+0x010	R	Maintenance Interrupt Status Register	0x00000000
GICH_EISR0	GICH_BA+0x020	R	End of Interrupt Status Register	0x00000000
GICH_ELSR0	GICH_BA+0x030	R	Empty List register Status Register	0x0000000F
GICH_APR0	GICH_BA+0x0F0	R/W	Active Priority Register	0x00000000
GICH_LR0	GICH_BA+0x100	R/W	List Register 0	0x00000000
GICH_LR1	GICH_BA+0x104	R/W	List Register 1	0x00000000
GICH_LR2	GICH_BA+0x108	R/W	List Register 2	0x00000000
GICH_LR3	GICH_BA+0x10C	R/W	List Register 3	0x00000000
GICV_CTLR	GICV_BA+0x0000	R/W	Virtual Machine Control Register	0x00000000
GICV_PMR	GICV_BA+0x0004	R/W	VM Priority Mask Register	0x00000000
GICV_BPR	GICV_BA+0x0008	R/W	VM Binary Point Register	0x00000002
GICV_IAR	GICV_BA+0x000C	R	VM Interrupt Acknowledge Register	0x000003FF
GICV_EOIR	GICV_BA+0x0010	W	VM End of Interrupt Register	0xXXXXXXXX
GICV_RPR	GICV_BA+0x0014	R	VM Running Priority Register	0x000000FF
GICV_HPPIR	GICV_BA+0x0018	R	VM Highest Priority Pending Interrupt Register	0x000003FF
GICV_ABPR	GICV_BA+0x001C	R/W	VM Aliased Binary Point Register	0x00000003
GICV_AIAR	GICV_BA+0x0020	R	VM Aliased Interrupt Acknowledge Register	0x000003FF
GICV_AEOIR	GICV_BA+0x0024	W	VM Aliased End of Interrupt Register	0xXXXXXXXX
GICV_AHPPIR	GICV_BA+0x0028	R	VM Aliased Highest Priority Pending Interrupt Register	0x000003FF
GICV_APR0	GICV_BA+0x00D0	R/W	VM Active Priority Register	0x00000000
GICV_IIDR	GICV_BA+0x00FC	R	VM CPU Interface Identification Register	0x0202143B
GICV_DIR	GICV_BA+0x1000	W	VM Deactivate Interrupt Register	0xXXXXXXXX

6.2 Arm Cortex-M4 Core

6.2.1 Overview

The Cortex-M4 core, a configurable, multistage, 32-bit RISC processor, has three AMBA AHB-Lite interfaces for best parallel performance and includes an NVIC component. The processor with optional hardware debug functionality can execute Thumb code and is compatible with other Cortex-M profile processors. The profile supports two modes ,–the Thread mode and Handler mode. Handler mode is entered as a result of an exception. An exception return can only be issued in Handler mode. Thread mode is entered on Reset, and can be entered as a result of an exception return. The Cortex-M4F core is a processor with the same capability as the Cortex-M4 core and includes floating point arithmetic functionality. The NuMicro MA35D1 series is embedded with Cortex-M4F core. Throughout this document, the name Cortex-M4 refers to both Cortex-M4 and Cortex-M4F cores.

6.2.2 Features

- A low gate count processor core, with low latency interrupt processing that has:
  - A subset of the Thumb instruction set, defined in the ARMv7-M Architecture Reference Manual
  - Banked Stack Pointer (SP)
  - Hardware integer divide instructions, SDIV and UDIV
  - Handler and Thread modes
  - Thumb and Debug states
  - Support for interruptible-continued instructions LDM, STM, PUSH, and POP for low interrupt latency
  - Automatic processor state saving and restoration for low latency Interrupt Service Routine (ISR) entry and exit
  - Support for ARMv7 big-endian byte-invariant or little-endian accesses
  - Support for ARMv7 unaligned accesses
- Floating Point Unit (FPU) in the Cortex-M4F processor providing:
  - 32-bit instructions for single-precision (C float) data-processing operations
  - Combined Multiply and Accumulate instructions for increased precision (Fused MAC)
  - Hardware support for conversion, addition, subtraction, multiplication with optional accumulate, division, and square-root
  - Hardware support for denormals and all IEEE rounding modes
  - 32 dedicated 32-bit single precision registers, also addressable as 16 double-word registers
  - Decoupled three stage pipeline
- Nested Vectored Interrupt Controller (NVIC) closely integrated with the processor core to achieve low latency interrupt processing. Features include:
  - External interrupts. Configurable from 1 to 240 (the NuMicro MA35D1 series configured with 111 interrupts)
  - Bits of priority, configurable from 3 to 8
  - Dynamic reprioritization of interrupts
  - Priority grouping which enables selection of preempting interrupt levels and nonpreempting interrupt levels

- Support for tril-chaining and late arrival of interrupts, which enables back-to-back interrupt processing without the overhead of state saving and restoration between interrupts.
- Processor state automatically saved on interrupt entry, and restored on interrupt exit with on instruction overhead
- Support for Wake-up Interrupt Controller (WIC) with Ultra-low Power Sleep mode

- Memory Protection Unit (MPU). An optional MPU for memory protection, including:
  - Eight memory regions
  - Sub Region Disable (SRD), enabling efficient use of memory regions
  - The ability to enable a background region that implements the default memory map attributes
- Low-cost debug solution that features:
  - Debug access to all memory and registers in the system, including access to memory mapped devices, access to internal core registers when the core is halted, and access to debug control registers even while SYSRESETn is asserted.
  - Serial Wire Debug Port(SW-DP) or Serial Wire JTAG Debug Port (SWJ-DP) debug access
  - Optional Flash Patch and Breakpoint (FPB) unit for implementing breakpoints and code patches
  - Optional Data Watchpoint and Trace (DWT) unit for implementing watchpoints, data tracing, and system profiling
  - Optional Instrumentation Trace Macrocell (ITM) for support of printf() style debugging
  - Optional Trace Port Interface Unit (TPIU) for bridging to a Trace Port Analyzer (TPA), including Single Wire Output (SWO) mode
  - Optional Embedded Trace Macrocell (ETM) for instruction trace.
- Bus interfaces:
  - Three Advanced High-performance Bus-Lite (AHB-Lite) interfaces: ICode, Dcode, and System bus interfaces
  - Private Peripheral Bus (PPB) based on Advanced Peripheral Bus (APB) interface
  - Bit-band support that includes atomic bit-band write and read operations.
  - Memory access alignment
  - Write buffer for buffering of write data
  - Exclusive access transfers for multiprocessor systems

6.2.3 Block Diagram

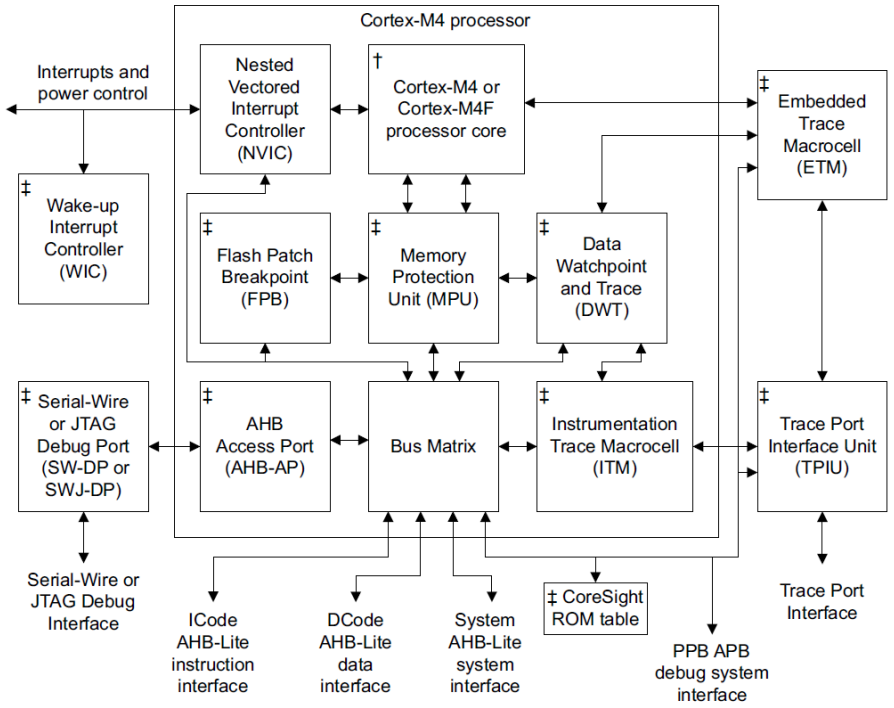


Figure 6.2-1 Arm Cortex-M4 Core Block Diagram

6.2.4 System Timer (SysTick)

The Cortex-M4 includes an integrated system timer, SysTick, which provides a simple, 24-bit clear-on-write, decrementing, wrap-on-zero counter with a flexible control mechanism. The counter can be used as a Real Time Operating System (RTOS) tick timer or as a simple counter.

When system timer is enabled, it will count down from the value in the SysTick Current Value Register (SYST_VAL) to zero, and reload (wrap) to the value in the SysTick Reload Value Register (SYST_LOAD) on the next clock cycle, and then decrement on subsequent clocks. When the counter transitions to zero, the COUNTFLAG status bit is set. The COUNTFLAG bit clears on reads.

The SYST_VAL value is UNKNOWN on reset. Software should write to the register to clear it to zero before enabling the feature. This ensures the timer will count from the SYST_LOAD value rather than an arbitrary value when it is enabled.

If the SYST_LOAD is zero, the timer will be maintained with a current value of zero after it is reloaded with this value. This mechanism can be used to disable the feature independently from the timer enable bit.

For more detailed information, please refer to the "*Arm Cortex-M4 Technical Reference Manual" and "Arm v7-M Architecture Reference Manual*".

6.2.4.1 Register Map

R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value
SYST Base Address: **SCS_BA = 0xE000_E000**				
SYST_CTRL	SCS_BA+0x10	R/W	SysTick Control and Status Register	0x0000_0000
SYST_LOAD	SCS_BA+0x14	R/W	SysTick Reload Value Register	0xXXXX_XXXX
SYST_VAL	SCS_BA+0x18	R/W	SysTick Current Value Register	0xXXXX_XXXX

6.2.4.2 Register Description

SysTick Control and Status Register (SYST_CTRL)

Register	Offset	R/W	Description	Reset Value
SYST_CTRL	SCS_BA+0x10	R/W	SysTick Control and Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							COUNTFLAG
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					CLKSRC	TICKINT	ENABLE

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**COUNTFLAG**	**System Tick Counter Flag** Returns 1 if timer counted to 0 since last time this register was read. COUNTFLAG is set by a count transition from 1 to 0. COUNTFLAG is cleared on read or by a write to the Current Value register.
[15:3]	**Reserved**	Reserved.
[2]	**CLKSRC**	**System Tick Clock Source Selection** 0 = Clock source is the (optional) external reference clock. 1 = Core clock used for SysTick.
[1]	**TICKINT**	**System Tick Interrupt Enabled** 0 = Counting down to 0 does not cause the SysTick exception to be pended. Software can use COUNTFLAG to determine if a count to zero has occurred. 1 = Counting down to 0 will cause the SysTick exception to be pended. Clearing the SysTick current value register by a register write in software will not cause SysTick to be pended.
[0]	**ENABLE**	**System Tick Counter Enabled** 0 = Counter Disabled. 1 = Counter will operate in a multi-shot manner.

SysTick Reload Value Register (SYST_LOAD)

Register	Offset	R/W	Description	Reset Value
SYST_LOAD	SCS_BA+0x14	R/W	SysTick Reload Value Register	0xXXXX_XXXX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RELOAD							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RELOAD							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RELOAD							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**RELOAD**	**System Tick Reload Value** Value to load into the Current Value register when the counter reaches 0.

SysTick Current Value Register (SYST_VAL)

Register	Offset	R/W	Description	Reset Value
SYST_VAL	SCS_BA+0x18	R/W	SysTick Current Value Register	0xXXXX_XXXX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CURRENT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CURRENT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CURRENT							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**CURRENT**	**System Tick Current Value** Current counter value. This is the value of the counter at the time it is sampled. The counter does not provide read-modify-write protection. The register is write-clear. A software write of any value will clear the register to 0.

6.2.5 Nested Vectored Interrupt Controller (NVIC)

The NVIC and the processor core interface are closely coupled to enable low latency interrupt processing and efficient processing of late arriving interrupts. The NVIC maintains knowledge of the stacked, or nested, interrupts to enable tail-chaining of interrupts. You can only fully access the NVIC from privileged mode, but you can cause interrupts to enter a pending state in user mode if you enable the Configuration and Control Register. Any other user mode access causes a bus fault. You can access all NVIC registers using byte, halfword, and word accesses unless otherwise stated. NVIC registers are located within the SCS (System Control Space). All NVIC registers and system debug registers are little-endian regardless of the endianness state of the processor.

The NVIC supports:

- An implementation-defined number of interrupts, in the range 1-111 interrupts.
- A programmable priority level of 0-16 for each interrupt; a higher level corresponds to a lower priority, so level 0 is the highest interrupt priority.
- Level and pulse detection of interrupt signals.
- Dynamic reprioritization of interrupts.
- Grouping of priority values into group priority and subpriority fields.
- Interrupt tail-chaining.
- An external Non Maskable Interrupt (NMI)
- WIC with Ultra-low Power Sleep mode support

The processor automatically stacks its state on exception entry and unstacks this state on exception exit, with no instruction overhead. This provides low latency exception handling.

6.2.5.1 Exception Model and System Interrupt Map

Table 6.2-1 lists the exception model supported by MA35D1 Series. Software can set 16 levels of priority on some of these exceptions as well as on all interrupts. The highest user-configurable priority is denoted as “0x00” and the lowest priority is denoted as “0xF0” (The 4-LSB always 0). The default priority of all the user-configurable interrupts is “0x00”. Note that priority “0” is treated as the fourth priority on the system, after three system exceptions “Reset”, “NMI” and “Hard Fault”.

When any interrupt is accepted, the processor will automatically fetch the starting address of the interrupt service routine (ISR) from a vector table in memory. On system reset, the vector table is fixed at address 0x00000000. Privileged software can write to the VTOR to relocate the vector table start address to a different memory location, in the range 0x00000080 to 0x3FFFFF80,

The vector table contains the initialization value for the stack pointer on reset, and the entry point addresses for all exception handlers. The vector number on previous page defines the order of entries in the vector table associated with exception handler entry as illustrated in previous section.

Exception Type	Vector Number	Vector Address	Priority
Reset	1	0x00000004	-3
NMI	2	0x00000008	-2
Hard Fault	3	0x0000000C	-1
Memory Manager Fault	4	0x00000010	Configurable
Bus Fault	5	0x00000014	Configurable
Usage Fault	6	0x00000018	Configurable

Reserved	7 ~ 10		Reserved
SVCall	11	0x0000002C	Configurable
Debug Monitor	12	0x00000030	Configurable
Reserved	13		Reserved
PendSV	14	0x00000038	Configurable
SysTick	15	0x0000003C	Configurable
Interrupt (IRQ0 ~ IRQ)	16 ~ 111	0x00000000 + (Vector Number)*4	Configurable

Table 6.2-1 Exception Model

Vector Number	Interrupt Number (Bit In Interrupt Registers)	Interrupt Name	Interrupt Description
0 ~ 15	-	-	System exceptions
16	0	LVD_INT	Low Voltage detection Interrupt
17	1	Reserved	Reserved
18	2	PWRWU_INT	Power Down Wake-up interrupt
19	3	HSEM0_INT	Hardware Semaphore 0 Interrupt (CA35, CM4)
20	4	CKFAIL_INT	Clock failed Interrupt
21	5	WRHO0_INT	Wormhole 0 Interrupt (CA35, CM4)
22	6	RTC_INT	Real Time Clock Interrupt
23	7	TAMPER_INT	Tamper detection Interrupt
24	8	WDT2_INT	Watchdog timer 2 Interrupt
25	9	WWDT2_INT	Window Watchdog timer 2 Interrupt
26	10	EINT0	External Input 0 Interrupt
27	11	EINT1	External Input 1 Interrupt
28	12	EINT2	External Input 2 Interrupt
29	13	EINT3	External Input 3 Interrupt
30	14	Reserved	Reserved
31	15	Reserved	Reserved
32	16	GPA_INT	GPIO Port A Interrupt
33	17	GPB_INT	GPIO Port B Interrupt
34	18	GPC_INT	GPIO Port C Interrupt
35	19	GPD_INT	GPIO Port D Interrupt
36	20	Reserved	Reserved
37	21	Reserved	Reserved

38	22	TMR2_INT	Timer 2 Interrupt
39	23	TMR3_INT	Timer 3 Interrupt
40	24	BRAKE0_INT	BRAKE0 Interrupt
41	25	EPWM0P0_INT	EPWM0P0 Interrupt
42	26	EPWM0P1_INT	EPWM0P1 Interrupt
43	27	EPWM0P2_INT	EPWM0P2 Interrupt
44	28	QEI0_INT	QEI0 Interrupt
45	29	ECAP0_INT	ECAP0 Interrupt
46	30	Reserved	Reserved
47	31	QSPI1_INT	QSPI1 Interrupt
48	32	Reserved	Reserved
49	33	Reserved	Reserved
50	34	Reserved	Reserved
51	35	UART1_INT	UART 1 Interrupt
52	36	UART2_INT	UART 2 Interrupt
53	37	UART3_INT	UART 3 Interrupt
54	38	UART4_INT	UART 4 Interrupt
55	39	UART5_INT	UART 5 Interrupt
56	40	EADC00_INT	EADC00 Interrupt
57	41	EADC01_INT	EADC01 Interrupt
58	42	EADC02_INT	EADC02 Interrupt
59	43	EADC03_INT	EADC03 Interrupt
60	44	Reserved	Reserved
61	45	I2C1_INT	I^2C 1 Interrupt
62	46	I2S0_INT	I^2S 0 Interrupt
63	47	MCAN00_INT	MCAN 00 Interrupt
64	48	SC0_INT	Smart Card 0 Interrupt
65	49	GPE_INT	GPIO Port E Interrupt
66	50	GPF_INT	GPIO Port F Interrupt
67	51	GPG_INT	GPIO Port G Interrupt
68	52	GPH_INT	GPIO Port H Interrupt
69	53	GPI_INT	GPIO Port I Interrupt
70	54	GPJ_INT	GPIO Port J Interrupt
71	55	TMR4_INT	Timer 4 Interrupt
72	56	TMR5_INT	Timer 5 Interrupt

73	57	TMR6_INT	Timer 6 Interrupt
74	58	TMR7_INT	Timer 7 Interrupt
75	59	BRAKE1_INT	BRAKE1 Interrupt
76	60	EPWM1P0_INT	EPWM1P0 Interrupt
77	61	EPWM1P1_INT	EPWM1P1 Interrupt
78	62	EPWM1P2_INT	EPWM1P2 Interrupt
79	63	QEI1_INT	QEI1 Interrupt
80	64	ECAP1_INT	ECAP1 Interrupt
81	65	SPI0_INT	SPI0 Interrupt
82	66	SPI1_INT	SPI1 Interrupt
83	67	PDMA2_INT	Peripheral DMA 2 Interrupt
84	68	PDMA3_INT	Peripheral DMA 3 Interrupt
85	69	UART6_INT	UART 6 Interrupt
86	70	UART7_INT	UART 7 Interrupt
87	71	UART8_INT	UART 8 Interrupt
88	72	UART9_INT	UART 9 Interrupt
89	73	UART10_INT	UART 10 Interrupt
90	74	UART11_INT	UART 11 Interrupt
91	75	I2C2_INT	I^2C 2 Interrupt
92	76	I2C3_INT	I^2C 3 Interrupt
93	77	I2S1_INT	I^2S 1 Interrupt
94	78	MCAN10_INT	MCAN 10 Interrupt
95	79	SC1_INT	Smart Card 1 Interrupt
96	80	GPK_INT	GPIO Port K Interrupt
97	81	GPL_INT	GPIO Port L Interrupt
98	82	GPM_INT	GPIO Port M Interrupt
99	83	GPN_INT	GPIO Port N Interrupt
100	84	TMR8_INT	Timer 8 Interrupt
101	85	TMR9_INT	Timer 9 Interrupt
102	86	TMR10_INT	Timer 10 Interrupt
103	87	TMR11_INT	Timer 11 Interrupt
104	88	BRAKE2_INT	BRAKE2_Interrupt
105	89	EPWM2P0_INT	EPWM2P0 Interrupt
106	90	EPWM2P1_INT	EPWM2P1 Interrupt
107	91	EPWM2P2_INT	EPWM2P2 Interrupt

108	92	QEI2_INT	QEI2 Interrupt
109	93	ECAP2_INT	ECAP2 Interrupt
110	94	SPI2_INT	SPI2 Interrupt
111	95	SPI3_INT	SPI3 Interrupt
112	96	UART12_INT	UART 12 Interrupt
113	97	UART13_INT	UART 13 Interrupt
114	98	UART14_INT	UART 14 Interrupt
115	99	UART15_INT	UART 15 Interrupt
116	100	UART16_INT	UART 16 Interrupt
117	101	I2C4_INT	I^2C 4 Interrupt
118	102	I2C5_INT	I^2C 5 Interrupt
119	103	MCAN20_INT	MCAN 20 Interrupt
120	104	MCAN30_INT	MCAN 30 Interrupt
121	105	KPI_INT	KPI Interrupt
122	106	MCAN01_INT	MCAN 01 Interrupt
123	107	MCAN11_INT	MCAN 11 Interrupt
124	108	MCAN21_INT	MCAN 21 Interrupt
125	109	MCAN31_INT	MCAN 31 Interrupt
126	110	ADC0_INT	ADC 0 (Touch Panel ADC) Interrupt

Table 6.2-2 Interrupt Number Table

6.2.5.2 Operation Description

NVIC interrupts can be enabled and disabled by writing to their corresponding Interrupt Set-Enable or Interrupt Clear-Enable register bit-field. The registers use a write-1-to-enable and write-1-to-clear policy, both registers reading back the current enabled state of the corresponding interrupts. When an interrupt is disabled, interrupt assertion will cause the interrupt to become Pending, however, the interrupt will not activate. If an interrupt is Active when it is disabled, it remains in its Active state until cleared by reset or an exception return. Clearing the enable bit prevents new activations of the associated interrupt.

NVIC interrupts can be pended/un-pended using a complementary pair of registers to those used to enable/disable the interrupts, named the Set-Pending Register and Clear-Pending Register respectively. The registers use a write-1-to-enable and write-1-to-clear policy, both registers reading back the current pended state of the corresponding interrupts. The Clear-Pending Register has no effect on the execution status of an Active interrupt.

NVIC interrupts are prioritized by updating an 8-bit field within a 32-bit register (each register supporting four interrupts).

The general registers associated with the NVIC are all accessible from a block of memory in the System Control Space and will be described in next section.

6.2.5.3 Register Map

R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value
NVIC Base Address: **NVIC_BA = 0xE000_E100**				
NVIC_ISER0	NVIC_BA+0x00	R/W	IRQ0 ~ IRQ108 Set-enable Control Register	0x0000_0000
NVIC_ISER1	NVIC_BA+0x04	R/W	IRQ0 ~ IRQ108 Set-enable Control Register	0x0000_0000
NVIC_ISER2	NVIC_BA+0x08	R/W	IRQ0 ~ IRQ108 Set-enable Control Register	0x0000_0000
NVIC_ISER3	NVIC_BA+0x0C	R/W	IRQ0 ~ IRQ108 Set-enable Control Register	0x0000_0000
NVIC_ICER0	NVIC_BA+0x80	R/W	IRQ0 ~ IRQ108 Clear-enable Control Register	0x0000_0000
NVIC_ICER1	NVIC_BA+0x84	R/W	IRQ0 ~ IRQ108 Clear-enable Control Register	0x0000_0000
NVIC_ICER2	NVIC_BA+0x88	R/W	IRQ0 ~ IRQ108 Clear-enable Control Register	0x0000_0000
NVIC_ICER3	NVIC_BA+0x8C	R/W	IRQ0 ~ IRQ108 Clear-enable Control Register	0x0000_0000
NVIC_ISPR0	NVIC_BA+0x100	R/W	IRQ0 ~ IRQ108 Set-pending Control Register	0x0000_0000
NVIC_ISPR1	NVIC_BA+0x104	R/W	IRQ0 ~ IRQ108 Set-pending Control Register	0x0000_0000
NVIC_ISPR2	NVIC_BA+0x108	R/W	IRQ0 ~ IRQ108 Set-pending Control Register	0x0000_0000
NVIC_ISPR3	NVIC_BA+0x10C	R/W	IRQ0 ~ IRQ108 Set-pending Control Register	0x0000_0000
NVIC_ICPR0	NVIC_BA+0x180	R/W	IRQ0 ~ IRQ108 Clear-pending Control Register	0x0000_0000
NVIC_ICPR1	NVIC_BA+0x184	R/W	IRQ0 ~ IRQ108 Clear-pending Control Register	0x0000_0000
NVIC_ICPR2	NVIC_BA+0x188	R/W	IRQ0 ~ IRQ108 Clear-pending Control Register	0x0000_0000
NVIC_ICPR3	NVIC_BA+0x18C	R/W	IRQ0 ~ IRQ108 Clear-pending Control Register	0x0000_0000
NVIC_IABR0	NVIC_BA+0x200	R/W	IRQ0 ~ IRQ108 Active Bit Register	0x0000_0000
NVIC_IABR1	NVIC_BA+0x204	R/W	IRQ0 ~ IRQ108 Active Bit Register	0x0000_0000
NVIC_IABR2	NVIC_BA+0x208	R/W	IRQ0 ~ IRQ108 Active Bit Register	0x0000_0000
NVIC_IABR3	NVIC_BA+0x20C	R/W	IRQ0 ~ IRQ108 Active Bit Register	0x0000_0000
NVIC_IPRn **n=0,1..28**	NVIC_BA+0x300 +0x4*n	R/W	IRQ0 ~ IRQ71 Priority Control Register	0x0000_0000
STIR	NVIC_BA+0xE00	R/W	Software Trigger Interrupt Registers	0x0000_0000

6.2.5.4 Register Description

IRQ0 ~ IRQ108 Set-enable Control Register (NVIC_ISER0)

Register	Offset	R/W	Description	Reset Value
NVIC_ISER0	NVIC_BA+0x00	R/W	IRQ0 ~ IRQ108 Set-enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
SETENA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SETENA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SETENA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SETENA							

Bits	Description	
[31:0]	**SETENA**	**Interrupt Set Enable Bit** The NVIC_ISER0-NVIC_ISER3 registers enable interrupts, and show which interrupts are enabled Write Operation: 0 = No effect. 1 = Interrupt Enabled. Read Operation: 0 = Interrupt Disabled. 1 = Interrupt Enabled.

IRQ0 ~ IRQ108 Set-enable Control Register (NVIC_ISER1)

Register	Offset	R/W	Description	Reset Value
NVIC_ISER1	NVIC_BA+0x04	R/W	IRQ0 ~ IRQ108 Set-enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
SETENA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SETENA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SETENA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SETENA							

Bits	Description	
[31:0]	**SETENA**	**Interrupt Set Enable Bit** The NVIC_ISER0-NVIC_ISER3 registers enable interrupts, and show which interrupts are enabled Write Operation: 0 = No effect. 1 = Interrupt Enabled. Read Operation: 0 = Interrupt Disabled. 1 = Interrupt Enabled.

IRQ0 ~ IRQ108 Set-enable Control Register (NVIC_ISER2)

Register	Offset	R/W	Description	Reset Value
NVIC_ISER2	NVIC_BA+0x08	R/W	IRQ0 ~ IRQ108 Set-enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
SETENA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SETENA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SETENA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SETENA							

Bits	Description	
[31:0]	**SETENA**	**Interrupt Set Enable Bit** The NVIC_ISER0-NVIC_ISER3 registers enable interrupts, and show which interrupts are enabled Write Operation: 0 = No effect. 1 = Interrupt Enabled. Read Operation: 0 = Interrupt Disabled. 1 = Interrupt Enabled.

IRQ0 ~ IRQ108 Set-enable Control Register (NVIC_ISER3)

Register	Offset	R/W	Description	Reset Value
NVIC_ISER3	NVIC_BA+0x0C	R/W	IRQ0 ~ IRQ108 Set-enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
SETENA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SETENA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SETENA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SETENA							

Bits	Description	
[31:0]	**SETENA**	**Interrupt Set Enable Bit** The NVIC_ISER0-NVIC_ISER3 registers enable interrupts, and show which interrupts are enabled Write Operation: 0 = No effect. 1 = Interrupt Enabled. Read Operation: 0 = Interrupt Disabled. 1 = Interrupt Enabled.

IRQ0 ~ IRQ108 Clear-enable Control Register (NVIC_ICER0)

Register	Offset	R/W	Description	Reset Value
NVIC_ICER0	NVIC_BA+0x80	R/W	IRQ0 ~ IRQ108 Clear-enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
CALENA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CALENA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CALENA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CALENA							

Bits	Description	
[31:0]	**CALENA**	**Interrupt Clear Enable Bit** The NVIC_ICER0-NVIC_ICER3 registers disable interrupts, and show which interrupts are enabled. Write Operation: 0 = No effect. 1 = Interrupt Disabled. Read Operation: 0 = Interrupt Disabled. 1 = Interrupt Enabled.

IRQ0 ~ IRQ108 Clear-enable Control Register (NVIC_ICER1)

Register	Offset	R/W	Description	Reset Value
NVIC_ICER1	NVIC_BA+0x84	R/W	IRQ0 ~ IRQ108 Clear-enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
CALENA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CALENA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CALENA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CALENA							

Bits	Description	
[31:0]	**CALENA**	**Interrupt Clear Enable Bit** The NVIC_ICER0-NVIC_ICER3 registers disable interrupts, and show which interrupts are enabled. Write Operation: 0 = No effect. 1 = Interrupt Disabled. Read Operation: 0 = Interrupt Disabled. 1 = Interrupt Enabled.

IRQ0 ~ IRQ108 Clear-enable Control Register (NVIC_ICER2)

Register	Offset	R/W	Description	Reset Value
NVIC_ICER2	NVIC_BA+0x88	R/W	IRQ0 ~ IRQ108 Clear-enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
CALENA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CALENA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CALENA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CALENA							

Bits	Description	
[31:0]	**CALENA**	**Interrupt Clear Enable Bit** The NVIC_ICER0-NVIC_ICER3 registers disable interrupts, and show which interrupts are enabled. Write Operation: 0 = No effect. 1 = Interrupt Disabled. Read Operation: 0 = Interrupt Disabled. 1 = Interrupt Enabled.

IRQ0 ~ IRQ108 Clear-enable Control Register (NVIC_ICER3)

Register	Offset	R/W	Description	Reset Value
NVIC_ICER3	NVIC_BA+0x8C	R/W	IRQ0 ~ IRQ108 Clear-enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
CALENA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CALENA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CALENA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CALENA							

Bits	Description	
[31:0]	**CALENA**	**Interrupt Clear Enable Bit** The NVIC_ICER0-NVIC_ICER3 registers disable interrupts, and show which interrupts are enabled. Write Operation: 0 = No effect. 1 = Interrupt Disabled. Read Operation: 0 = Interrupt Disabled. 1 = Interrupt Enabled.

IRQ0 ~ IRQ108 Set-pending Control Register (NVIC_ISPR0)

Register	Offset	R/W	Description	Reset Value
NVIC_ISPR0	NVIC_BA+0x100	R/W	IRQ0 ~ IRQ108 Set-pending Control Register	0x0000_0000

31	30	29	28	27	26	25	24
SETPEND							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SETPEND							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SETPEND							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SETPEND							

Bits	Description	
[31:0]	**SETPEND**	**Interrupt Set-pending** The NVIC_ISPR0-NVIC_ISPR3 registers force interrupts into the pending state, and show which interrupts are pending Write Operation: 0 = No effect. 1 = Changes interrupt state to pending. Read Operation: 0 = Interrupt is not pending. 1 = Interrupt is pending.

IRQ0 ~ IRQ108 Set-pending Control Register (NVIC_ISPR1)

Register	Offset	R/W	Description	Reset Value
NVIC_ISPR1	NVIC_BA+0x104	R/W	IRQ0 ~ IRQ108 Set-pending Control Register	0x0000_0000

31	30	29	28	27	26	25	24
SETPEND							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SETPEND							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SETPEND							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SETPEND							

Bits	Description	
[31:0]	**SETPEND**	**Interrupt Set-pending** The NVIC_ISPR0-NVIC_ISPR3 registers force interrupts into the pending state, and show which interrupts are pending Write Operation: 0 = No effect. 1 = Changes interrupt state to pending. Read Operation: 0 = Interrupt is not pending. 1 = Interrupt is pending.

IRQ0 ~ IRQ108 Set-pending Control Register (NVIC_ISPR2)

Register	Offset	R/W	Description	Reset Value
NVIC_ISPR2	NVIC_BA+0x108	R/W	IRQ0 ~ IRQ108 Set-pending Control Register	0x0000_0000

31	30	29	28	27	26	25	24
SETPEND							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SETPEND							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SETPEND							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SETPEND							

Bits	Description	
[31:0]	**SETPEND**	**Interrupt Set-pending** The NVIC_ISPR0-NVIC_ISPR3 registers force interrupts into the pending state, and show which interrupts are pending Write Operation: 0 = No effect. 1 = Changes interrupt state to pending. Read Operation: 0 = Interrupt is not pending. 1 = Interrupt is pending.

IRQ0 ~ IRQ108 Set-pending Control Register (NVIC_ISPR3)

Register	Offset	R/W	Description	Reset Value
NVIC_ISPR3	NVIC_BA+0x10C	R/W	IRQ0 ~ IRQ108 Set-pending Control Register	0x0000_0000

31	30	29	28	27	26	25	24
SETPEND							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SETPEND							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SETPEND							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SETPEND							

Bits	Description	
[31:0]	**SETPEND**	**Interrupt Set-pending** The NVIC_ISPR0-NVIC_ISPR3 registers force interrupts into the pending state, and show which interrupts are pending Write Operation: 0 = No effect. 1 = Changes interrupt state to pending. Read Operation: 0 = Interrupt is not pending. 1 = Interrupt is pending.

IRQ0 ~ IRQ108 Clear-pending Control Register (NVIC_ICPR0)

Register	Offset	R/W	Description	Reset Value
NVIC_ICPR0	NVIC_BA+0x180	R/W	IRQ0 ~ IRQ108 Clear-pending Control Register	0x0000_0000

31	30	29	28	27	26	25	24
CALPEND							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CALPEND							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CALPEND							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CALPEND							

Bits	Description	
[31:0]	**CALPEND**	**Interrupt Clear-pending** The NVIC_ICPR0-NVIC_ICPR3 registers remove the pending state from interrupts, and show which interrupts are pending Write Operation: 0 = No effect. 1 = Removes pending state an interrupt. Read Operation: 0 = Interrupt is not pending. 1 = Interrupt is pending.

IRQ0 ~ IRQ108 Clear-pending Control Register (NVIC_ICPR1)

Register	Offset	R/W	Description	Reset Value
NVIC_ICPR1	NVIC_BA+0x184	R/W	IRQ0 ~ IRQ108 Clear-pending Control Register	0x0000_0000

31	30	29	28	27	26	25	24
CALPEND							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CALPEND							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CALPEND							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CALPEND							

Bits	Description	
[31:0]	**CALPEND**	**Interrupt Clear-pending** The NVIC_ICPR0-NVIC_ICPR3 registers remove the pending state from interrupts, and show which interrupts are pending Write Operation: 0 = No effect. 1 = Removes pending state an interrupt. Read Operation: 0 = Interrupt is not pending. 1 = Interrupt is pending.

IRQ0 ~ IRQ108 Clear-pending Control Register (NVIC_ICPR2)

Register	Offset	R/W	Description	Reset Value
NVIC_ICPR2	NVIC_BA+0x188	R/W	IRQ0 ~ IRQ108 Clear-pending Control Register	0x0000_0000

31	30	29	28	27	26	25	24
CALPEND							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CALPEND							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CALPEND							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CALPEND							

Bits	Description	
[31:0]	**CALPEND**	**Interrupt Clear-pending** The NVIC_ICPR0-NVIC_ICPR3 registers remove the pending state from interrupts, and show which interrupts are pending Write Operation: 0 = No effect. 1 = Removes pending state an interrupt. Read Operation: 0 = Interrupt is not pending. 1 = Interrupt is pending.

IRQ0 ~ IRQ108 Clear-pending Control Register (NVIC_ICPR3)

Register	Offset	R/W	Description	Reset Value
NVIC_ICPR3	NVIC_BA+0x18C	R/W	IRQ0 ~ IRQ108 Clear-pending Control Register	0x0000_0000

31	30	29	28	27	26	25	24
CALPEND							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CALPEND							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CALPEND							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CALPEND							

Bits	Description	
[31:0]	**CALPEND**	**Interrupt Clear-pending** The NVIC_ICPR0-NVIC_ICPR3 registers remove the pending state from interrupts, and show which interrupts are pending Write Operation: 0 = No effect. 1 = Removes pending state an interrupt. Read Operation: 0 = Interrupt is not pending. 1 = Interrupt is pending.

IRQ0 ~ IRQ108 Active Bit Register (NVIC_IABR0)

Register	Offset	R/W	Description	Reset Value
NVIC_IABR0	NVIC_BA+0x200	R/W	IRQ0 ~ IRQ108 Active Bit Register	0x0000_0000

31	30	29	28	27	26	25	24
ACTIVE							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ACTIVE							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ACTIVE							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ACTIVE							

Bits	Description	
[31:0]	**ACTIVE**	**Interrupt Active Flags** The NVIC_IABR0-NVIC_IABR3 registers indicate which interrupts are active. 0 = interrupt not active. 1 = interrupt active.

IRQ0 ~ IRQ108 Active Bit Register (NVIC_IABR1)

Register	Offset	R/W	Description	Reset Value
NVIC_IABR1	NVIC_BA+0x204	R/W	IRQ0 ~ IRQ108 Active Bit Register	0x0000_0000

31	30	29	28	27	26	25	24
ACTIVE							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ACTIVE							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ACTIVE							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ACTIVE							

Bits	Description	
[31:0]	**ACTIVE**	**Interrupt Active Flags** The NVIC_IABR0-NVIC_IABR3 registers indicate which interrupts are active. 0 = interrupt not active. 1 = interrupt active.

IRQ0 ~ IRQ108 Active Bit Register (NVIC_IABR2)

Register	Offset	R/W	Description	Reset Value
NVIC_IABR2	NVIC_BA+0x208	R/W	IRQ0 ~ IRQ108 Active Bit Register	0x0000_0000

31	30	29	28	27	26	25	24
ACTIVE							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ACTIVE							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ACTIVE							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ACTIVE							

Bits	Description	
[31:0]	**ACTIVE**	**Interrupt Active Flags** The NVIC_IABR0-NVIC_IABR3 registers indicate which interrupts are active. 0 = interrupt not active. 1 = interrupt active.

IRQ0 ~ IRQ108 Active Bit Register (NVIC_IABR3)

Register	Offset	R/W	Description	Reset Value
NVIC_IABR3	NVIC_BA+0x20C	R/W	IRQ0 ~ IRQ108 Active Bit Register	0x0000_0000

31	30	29	28	27	26	25	24
ACTIVE							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ACTIVE							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ACTIVE							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ACTIVE							

Bits	Description	
[31:0]	**ACTIVE**	**Interrupt Active Flags** The NVIC_IABR0-NVIC_IABR3 registers indicate which interrupts are active. 0 = interrupt not active. 1 = interrupt active.

IRQ0 ~ IRQ71 Interrupt Priority Register (NVIC_IPRn)

Register	Offset	R/W	Description	Reset Value
NVIC_IPRn n=0,1..28	NVIC_BA+0x300 +0x4*n	R/W	IRQ0 ~ IRQ71 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_4n_3				Reserved			
23	22	21	20	19	18	17	16
PRI_4n_2				Reserved			
15	14	13	12	11	10	9	8
PRI_4n_1				Reserved			
7	6	5	4	3	2	1	0
PRI_4n_0				Reserved			

Bits	Description	
[31:28]	**PRI_4n_3**	**Priority of IRQ_4n+3** “0” denotes the highest priority and “15” denotes the lowest priority
[27:24]	**Reserved**	Reserved.
[23:20]	**PRI_4n_2**	**Priority of IRQ_4n+2** “0” denotes the highest priority and “15” denotes the lowest priority
[19:16]	**Reserved**	Reserved.
[15:12]	**PRI_4n_1**	**Priority of IRQ_4n+1** “0” denotes the highest priority and “15” denotes the lowest priority
[11:8]	**Reserved**	Reserved.
[7:4]	**PRI_4n_0**	**Priority of IRQ_4n+0** “0” denotes the highest priority and “15” denotes the lowest priority
[3:0]	**Reserved**	Reserved.

Software Trigger Interrupt Register (STIR)

Register	Offset	R/W	Description	Reset Value
STIR	NVIC_BA+0xE00	R/W	Software Trigger Interrupt Registers	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**INTID**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INTID							

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8:0]	**INTID**	**Interrupt ID** Write to the STIR To Generate An Interrupt from Software When the USERSETMPEND bit in the SCR is set to 1, unprivileged software can access the STIR Interrupt ID of the interrupt to trigger, in the range 0-63. For example, a value of 0x03 specifies interrupt IRQ3.

6.2.6 System Control Register

The Cortex-M4 status and operation mode control are managed by System Control Registers. The CPUID, Cortex-M4 interrupt priority and Cortex-M4 power management can be controlled through these system control registers.

For more detailed information, please refer to the "*Arm Cortex-M4 Technical Reference Manual" and "Arm v7-M Architecture Reference Manual*".

6.2.6.1 Register Map

R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value
SCR Base Address: **SCS_BA = 0xE000_E000**				
ICSR	SCS_BA+0xD04	R/W	Interrupt Control and State Register	0x0000_0000
AIRCR	SCS_BA+0xD0C	R/W	Application Interrupt and Reset Control Register	0xFA05_0000
SCR	SCS_BA+0xD10	R/W	System Control Register	0x0000_0000
SHPR1	SCS_BA+0xD18	R/W	System Handler Priority Register 1	0x0000_0000
SHPR2	SCS_BA+0xD1C	R/W	System Handler Priority Register 2	0x0000_0000
SHPR3	SCS_BA+0xD20	R/W	System Handler Priority Register 3	0x0000_0000

6.2.6.2 Register Description

Interrupt Control State Register (ICSR)

Register	Offset	R/W	Description	Reset Value
ICSR	SCS_BA+0xD04	R/W	Interrupt Control and State Register	0x0000_0000

31	30	29	28	27	26	25	24
NMIPENDSET	Reserved		PENDSVSET	PENDSVRTC_CAL	PENDSTSET	PENDSTRTC_CAL	Reserved
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ISRPREEMPT	ISRPENDING	Reserved				VECTPENDING	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
VECTPENDING				RETTOBASE	Reserved		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	VECTACTIVE						

Bits	Description	
[31]	**NMIPENDSET**	**NMI Set-pending Bit** Write Operation: 0 = No effect. 1 = Change NMI exception state to pending. Read Operation: 0 = NMI exception is not pending. 1 = NMI exception is pending. **Note:** Because NMI is the highest-priority exception, normally the processor enters the NMI exception handler as soon as it detects a write of 1 to this bit. Entering the handler then clears this bit to 0. This means a read of this bit by the NMI exception handler returns 1 only if the NMI signal is reasserted while the processor is executing that handler.
[30:29]	**Reserved**	Reserved.
[28]	**PENDSVSET**	**PendSV Set-pending Bit** Write Operation: 0 = No effect. 1 = Change PendSV exception state to pending. Read Operation: 0 = PendSV exception is not pending. 1 = PendSV exception is pending. **Note:** Writing 1 to this bit is the only way to set the PendSV exception state to pending.
[27]	**PENDSVRTC_CAL**	**PendSV Clear-pending Bit** Write Operation: 0 = No effect. 1 = Remove the pending state from the PendSV exception. **Note:** This is a write only bit. To clear the PENDSV bit, you must “write 0 to PENDSVSET and write 1 to PENDSVRTC_CAL” at the same time.

[26]	**PENDSTSET**	**SysTick Exception Set-pending Bit** Write Operation: 0 = No effect. 1 = Change SysTick exception state to pending. Read Operation: 0 = SysTick exception is not pending. 1 = SysTick exception is pending.
[25]	**PENDSTRTC_CAL**	**SysTick Exception Clear-pending Bit** Write Operation: 0 = No effect. 1 = Remove the pending state from the SysTick exception. **Note:** This is a write only bit. To clear the PENDST bit, you must "write 0 to PENDSTSET and write 1 to PENDSTRTC_CAL" at the same time.
[24]	**Reserved**	Reserved.
[23]	**ISRPREEMPT**	**Interrupt Preempt Bit (Read Only)** If set, a pending exception will be serviced on exit from the debug halt state.
[22]	**ISRPENDING**	**Interrupt Pending Flag, Excluding NMI and Faults (Read Only)** 0 = Interrupt not pending. 1 = Interrupt pending.
[21:18]	**Reserved**	Reserved.
[17:12]	**VECTPENDING**	**Number of the Highest Pended Exception** Indicate the Exception Number of the Highest Priority Pending Enabled Exception 0 = No pending exceptions. Nonzero = the exception number of the highest priority pending enabled exception. The value indicated by this field includes the effect of the BASEPRI and FAULTMASK registers, but not any effect of the PRIMASK register.
[11]	**RETTOBASE**	**Preempted Active Exceptions Indicator** Indicate whether There are Preempted Active Exceptions 0 = there are preempted active exceptions to execute. 1 = there are no active exceptions, or the currently-executing exception is the only active exception.
[10:7]	**Reserved**	Reserved.
[6:0]	**VECTACTIVE**	**Number of the Current Active Exception** 0 = Thread mode. Non-zero = The exception number of the currently active exception.

Application Interrupt and Reset Control Register (AIRCR)

Register	Offset	R/W	Description	Reset Value
AIRCR	SCS_BA+0xD0C	R/W	Application Interrupt and Reset Control Register	0xFA05_0000

31	30	29	28	27	26	25	24
VECTORKEY							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
VECTORKEY							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ENDIANNESS	Reserved				PRIGROUP		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					SYSRESETREQ	VECTCLRACTIVE	VECTRESET

Bits	Description	
[31:16]	**VECTORKEY**	**Register Access Key** When writing this register, this field should be 0x05FA, otherwise the write action will be unpredictable. The VECTORKEY filed is used to prevent accidental write to this register from resetting the system or clearing of the exception status.
[15]	**ENDIANNESS**	**Data Endianness** 0 = Little-endian. 1 = Big-endian.
[14:11]	**Reserved**	Reserved.
[10:8]	**PRIGROUP**	**Interrupt Priority Grouping** This field determines the Split Of Group priority from subpriority,
[7:3]	**Reserved**	Reserved.
[2]	**SYSRESETREQ**	**System Reset Request** Writing This Bit to 1 Will Cause A Reset Signal To Be Asserted To The Chip And Indicate A Reset Is Requested This bit is write only and self-cleared as part of the reset sequence.
[1]	**VECTCLRACTIVE**	**Exception Active Status Clear Bit** Setting This Bit To 1 Will Clears All Active State Information For Fixed And Configurable Exceptions This bit is write only and can only be written when the core is halted. **Note:** It is the debugger's responsibility to re-initialize the stack.
[0]	**VECTRESET**	Reserved.

PRIGROUP	Binary Point	Group Priority Bits	Subpriority Bits	Number Of Group Priorities	Subpriorities
0b000	bxxxxxxx.y	[7:1]	[0]	128	2
0b001	bxxxxxx.yy	[7:2]	[1:0]	64	4
0b010	bxxxxx.yyy	[7:3]	[2:0]	32	8
0b011	bxxxx.yyyy	[7:4]	[3;0]	16	16
0b100	bxxx.yyyyy	[7:5]	[4:0]	8	32
0b101	bxx.yyyyyy	[7:6]	[5:0]	4	64
0b110	bx.yyyyyyy	[7]	[6:0]	2	128
0b111	b.yyyyyyyy	None	[7:0]	1	256

Table 6.2-3 Priority Grouping

System Control Register (SCR)

Register	Offset	R/W	Description	Reset Value
SCR	SCS_BA+0xD10	R/W	System Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			SEVONPEND	Reserved	SLEEPDEEP	SLEEPONEXIT	Reserved

Bits	Description	
[31:5]	**Reserved**	Reserved.
[4]	**SEVONPEND**	**Send Event on Pending** 0 = Only enabled interrupts or events can wake up the processor, while disabled interrupts are excluded. 1 = Enabled events and all interrupts, including disabled interrupts, can wake up the processor. When an event or interrupt enters pending state, the event signal wakes up the processor from WFE. If the processor is not waiting for an event, the event is registered and affects the next WFE. The processor also wakes up on execution of an SEV instruction or an external event.
[3]	**Reserved**	Reserved.
[2]	**SLEEPDEEP**	**Processor Deep Sleep and Sleep Mode Selection** Control whether the Processor Uses Sleep Or Deep Sleep as its Low Power Mode. 0 = Sleep. 1 = Deep sleep.
[1]	**SLEEPONEXIT**	**Sleep-on-exit Enable Control** This bit indicate Sleep-On-Exit when Returning from Handler Mode to Thread Mode. 0 = Do not sleep when returning to Thread mode. 1 = Enter sleep, or deep sleep, on return from an ISR to Thread mode. Setting this bit to 1 enables an interrupt driven application to avoid returning to an empty main application.
[0]	**Reserved**	Reserved.

System Handler Priority Register 1 (SHPR1)

Register	Offset	R/W	Description	Reset Value
SHPR1	SCS_BA+0xD18	R/W	System Handler Priority Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PRI_6				**Reserved**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PRI_5				**Reserved**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRI_4				**Reserved**			

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:20]	**PRI_6**	Priority of system handler 6, UsageFault
[20:16]	**Reserved**	Reserved.
[15:12]	**PRI_5**	Priority of system handler 5, BusFault
[11:8]	**Reserved**	Reserved.
[7:4]	**PRI_4**	Priority of system handler 4, MemManage
[3:0]	**Reserved**	Reserved.

System Handler Priority Register 2 (SHPR2)

Register	Offset	R/W	Description	Reset Value
SHPR2	SCS_BA+0xD1C	R/W	System Handler Priority Register 2	0x0000_0000

31	30	29	28	27	26	25	24
PRI_11				Reserved			
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							

Bits	Description	
[31:28]	**PRI_11**	**Priority of System Handler 11 – SVCall** “0” denotes the highest priority and “3” denotes the lowest priority.
[27:0]	**Reserved**	Reserved.

System Handler Priority Register 3 (SHPR3)

Register	Offset	R/W	Description	Reset Value
SHPR3	SCS_BA+0xD20	R/W	System Handler Priority Register 3	0x0000_0000

31	30	29	28	27	26	25	24
PRI_15				**Reserved**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PRI_14				**Reserved**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:28]	**PRI_15**	**Priority of System Handler 15 – SysTick** "0" denotes the highest priority and "3" denotes the lowest priority.
[27:24]	**Reserved**	Reserved.
[23:20]	**PRI_14**	**Priority of System Handler 14 – PendSV** "0" denotes the highest priority and "3" denotes the lowest priority.
[19:0]	**Reserved**	Reserved.

6.3 Arm TrustZone for Armv8-A

6.3.1 Overview

Arm TrustZone can be considered as a physical partition that divides the memory system into two worlds: **Secure** world and **Non-secure** world. Each of the world has its resources, such as processor, SRAM, DRAM, GPIO and other peripherals. The Non-secure world, in general, is a rich execution environment (REE) executing complex and various software applications. The Secure world is an isolated execution environment, in which resources are protected and cannot be accessed directly by master in Non-secure world. It is typically used to provide trusted execution environment (TEE) to run trusted software.

A TrustZone-aware PE (processing element), such as Cortex-A35 processor core, can behave as a Secure master or Non-secure master depending on its security state. When the processor is in Secure state, it is a Secure master and can access resources in both Secure world and Non-secure world; while the processor is in Non-secure state, it is a Non-secure master and can access resources in Non-secure world only.

6.3.2 TrustZone of Cortex-A35 Processor

Cortex-A35 is a processor implementing Armv8-A architecture. As depicted in Table 6.3-1, the Cortex-A35 has four exception levels (EL) and two security states. Lower exception level indicates lower privilege. That is, EL3 has the highest permission and can access all resources in the system, while EL0 can only access restricted resources that is managed by EL1, EL2 or EL3. The security state decides whether the PE is currently executed as Secure or Non-secure.

Following TrustZone architecture, Cortex-A35 PE has Secure EL0, EL1 and EL3 (denoted to S.EL0, S.EL1 and S.EL3) and Non-secure EL0, EL1, EL2 (denoted to NS.EL0, NS.EL1 and NS.EL2). Each of them is designed for different function of software. For example, the Linux OS is usually executed at NS.EL1, and user application is executed at NS.EL0. Thus, Linux OS is able to manage the resource for all user applications.

When the PE runs at EL3, it is always in Secure state, denoted as S.EL3. When the PE runs in other exception level, it can be in Secure state or Non-secure state. The security state of the PE is set by SCR_EL3.NS bit, which can be set by S.EL3 software only. In other words, the security state of PE can be changed by S.EL3 software only.

Security State / **Exception Level**	**Secure**		**Non-Secure**	
EL0	S.EL0	Trusted App.	NS.EL0	App.
EL1	S.EL1	Trusted OS (e.g. OPTEE)	NS.EL1	Rich OS (e.g. Linux)
EL2	S.EL2	-	NS.EL2	Hypervisor
EL3	S.EL3	Secure monitor	-	-

Table 6.3-1 Cortex-A35 AArch64 Exception Levels vs. Security States and Typical Function of Each State

6.3.2.1 Address Space Partition of Armv8-A

Armv8-A memory system is Virtual Memory System Architecture, VMSA, which means the memory system of Cortex-A35 processor has virtual address (VA) and physical address (PA) and uses memory management unit (MMU) to map VA to PA.

When MMU is enabled, software executed by Cortex-A35 uses VA to access the memory system and

MMU uses translation table to map the VA to PA; When MMU is disabled, software uses PA to access the system directly. TrustZone separates the physical memory to Secure space and Non-secure space. The VA of Secure software can be translated to Secure or Non-secure physical address space, while the VA of Non-secure software is always translated to Non-secure physical address space. Thus, Non-secure software can only see Non-secure resources, but can never see Secure resources.

To learn more about TrustZone of Armv8-A, refer to the section 3 “TrustZone in the processor” of the Arm document “TrustZone for Armv8-A”, ARM062-1010708621-28, for more details.

6.3.3 TrustZone System Architecture (SSMCC, SSPCC)

The TrustZone architecture of Cortex-A35 makes the software to view the memory system as Secure world and Non-secure world. However, the TrustZone physical partition is not just all about that. It requires support of the system. As shown in Figure 6.3-1 the system of this chip, including system bus, bus masters and bus slaves, supports TrustZone.

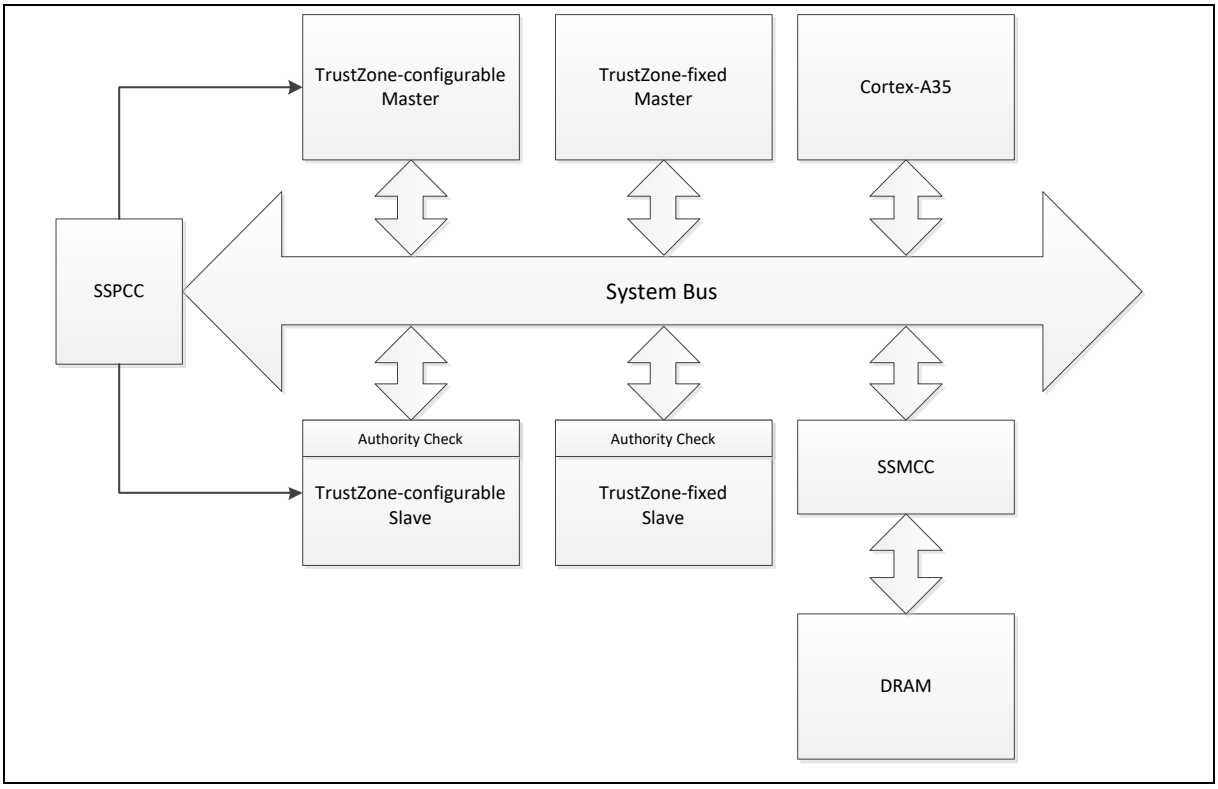


Figure 6.3-1 TrustZone System Architecture Block Diagram

The system bus takes charge of transferring information between masters and slaves. For each transaction, the system bus transfers the information indicating the security state of the bus master to bus slave, so that the bus slave can do the authority check based on the information.

6.3.3.1 Security Attribute Configuration

Bus masters include processors and master peripherals, such as PDMA and GMAC. Except Cortex-A35 processor, whose security state depends on the current exception level and the value of SCR_EL3.NS bit, masters’ security state can be configured by SSPCC (TrustZone-configurable) or be fixed (TrustZone-fixed) to either Secure or Non-secure. For example, PDMA0 can be configured to Secure or Non-secure by SSPCC while GMAC is fixed to Non-secure. Refer to SSPCC section for more details.

A Secure master can access Secure slave and Non-secure slave, while a Non-secure master can only access Non-secure slave. If the Non-secure master tries to access a Secure slave, it will not pass the

authority check and will get error response from the slave.

Most of the peripherals in the system are bus slaves, which, similar to bus masters, are also TrustZone-configurable or TrustZone-fixed. Secure software can set the security state of TrustZone-configurable slaves through SSPCC.

The normal memory in the system is also TrustZone-configurable:

- SRAM can be partitioned to Secure or Non-secure regions by setting the boundary through SSPCC.
- DRAM can be partitioned to Secure or Non-secure regions through SSMCC.

Refer to SSPCC and SSMCC section for more details.

6.3.3.2 Authority Check and Violation Report

For each transaction, the bus slave will perform the authority check based on the memory access policy, which is:

- Non-secure slave accepts access from both Secure and Non-secure master.
- Secure slave accepts access from Secure master.
- Secure slave will report the violation access event and reply error response when receiving the access from Non-secure master.

The SSPCC receives violation access event from Secure slaves and further generates interrupt to notify the Cortex-A35 if interrupt is enabled.

Unlike other bus slaves, the authority check of DRAM is performed by SSMCC. All transactions to access DRAM will be checked by SSMCC based on the memory access policy. The Secure software should configure SSMCC properly before using DRAM.

6.3.3.3 Partition Beyond TrustZone

Apart from TrustZone architecture, this chip has a scale-configurable subsystem for real time processing, called subM system, which is an operating environment for the real-time processor: Cortex-M4. Similar to TrustZone architecture of this chip, some resources are fixed to that subsystem and some resources are configurable. Refer to SSPCC section for more details.

6.4 System Manager

6.4.1 Overview

The system management describes following information and functions.

- System Resets
- System Power Architecture
- System Memory Map
- System Control Registers for Product Identifier (PDID), Power-On Setting, Reset Control for on-chip controllers/peripherals, pin multi-function control and miscellaneous function control.

6.4.2 System Reset

The system reset can be issued by one of the following listed events. These reset event flags can be read from SYS_RSTSTS register to determine the reset source. Hardware reset sources are from peripheral signals while software reset can trigger reset through setting control registers.

- Hardware Reset Sources to reset chip
  - Power-On Reset
  - Low Voltage Reset (LVR)
  - Low level on the nRESET pin
  - Watchdog Time-out Reset and Window Watchdog Reset (WDT/WWDT Reset)
- Hardware Reset Sources to reset RTP Cortex-M4 core
  - RTP Cortex-M4 CPU Lockup Reset
- Software Reset Sources to reset chip
  - CHIP Reset to reset whole chip by writing 1 to CHIPRST (SYS_IPRST0[0])
- Software Reset Sources to reset Cortex-A35 core
  - CPU Reset for Cortex-A35 core 0 only by writing 1 to CA35CR0RST (SYS_IPRST0[1])
  - Warm Reset request for Cortex-A35 core 0 only by writing 1 to the bit RR (RMR[1])
  - CPU Reset for Cortex-A35 core 1 only by writing 1 to CA35CR1RST (SYS_IPRST0[2])
  - Warm Reset request for Cortex-A35 core 1 only by writing 1 to the bit RR (RMR[1])
- Software Reset Sources to reset RTP Cortex-M4 core
  - CPU Reset for RTP Cortex-M4 core only by writing 1 to CM4RST (SYS_IPRST0[3])
  - CPU Reset for RTP Cortex-M4 core only by writing 1 to the bit SYSRESETREQ (AIRCR[2])

6.4.2.1 Power-On Reset (POR)

The Power-on reset (POR) is used to generate a stable system reset signal and forces the system to be reset when power-on to avoid unexpected behavior of MPU. When applying the power to MPU, the POR module detects the rising voltage and generates reset signal to system until the voltage is ready for MPU operation. At POR reset, the PORF (SYS_RSTSTS[0]) will be set to 1 to indicate there is a POR reset event. The PORF (SYS_RSTSTS[0]) bit can be cleared by writing 1 to it. Figure 6.4-1 shows the power-on reset waveform.

Please note that POR powered by AV_{DD} in BGA312 and BGA364 package type while powered by V_{DDIO0} in LQFP216-EP package type.

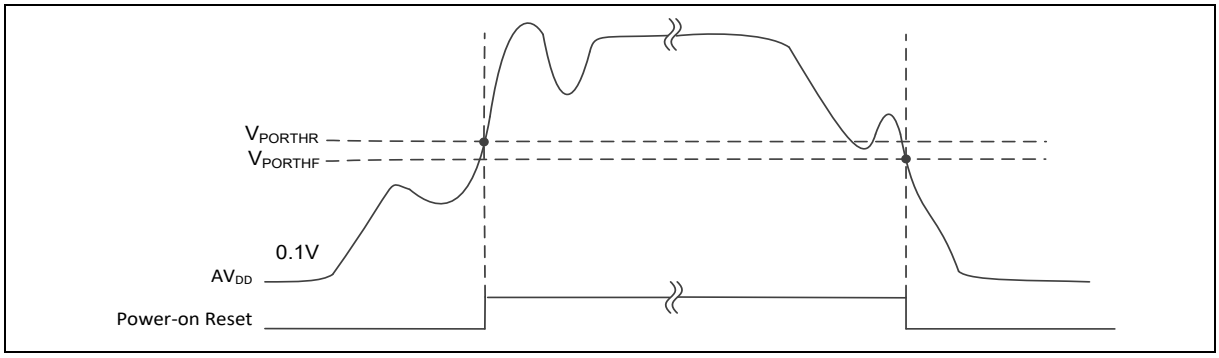


Figure 6.4-1 Power-on Reset (POR) Waveform

6.4.2.2 Low Voltage Reset (LVR)

If the Low Voltage Reset function is enabled by setting the Low Voltage Reset Enable Bit LVREN (SYS_LVRDCR[0]) to 1, after 200us delay, LVR detection circuit will be stable and the LVR function will be active. Then LVR function will detect AV_{DD} during system operation. When the AV_{DD} voltage is lower than V_{LVR} and the state keeps longer than De-glitch time set by LVRDGSEL (SYS_LVRDCR[3:1]), chip will be reset. The LVR reset will control the chip in reset state until the AV_{DD} voltage rises above V_{LVR} and the state keeps longer than De-glitch time set by LVRDGSEL (SYS_LVRDCR[3:1]). The default setting of Low Voltage Reset is enabled without De-glitch function. Figure 6.4-2 shows the Low Voltage Reset waveform.

Please note that LVR powered by AV_{DD} in BGA312 and BGA364 package type while powered by V_{DDIO0} in LQFP216-EP package type.

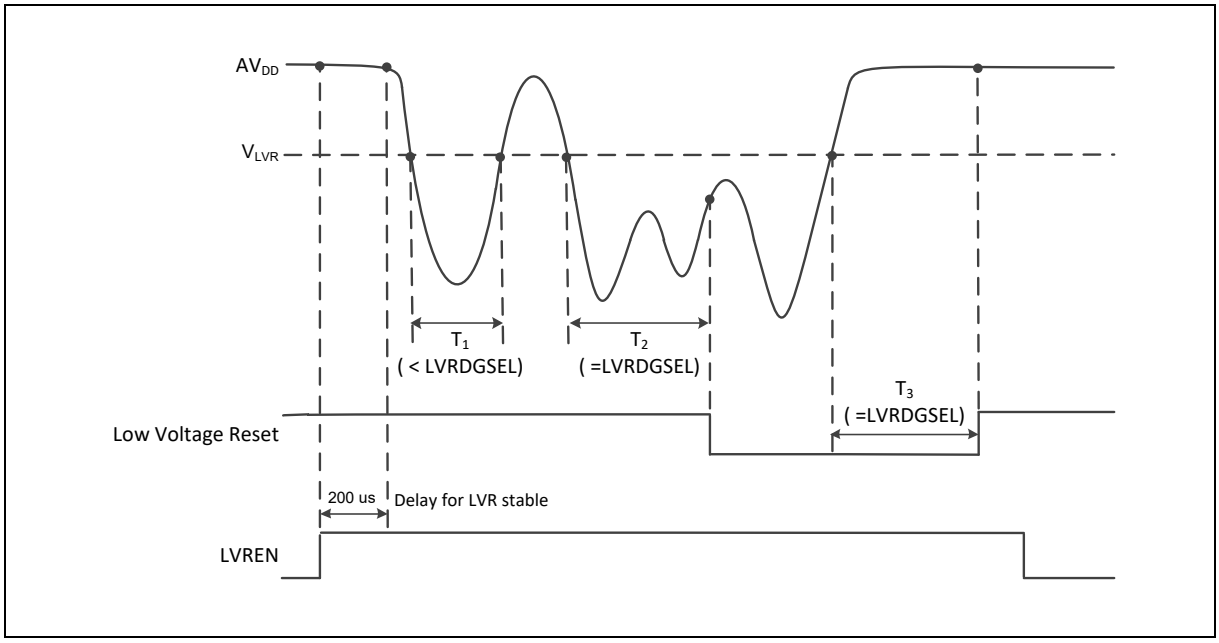


Figure 6.4-2 Low Voltage Reset (LVR) Waveform

6.4.2.3 Pin nRESET Reset

The nRESET reset means to generate a reset signal by pulling nRESET pin low, which is an asynchronous reset input pin and can be used to reset system at any time. When the nRESET voltage is lower than 0.2 V_{DDIO0} and the state keeps longer than 100 us (glitch filter), chip will be reset. The nRESET reset will control the chip in reset state until the nRESET voltage rises above 0.7 V_{DDIO0} and the state keeps longer than 100 us (glitch filter). The PINRF (SYS_RSTSTS[1]) will be set to 1 if the

previous reset source is nRESET reset. Figure 6.4-3 shows the nRESET reset waveform.

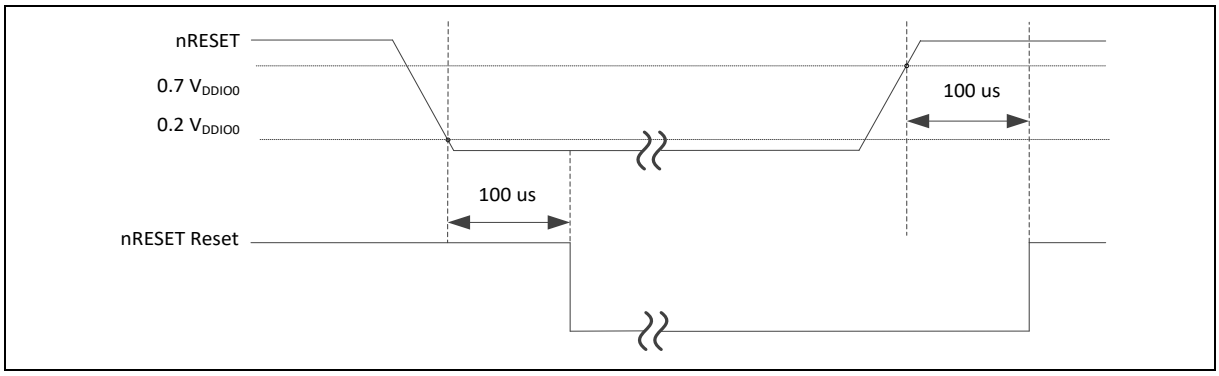


Figure 6.4-3 nRESET Reset Waveform

6.4.2.4 Watchdog Timer Reset (WDT)

In most industrial applications, system reliability is quite important. Recovering the system from failure status automatically is one way to improve system reliability. The watchdog timer (WDT) is widely used to check if the system works fine. If the system is crashed or out of control, it may cause the WDT time-out. User may decide to enable system reset during WDT time-out to recover the system and take action for the system crash/out-of-control after reset.

The MA35D1 series is equipped with three WDTs: WDT0, WDT1 and WDT2. The WDT0 is for TrustZone Secure (TZS), WDT1 is for TrustZone Secure/Non-Secure (TZS/TZNS) and WDT2 is for SubM.

The WDT0 time-out resets system automatically.

The WDT1 time-out resets Cortex-A35 core and controllers/peripherals configured in TZS/TZNS only when WDT1RSTAEN (SYS_MISCRFCR[16]) is 1. The WDT1 time-out resets Cortex-M4 core and peripherals configured in SubM only when WDT1RSTAEN (SYS_MISCRFCR[16]) and WDT1RSTMEN (SYS_MISCRFCR[18]) are both 1.

The WDT2 time-out resets Cortex-A35 core and controllers/peripherals configured in TZS/TZNS only when WDT2RSTAEN (SYS_MISCRFCR[17]) is 1. The WDT2 time-out resets Cortex-M4 core and peripherals configured in SubM automatically.

Software can check if the reset is caused by time-out of WDT0, WDT1 or WDT2 to indicate the previous reset is a WDT0, WDT1 or WDT2 reset and handle the failure of system after WDT0, WDT1 or WDT2 time-out reset by checking WDT0RF (SYS_RSTSTS[2]), WDT1RF (SYS_RSTSTS[10]), WDT2RFA (SYS_RSTSTS[11]), WDT1RFM (SYS_RSTSTS[18]) and WDT2RF (SYS_RSTSTS[19]).

6.4.2.5 RTP Cortex-M4 CPU Lockup Reset

The RTP Cortex-M4 CPU enters lockup state after RTP Cortex-M4 CPU produces HardFault at HardFault handler and chip gives immediate indication of seriously errant kernel software. This is the result of the RTP Cortex-M4 CPU being locked because of an unrecoverable exception following the activation of the processor's built-in system state protection hardware. When chip enters debug mode, the RTP Cortex-M4 CPU lockup reset will be ignored.

Software can check if the reset is caused by RTP Cortex-M4 CPU entering lockup state to indicate the previous reset is a RTP Cortex-M4 CPU lockup reset by checking RTPM4LKRF (SYS_RSTSTS[20]).

6.4.2.6 CHIP Reset

The CHIP Reset is the same with Power-on Reset. The Cortex-A35 core, RTP Cortex-M4 core and controllers/peripherals are reset.

User sets CHIPRST (SYS_IPRST0[0]) to 1 to assert the CHIP Reset.

6.4.2.7 Cortex-A35 processor CPU Reset and Warm Reset

The Cortex-A35 processor CPU Reset means that only Cortex-A35 core is reset and all other controllers/peripherals remain the same status after CPU reset. User can set CA35CR0RST (SYS_IPRST0[1]) and CA35CR1RST (SYS_IPRST0[2]) to 1 to assert CPU Reset to Cortex-A35 core 0 and 1 respectively.

The Cortex-A35 processor Warm Reset requests a warm reset for Cortex-A35 core. User can set the bit RR (RMR[1]) of Cortex-A35 core 0 and 1 to assert Warm Reset to Cortex-A35 core 0 and 1 respectively.

6.4.2.8 RTP Cortex-M4 processor CPU Reset

The RTP Cortex-M4 processor CPU Reset means that only RTP Cortex-M4 core is reset and all other peripherals remain the same status after CPU reset. User can set the bit CM4RST (SYS_IPRST0[3]) or SYSRESETREQ (AIRCR[2]) to 1 to assert CPU Reset to RTP Cortex-M4 core.

6.4.3 System Power Distribution

In this chip, the power distribution is divided into several segments.

- Analog power from AV_{DD} provides 3.3V voltage to analog components operation. These analog components including POR33, LVR and Temperature Sensor.
- Analog power from AV_{DD_EADC0} provides 3.3V voltage to 12-bit 4.7 MSPS SAR-ADC.
- Analog power from AV_{DD_ADC0} provides 3.3V voltage to 12-bit 500k SPS SAR-ADC.
- Analog power from V_{DD_OTP} provides 2.5V voltage to POR25 and OTP memory.
- Power provides voltage to PLL and internal RC oscillator including:
  - AV_{DDL_PLL0} provides 1.2V voltage to CA-PLL and SYS-PLL.
  - AV_{DDL_PLL1} and AV_{DDH_PLL1} provide 1.2V and 3.3V voltage respectively to DDR-PLL.
  - V_{DD_PLL1} provides 1.2V voltage to DDR-PLL, EPLL, APLL and VPLL.
  - AV_{DDL_PLL2} and AV_{DDH_PLL2} provide 1.2V and 3.3V voltage respectively to EPLL, APLL and VPLL.
  - AV_{DDL_ROSC} provides 1.2V voltage to two 12 MHz RC oscillators (HIRC).
- Power provides voltage to a DDR32 PHY_AC macro, two DDR32 PHY_DATX8 macro and SSTL I/O including:
  - MV_{DD} provides 1.35V, 1.5V or 1.8V when SDRAM connected is DDR3L, DDR3 or DDR2 type.
  - $MV_{DD_DPHYPLL}$ provides 2.5V voltage to PLL of DDR32 PHY_AC and PHY_DATX8 macros.
  - MV_{REF_CA} provides $MV_{DD}/2$ voltage to DDR32 PHY_AC macro and address, command SSTL I/O as a reference voltage.
  - MV_{REF_DQ} provides $MV_{DD}/2$ voltage to DDR32 PHY_DATX8 macros and data SSTL I/O as a reference voltage.
- Power from V_{DD_HSUSB0} and V_{DD_HSUSB1} provide 3.3V voltage to USB 2.0 PHY 0 and USB 2.0 PHY 1 respectively while V_{DD_CORE} provides 1.25V[*1] (BGA package type) or 1.28V[*1] (LQFP package type) voltage to both USB 2.0 PHY 0 and USB 2.0 PHY 1.
- Power from V_{BAT} provides 3.3V voltage to 32.768 kHz Crystal Oscillator (LXT), 32 kHz RC oscillator (LIRC), LVR and RTC logic.
- Digital power from V_{DD_CPU} provides 1.28V[*1] voltage to dual Arm Cortex-A35 core, L1 Instruction/Data cache and L2 cache SRAM.
- Digital power from V_{DD_CORE} provides 1.25V[*1] (BGA package type) or 1.28V[*1] (LQFP package type) voltage to POR11, ROM, SRAM and all digital logic.
- Power from V_{DDIO0}, V_{DDIO1}, V_{DDIO2}, V_{DDIO3}, V_{DDIO4}, V_{DDIO5}, V_{DDIO6}, V_{DDIO7}, V_{DDIO8}, V_{DDIO9} and V_{DDIO10} provides 1.8V or 3.3V voltage to 24 MHz Crystal Oscillator (HXT) and I/O pins (PA ~ PN).

Note [*1]: For the actual input voltage ranges of V_{DD_CORE} and V_{DD_CPU}, please refer to the Minimum/Typical/Maximum values in section 8.2.1 of Datasheet document.

Figure 6.4-4 shows the power distribution of the MA35D1 series.

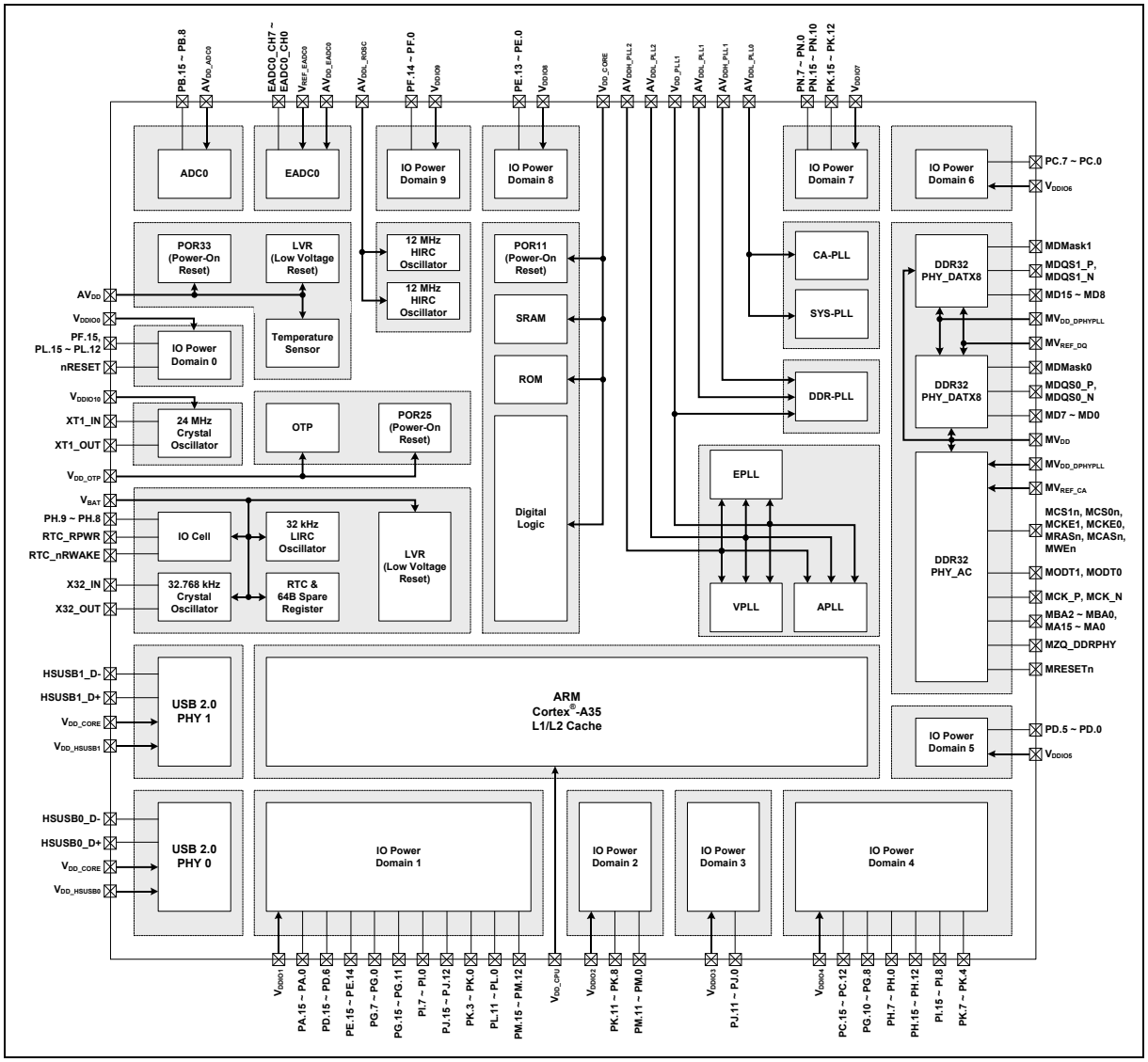


Figure 6.4-4 MA35D1 Series Power Distribution Diagram

6.4.4 Memory Organization

This chip supports only little-endian data format and provides 4G-byte addressing space. Some parts of memory space have different definitions for Cortex-A35 and RTP Cortex-M4 core. Figure 6.4-5 describes the detailed memory space definition.

The DDR SDRAM address space 0x8002_0000 ~ 0x803F_FFFF is aliased to 0x0002_0000 ~ 0x003F_FFFF in memory space view of RTP Cortex-M4 core. This address space aliasing also applies to PDMA 2 and PDMA 3.

The reserved memory space is un-accessible. Chip's behavior is undefined and unpredictable while accessing to reserved memory space.

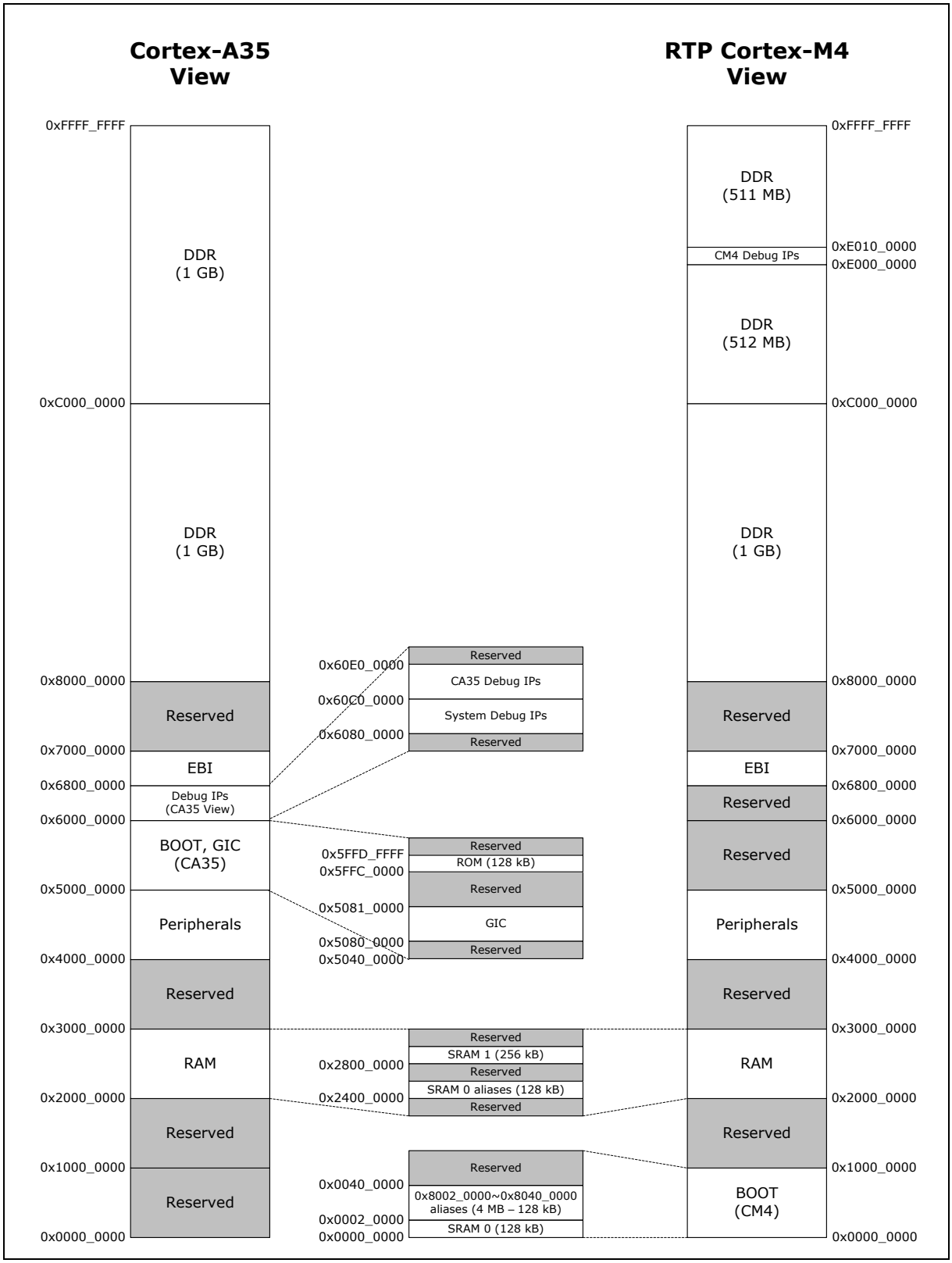


Figure 6.4-5 MA35D1 System Memory Map Diagram

The addressing space assigned to each on-chip controller or peripheral described in Table 6.4-1. The

detailed register definition, addressing space, and programming details will be described in the following sections.

Address Space	Token	Controllers
SRAM, SDRAM, External Memory and IBR (Internal Boot ROM) Space		
0x0000_0000 - 0x0001_FFFF	SRAM0_BA	SRAM 0 Memory Space (128 KB)
0x0002_0000 - 0x003F_FFFF	SDRAM4M_BA	SDRAM Address 0x8002_0000~0x8040_0000 Alias Memory Space (4 MB-128 KB)
0x2400_0000 - 0x2403_FFFF	SRAM0A_BA	SRAM 0 Alias Memory Space (256 KB)
0x2800_0000 - 0x2803_FFFF	SRAM1_BA	SRAM 1 Memory Space (256 KB)
0x5FFC_0000 - 0x5FFF_FFFF	IBR_BA	Internal Boot ROM Space (128 KB)
0x6800_0000 - 0x6FFF_FFFF	EXTMEM_BA	External Memory Space (128 MB)
0x8000_0000 - 0xFFFF_FFFF	SDRAM_BA	SDRAM Memory Space (2 GB)
AXI/AHB Peripheral Register Space		
0x4004_0000 – 0x4004_0FFF	GPIO_BA	GPIO Control Registers
0x4008_0000 – 0x4008_0FFF	PDMA0_BA	Peripheral DMA 0 Control Registers
0x4009_0000 – 0x4009_0FFF	PDMA1_BA	Peripheral DMA 1 Control Registers
0x400A_0000 – 0x400A_0FFF	PDMA2_BA	Peripheral DMA 2 Control Registers
0x400B_0000 – 0x400B_0FFF	PDMA3_BA	Peripheral DMA 3 Control Registers
0x4010_0000 – 0x4010_0FFF	EBI_BA	External Bus Interface Control Registers
0x4012_0000 – 0x4012_0FFF	GMAC0_BA	Gigabit Ethernet MAC 0 Control Registers
0x4013_0000 – 0x4013_0FFF	GMAC1_BA	Gigabit Ethernet MAC 1 Control Registers
0x4014_0000 – 0x4014_0FFF	HSUSBH0_BA	HSUSBH Host 0 Control Registers
0x4015_0000 – 0x4015_0FFF	USBH0_BA	USBH Host 0 Control Registers
0x4018_0000 – 0x4018_0FFF	SDH0_BA	SD 2.0 HOST 0 Control Registers
0x4019_0000 – 0x4019_0FFF	SDH1_BA	SD 3.0 HOST 1 Control Registers
0x401A_0000 – 0x401A_0FFF	NAND_BA	NAND Flash Memory Control Registers
0x401C_0000 – 0x401C_0FFF	HSUSBH1_BA	HSUSBH Host 1 Control Registers
0x401D_0000 – 0x401D_0FFF	USBH1_BA	USBH Host 1 Control Registers
0x4020_0000 – 0x4023_FFFF	HSUSBD_BA	HSUSBD Control Registers
0x4024_0000 – 0x4024_0FFF	CCAP0_BA	CCAP 0 Control Registers
0x4025_0000 – 0x4025_0FFF	CCAP1_BA	CCAP 1 Control Registers
0x4026_0000 – 0x4026_0FFF	DISP_BA	LCD Display Control Registers
0x4028_0000 – 0x4028_0FFF	GFX_BA	Graphic Control Registers
0x4029_0000 – 0x4029_0FFF	VDEC_BA	Video Decoder Control Registers
0x4038_0000 – 0x4038_0FFF	HWSEM0_BA	Hardware Semaphore 0 Control Registers
0x403A_0000 – 0x403A_0FFF	WRHO0_BA	Wormhole 0 Control Registers

0x403C_0000 – 0x403C_0FFF	CANFD0_BA	CAN FD 0 Control Registers
0x403D_0000 – 0x403D_0FFF	CANFD1_BA	CAN FD 1 Control Registers
0x403E_0000 – 0x403E_0FFF	CANFD2_BA	CAN FD 2 Control Registers
0x403F_0000 – 0x403F_0FFF	CANFD3_BA	CAN FD 3 Control Registers
0x503B_0000 – 0x503B_0FFF	WRHO1_BA	Wormhole 1 Control Registers
0x5080_0000 – 0x5080_7FFF	GIC_BA	Generic Interrupt Control Registers
APB Peripheral Register Space		
0x4040_0000 – 0x4040_0FFF	WDT0_BA	Watchdog Timer 0 Control Registers
0x4041_0000 – 0x4041_0FFF	RTC_BA	Real Time Clock (RTC) Control Register
0x4042_0000 – 0x4042_0FFF	ADC0_BA	Analog-Digital-Converter 0 (ADC0) Control Registers
0x4043_0000 – 0x4043_0FFF	EADC0_BA	Enhanced Analog-Digital-Converter 0 (EADC0) Control Registers
0x4044_0000 – 0x4044_0FFF	WDT1_BA	Watchdog Timer 1 Control Registers
0x4046_0000 – 0x4046_01FF	SYS_BA	System Control Registers
0x4046_0200 – 0x4046_02FF	CLK_BA	Clock Control Registers
0x4046_0300 – 0x4046_03FF	NMI_BA	NMI Control Registers
0x4048_0000 – 0x4048_0FFF	I2S0_BA	I^2S 0 Interface Control Registers
0x4049_0000 – 0x4049_0FFF	I2S1_BA	I^2S 1 Interface Control Registers
0x404A_0000 – 0x404A_0FFF	KPI_BA	KeyPad Interface Control Registers
0x404C_0000 – 0x404C_0FFF	DDRPHY_BA	DDR PHY PUB Control Registers
0x404D_0000 – 0x404D_0FFF	UMCTL2_BA	DDR (MCTL2) Control Registers
0x404E_0000 – 0x404E_0FFF	SSMCC_BA	SSMCC Control Registers
0x404F_0000 – 0x404F_0FFF	SSPCC_BA	SSPCC Control Registers
0x4050_0000 – 0x4050_0FFF	TMR01_BA	Timer 0 and Timer 1 Control Registers
0x4051_0000 – 0x4051_0FFF	TMR23_BA	Timer 2 and Timer 3 Control Registers
0x4052_0000 – 0x4052_0FFF	TMR45_BA	Timer 4 and Timer 5 Control Registers
0x4053_0000 – 0x4053_0FFF	TMR67_BA	Timer 6 and Timer 7 Control Registers
0x4054_0000 – 0x4054_0FFF	TMR89_BA	Timer 8 and Timer 9 Control Registers
0x4055_0000 – 0x4055_0FFF	TMR1011_BA	Timer 10 and Timer 11 Control Registers
0x4058_0000 – 0x4058_0FFF	EPWM0_BA	EPWM 0 Control Registers
0x4059_0000 – 0x4059_0FFF	EPWM1_BA	EPWM 1 Control Registers
0x405A_0000 – 0x405A_0FFF	EPWM2_BA	EPWM 2 Control Registers
0x4060_0000 – 0x4060_0FFF	SPI0_BA	SPI 0 Control Registers
0x4061_0000 – 0x4061_0FFF	SPI1_BA	SPI 1 Control Registers
0x4062_0000 – 0x4062_0FFF	SPI2_BA	SPI 2 Control Registers
0x4063_0000 – 0x4063_0FFF	SPI3_BA	SPI 3 Control Registers

0x4068_0000 – 0x4068_0FFF	QSPI0_BA	QSPI 0 Control Registers
0x4069_0000 – 0x4069_0FFF	QSPI1_BA	QSPI 1 Control Registers
0x4070_0000 – 0x4070_0FFF	UART0_BA	UART 0 Control Registers
0x4071_0000 – 0x4071_0FFF	UART1_BA	UART 1 Control Registers
0x4072_0000 – 0x4072_0FFF	UART2_BA	UART 2 Control Registers
0x4073_0000 – 0x4073_0FFF	UART3_BA	UART 3 Control Registers
0x4074_0000 – 0x4074_0FFF	UART4_BA	UART 4 Control Registers
0x4075_0000 – 0x4075_0FFF	UART5_BA	UART 5 Control Registers
0x4076_0000 – 0x4076_0FFF	UART6_BA	UART 6 Control Registers
0x4077_0000 – 0x4077_0FFF	UART7_BA	UART 7 Control Registers
0x4078_0000 – 0x4078_0FFF	UART8_BA	UART 8 Control Registers
0x4079_0000 – 0x4079_0FFF	UART9_BA	UART 9 Control Registers
0x407A_0000 – 0x407A_0FFF	UART10_BA	UART 10 Control Registers
0x407B_0000 – 0x407B_0FFF	UART11_BA	UART 11 Control Registers
0x407C_0000 – 0x407C_0FFF	UART12_BA	UART 12 Control Registers
0x407D_0000 – 0x407D_0FFF	UART13_BA	UART 13 Control Registers
0x407E_0000 – 0x407E_0FFF	UART14_BA	UART 14 Control Registers
0x407F_0000 – 0x407F_0FFF	UART15_BA	UART 15 Control Registers
0x4080_0000 – 0x4080_0FFF	I2C0_BA	I^2C 0 Control Registers
0x4081_0000 – 0x4081_0FFF	I2C1_BA	I^2C 1 Control Registers
0x4082_0000 – 0x4082_0FFF	I2C2_BA	I^2C 2 Control Registers
0x4083_0000 – 0x4083_0FFF	I2C3_BA	I^2C 3 Control Registers
0x4084_0000 – 0x4084_0FFF	I2C4_BA	I^2C 4 Control Registers
0x4085_0000 – 0x4085_0FFF	I2C5_BA	I^2C 5 Control Registers
0x4088_0000 – 0x4088_0FFF	UART16_BA	UART 16 Control Registers
0x4090_0000 – 0x4090_0FFF	SC0_BA	Smartcard Host 0 Control Registers
0x4091_0000 – 0x4091_0FFF	SC1_BA	Smartcard Host 1 Control Registers
0x4098_0000 – 0x4098_0FFF	WDT2_BA	Watchdog Timer 2 Control Registers
0x40B0_0000 – 0x40B0_0FFF	QEI0_BA	QEI 0 Control Registers
0x40B1_0000 – 0x40B1_0FFF	QEI1_BA	QEI 1 Control Registers
0x40B2_0000 – 0x40B2_0FFF	QEI2_BA	QEI 2 Control Registers
0x40B4_0000 – 0x40B4_0FFF	ECAP0_BA	ECAP 0 Control Registers
0x40B5_0000 – 0x40B5_0FFF	ECAP1_BA	ECAP 1 Control Registers
0x40B6_0000 – 0x40B6_0FFF	ECAP2_BA	ECAP 2 Control Registers
0x40B9_0000 – 0x40B9_0FFF	TRNG_BA	TRNG Control Registers

Debug IP Register Space		
0x6080_0000 - 0x608F_FFFF	SYSDBG_BA	System Debug IP Control Registers
0x60C0_0000 - 0x60DF_FFFF	A35DBG_BA	Cortex-A35 Debug IP Control Registers
0xE000_0000 - 0xE00F_FFFF	RPTDBG_BA	RTP Cortex-M4 Debug IP Control Registers

Table 6.4-1 Address Space Assignments for On-Chip Controllers

6.4.5 Power-On Setting

After power on reset, Power-On setting registers are latched to configure this chip. The Table 6.4-2 describes the definition of each power-on setting bit.

PWRONSRC (SYS_PWRONOTP[0])		**Description**	
1	0		
SYS_PWRONOTP	SYS_PWRONPIN		
[16]	-	USBP0ID	USB Port 0 ID Pin Status 0 = USB Port 0 act as a USB host. 1 = USB Port 0 act as a USB device. **Note:** The value of USBP0ID always comes from pin HSUSB0_ID.
[15:14]	[7:6]	MISCCFG	Miscellaneous Configuration If BTSRCSEL = 01, Boot from SD/eMMC. MISCCFG[0]: 0 = SD0/eMMC0 booting. (Default) 1 = SD1/eMMC1 booting. MISCCFG[1]: 0 = eMMC 4-bit booting. (Default) 1 = eMMC 8-bit booting. If BTSRCSEL = 10, the Boot from NAND Flash. 00 = Ignore. 01 = ECC is BCH T12. 10 = ECC is BCH T24. 11 = No ECC. If BTSRCSEL = 00, the Boot from SPI Flash. 00 = SPI-NAND Flash with 1-bit mode booting (Default). 10 = SPI-NOR Flash with 1-bit mode booting. If BTSRCSEL = 11, the Boot from USB. X0 = over-current low active detect. X1 = over-current high active detect. **Note:** If PWRONSRC = 0, the value of pin PG[7:6] latched to MISCCFG when pin nRESET transited from low to high. If PWRONSRC = 1, the value of MISCCFG latched from OTP's BTOPTION.

[13:12]	[5:4]	NPAGESEL	If BTSRCSEL = 10, the Boot from NAND Flash, these two bits indicates NAND Flash Page Size Selection. 00 = ignore. 01 = NAND Flash page size is 2 KB. 10 = NAND Flash page size is 4 KB. 11 = NAND Flash page size is 8 KB. If BTSRCSEL = 11, the Boot from USB, these two bits indicates USB Role and Port Selection. 00 = USBD booting. 01 = USBH port 0 boot. 10 = USBD booting. 11 = USBH port 1 boot. **Note:** If PWRONSRC = 0, the value of pin PG[5:4] latched to NPAGSEL when pin nRESET transited from low to high. If PWRONSRC = 1, the value of NPAGSEL latched from OTP's BTNANDPS.
[11:10]	[3:2]	BTSRCSEL	Boot Source Selection 00 = Boot from SPI Flash (Default). 01 = Boot from SD/eMMC. 10 = Boot from NAND Flash. 11 = Boot from USB. **Note:** If PWRONSRC = 0, the value of pin PG[3:2] latched to BTSRCSEL when pin nRESET transited from low to high. If PWRONSRC = 1, the value of BTSRCSEL latched from OTP's BTSRCSEL.
[9]	[1]	BTSRCVOL	Boot Source Interface IO Voltage 0 = Boot source interface IO voltage is 3.3V. 1 = Boot source interface IO voltage is 1.8V. **Note:** If PWRONSRC = 0, the value of pin PG[1] latched to BTSRCVOL when pin nRESET transited from low to high. If PWRONSRC = 1, the value of BTSRCVOL latched from OTP's BTSRCVOL.
[8]	[0]	SECBTDIS	Secure Boot Disable Bit 0 = Secure Boot Enabled (Default). 1 = Secure Boot Disabled. **Note:** If PWRONSRC = 0, the value of pin PG[0] latched to SECBTDIS when pin nRESET transited from low to high. If PWRONSRC = 1, the secure boot disable controlled by OTP's SECBTPSWD.
[5]	-	SD0BKEN	SD0 Back Up Boot Enable Bit 0 = SD0 back up boot Disabled (Default). 1 = SD0 back up boot Enabled. **Note:** The value of SD0BKEN always defined by OTP's SD0BKEN.
[4]	-	UR0DBGDIS	UART 0 Debug Message Output Disable Bit 0= UART 0 debug message output Enabled. 1= UART 0 debug message output Disabled. **Note:** The value of UR0DBGDIS always defined by OTP's UR0MSGODIS.

[2]	-	WDT1ON	Watchdog Timer 1 ON/OFF Selection 0 = After power-on, WDT 1 Disabled. 1 = after power-on WDT 1 Enabled. **Note:** The value of WDT1ON always defined by OTP's WDT1EN.
[1]	-	QSPI0CKSEL	QSPI0_CLK Frequency Selection 0 = QSPI0_CLK frequency is 30 MHz. 1 = QSPI0_CLK frequency is 50 MHz. **Note:** The value of QSPI0CKSEL always defined by OTP's QSPI0CKF.

Table 6.4-2 Power-On Setting Bit Description

6.4.6 Control Registers Access Attribute

The system manager control registers access attribute are shown in Table 6.4-3.

Register	SYSSIAEN (SSPCC_SINFAEN[1]) =0			SYSSIAEN (SSPCC_SINFAEN[1]) =1		
	TZS	TZNS	SUBM	TZS	TZNS	SUBM
SYS_PDID	R			R		
SYS_PWRONOTP	R			R		
SYS_PWRONPIN	R			R		
SYS_RSTSTS	R/W			R/W		
SYS_MISCRFCR	R/W	R		R/W	R	
SYS_RSTDEBCTL	R/W	R		R/W	R	
SYS_LVRDCR	R/W	R		R/W	R	
SYS_IPRST0	R/W			R/W		
SYS_IPRST1	R/W			R/W		
SYS_IPRST2	R/W			R/W		
SYS_IPRST3	R/W			R/W		
SYS_PMUCR	R/W	R		R/W	R	
SYS_DDRCQCSR	R/W	R		R/W	R	
SYS_PMUIEN	R/W	R		R/W	R	
SYS_PMUSTS	R/W	R		R/W	R	
SYS_CA35WRBADR1	R/W	R		R/W	R	
SYS_CA35WRBPAR1	R/W	R		R/W	R	
SYS_CA35WRBADR2	R/W	R		R/W	R	
SYS_CA35WRBPAR2	R/W	R		R/W	R	
SYS_USBPMISCR	R/W		R	R/W		R
SYS_USBP0PCR	R/W	R		R/W		R
SYS_USBP1PCR	R/W		R	R/W		R
SYS_MISCFCR0	R/W		R	R/W		R
SYS_MISCFCR1	R/W			R/W		
SYS_MISCIER	R/W		R	R/W		R
SYS_MISCISR	R/W		R	R/W		R
SYS_GPA_MFPL	R/W	R		R/W		
SYS_GPA_MFPH	R/W	R		R/W		
SYS_GPB_MFPL	R/W	R		R/W		
SYS_GPB_MFPH	R/W	R		R/W		
SYS_GPC_MFPL	R/W	R		R/W		

SYS_GPC_MFPH	R/W	R	R/W	
SYS_GPD_MFPL	R/W	R	R/W	
SYS_GPD_MFPH	R/W	R	R/W	
SYS_GPE_MFPL	R/W	R	R/W	
SYS_GPE_MFPH	R/W	R	R/W	
SYS_GPF_MFPL	R/W	R	R/W	
SYS_GPF_MFPH	R/W	R	R/W	
SYS_GPG_MFPL	R/W	R	R/W	
SYS_GPG_MFPH	R/W	R	R/W	
SYS_GPH_MFPL	R/W	R	R/W	
SYS_GPH_MFPH	R/W	R	R/W	
SYS_GPI_MFPL	R/W	R	R/W	
SYS_GPI_MFPH	R/W	R	R/W	
SYS_GPJ_MFPL	R/W	R	R/W	
SYS_GPJ_MFPH	R/W	R	R/W	
SYS_GPK_MFPL	R/W	R	R/W	
SYS_GPK_MFPH	R/W	R	R/W	
SYS_GPL_MFPL	R/W	R	R/W	
SYS_GPL_MFPH	R/W	R	R/W	
SYS_GPM_MFPL	R/W	R	R/W	
SYS_GPM_MFPH	R/W	R	R/W	
SYS_GPN_MFPL	R/W	R	R/W	
SYS_GPN_MFPH	R/W	R	R/W	
SYS_TSENSRFCR	R/W	R	R/W	
SYS_GMAC0MISCR	R/W	R	R/W	R
SYS_GMAC1MISCR	R/W	R	R/W	R
SYS_MACAD0LSR	R		R	
SYS_MACAD0HSR	R		R	
SYS_MACAD1LSR	R		R	
SYS_MACAD1HSR	R		R	
SYS_CSDBGCTL	R/W	R	R/W	R
SYS_GPAB_MFOS	R/W	R	R/W	
SYS_GPCD_MFOS	R/W	R	R/W	
SYS_GPEF_MFOS	R/W	R	R/W	
SYS_GPGH_MFOS	R/W	R	R/W	

SYS_GPIJ_MFOS	R/W	R	R/W	
SYS_GPKL_MFOS	R/W	R	R/W	
SYS_GPMN_MFOS	R/W	R	R/W	
SYS_UID0	R		R	
SYS_UID1	R		R	
SYS_UID2	R		R	
SYS_UCID0	R		R	
SYS_UCID1	R		R	
SYS_UCID2	R		R	
SYS_RLKTZS	R/W	RAZ/WI	R/W	RAZ/WI
SYS_RLKTZNS	R/W	R	R/W	R
SYS_RLKSUBM	R	R/W	R	R/W
SYS_NMIIEN	R	R/W	R	R/W
SYS_NMISTS	R		R	

Table 6.4-3 System Manager Control Registers Access Attribute

6.4.7 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
SYS Base Address: **SYS_BA = 0x4046_0000**				
SYS_PDID	SYS_BA+0x000	R	Product and Device Identifier Register (TZNS)	0xXXXX_XXXX
SYS_PWRONOTP	SYS_BA+0x004	R	Power-on Setting OTP Source Register (TZNS)	0xXXXX_XXXX
SYS_PWRONPIN	SYS_BA+0x008	R	Power-on Setting Pin Source Register (TZNS)	0xXXXX_XXXX
SYS_RSTSTS	SYS_BA+0x010	R/W	Reset Source Active Status Register (Shared)	0x0080_X04B
SYS_MISCRFCR	SYS_BA+0x014	R/W	Miscellaneous Reset Function Control Register	0x0005_0000
SYS_RSTDEBCTL	SYS_BA+0x018	R/W	Reset Pin De-bounce Control Register	0x8000_04B0
SYS_LVRDCR	SYS_BA+0x01C	R/W	Low Voltage Reset and Detect Control Register	0x0000_0001
SYS_IPRST0	SYS_BA+0x020	R/W	Reset Control Register 0 (Shared)	0x7000_0008
SYS_IPRST1	SYS_BA+0x024	R/W	Reset Control Register 1 (Shared)	0x0000_0000
SYS_IPRST2	SYS_BA+0x028	R/W	Reset Control Register 2 (Shared)	0x0000_0000
SYS_IPRST3	SYS_BA+0x02C	R/W	Reset Control Register 3 (Shared)	0x0000_0000
SYS_PMUCR	SYS_BA+0x030	R/W	Power Management Unit Control Register	0x0000_0000
SYS_DDRCQCSR	SYS_BA+0x034	R/W	DDR Controller Q Channel Control and Status Register	0x0001_00FF
SYS_PMUIEN	SYS_BA+0x038	R/W	Power Management Unit Interrupt Enable Register	0x0000_0000
SYS_PMUSTS	SYS_BA+0x03C	R/W	Power Management Unit Status Register	0x0000_0000
SYS_CA35WRBADR0	SYS_BA+0x040	R/W	Cortex-A35 Core 0 Warm-boot Address Register	0x0000_0000
SYS_CA35WRBPAR0	SYS_BA+0x044	R/W	Cortex-A35 Core 0 Warm-boot Parameter Register	0x0000_0000
SYS_CA35WRBADR1	SYS_BA+0x048	R/W	Cortex-A35 Core 1 Warm-boot Address Register	0x0000_0000
SYS_CA35WRBPAR1	SYS_BA+0x04C	R/W	Cortex-A35 Core 1 Warm-boot Parameter Register	0x0000_0000
SYS_USBPMISCR	SYS_BA+0x060	R/W	USB PHY Miscellaneous Control Register (TZNS)	0x7885_7885
SYS_USBP0PCR	SYS_BA+0x064	R/W	USB Port 0 PHY Control Register	0x7344_3C31
SYS_USBP1PCR	SYS_BA+0x068	R/W	USB Port 1 PHY Control Register (TZNS)	0x7344_3C31
SYS_MISCFCR0	SYS_BA+0x070	R/W	Miscellaneous Function Control Register 0 (Shared)	0x2800_0230
SYS_MISCFCR1	SYS_BA+0x074	R/W	Miscellaneous Function Control Register 1 (Shared)	0x0073_0100
SYS_MISCIER	SYS_BA+0x078	R/W	Miscellaneous Interrupt Enable Register (TZNS)	0x0000_0000
SYS_MISCISR	SYS_BA+0x07C	R/W	Miscellaneous Interrupt Status Register (TZNS)	0x0001_0000
SYS_GPA_MFPL	SYS_BA+0x080	R/W	GPIOA Low Byte Multiple Function Control Register	0x0XXX_XX00

SYS_GPA_MFPH	SYS_BA+0x084	R/W	GPIOA High Byte Multiple Function Control Register	0x0000_0000
SYS_GPB_MFPL	SYS_BA+0x088	R/W	GPIOB Low Byte Multiple Function Control Register	0x0000_0000
SYS_GPB_MFPH	SYS_BA+0x08C	R/W	GPIOB High Byte Multiple Function Control Register	0x0000_0000
SYS_GPC_MFPL	SYS_BA+0x090	R/W	GPIOC Low Byte Multiple Function Control Register	0x0000_0000
SYS_GPC_MFPH	SYS_BA+0x094	R/W	GPIOC High Byte Multiple Function Control Register	0x0000_0000
SYS_GPD_MFPL	SYS_BA+0x098	R/W	GPIOD Low Byte Multiple Function Control Register	0x0000_0000
SYS_GPD_MFPH	SYS_BA+0x09C	R/W	GPIOD High Byte Multiple Function Control Register	0x0000_0000
SYS_GPE_MFPL	SYS_BA+0x0A0	R/W	GPIOE Low Byte Multiple Function Control Register	0x0000_0000
SYS_GPE_MFPH	SYS_BA+0x0A4	R/W	GPIOE High Byte Multiple Function Control Register	0x0000_0000
SYS_GPF_MFPL	SYS_BA+0x0A8	R/W	GPIOF Low Byte Multiple Function Control Register	0x0000_0000
SYS_GPF_MFPH	SYS_BA+0x0AC	R/W	GPIOF High Byte Multiple Function Control Register	0x0000_0000
SYS_GPG_MFPL	SYS_BA+0x0B0	R/W	GPIOG Low Byte Multiple Function Control Register	0x0000_0000
SYS_GPG_MFPH	SYS_BA+0x0B4	R/W	GPIOG High Byte Multiple Function Control Register	0xXXXX_X000
SYS_GPH_MFPL	SYS_BA+0x0B8	R/W	GPIOH Low Byte Multiple Function Control Register	0x0000_0000
SYS_GPH_MFPH	SYS_BA+0x0BC	R/W	GPIOH High Byte Multiple Function Control Register	0x0000_0000
SYS_GPI_MFPL	SYS_BA+0x0C0	R/W	GPIOI Low Byte Multiple Function Control Register	0x0000_0000
SYS_GPI_MFPH	SYS_BA+0x0C4	R/W	GPIOI High Byte Multiple Function Control Register	0x0000_0000
SYS_GPJ_MFPL	SYS_BA+0x0C8	R/W	GPIOJ Low Byte Multiple Function Control Register	0x0000_0000
SYS_GPJ_MFPH	SYS_BA+0x0CC	R/W	GPIOJ High Byte Multiple Function Control Register	0x0000_0000
SYS_GPK_MFPL	SYS_BA+0x0D0	R/W	GPIOK Low Byte Multiple Function Control Register	0x0000_0000
SYS_GPK_MFPH	SYS_BA+0x0D4	R/W	GPIOK High Byte Multiple Function Control Register	0x0000_0000
SYS_GPL_MFPL	SYS_BA+0x0D8	R/W	GPIOL Low Byte Multiple Function Control Register	0x0000_0000
SYS_GPL_MFPH	SYS_BA+0x0DC	R/W	GPIOL High Byte Multiple Function Control Register	0x0000_0000
SYS_GPM_MFPL	SYS_BA+0x0E0	R/W	GPIOM Low Byte Multiple Function Control Register	0x0000_0000
SYS_GPM_MFPH	SYS_BA+0x0E4	R/W	GPIOM High Byte Multiple Function Control Register	0x0000_0000
SYS_GPN_MFPL	SYS_BA+0x0E8	R/W	GPION Low Byte Multiple Function Control Register	0x0000_0000
SYS_GPN_MFPH	SYS_BA+0x0EC	R/W	GPION High Byte Multiple Function Control Register	0x0000_0000
SYS_TSENSRFCR	SYS_BA+0x104	R/W	Temperature Sensor Function Control Register	0x1000_XXXX
SYS_GMAC0MISCR	SYS_BA+0x108	R/W	GMAC 0 Miscellaneous Control Register (TZNS)	0x0000_0000
SYS_GMAC1MISCR	SYS_BA+0x10C	R/W	GMAC 1 Miscellaneous Control Register (TZNS)	0x0000_0000
SYS_MACAD0LSR	SYS_BA+0x110	R	MAC Address 0 Low Significant Word Register (TZNS)	0xXXXX_XXXX

SYS_MACAD0HSR	SYS_BA+0x114	R	MAC Address 0 High Significant Word Register (TZNS)	0xXXXX_XXXX
SYS_MACAD1LSR	SYS_BA+0x118	R	MAC Address 1 Low Significant Word Register (TZNS)	0xXXXX_XXXX
SYS_MACAD1HSR	SYS_BA+0x11C	R	MAC Address 1 High Significant Word Register (TZNS)	0xXXXX_XXXX
SYS_CSDBGCTL	SYS_BA+0x120	R/W	CoreSight Debug Control Register	0x0000_0000
SYS_GPAB_MFOS	SYS_BA+0x140	R/W	GPIOA and GPIOB Multiple Function Output Mode Select Register	0x0000_0000
SYS_GPCD_MFOS	SYS_BA+0x144	R/W	GPIOC and GPIOD Multiple Function Output Mode Select Register	0x0000_0000
SYS_GPEF_MFOS	SYS_BA+0x148	R/W	GPIOE and GPIOF Multiple Function Output Mode Select Register	0x0000_0000
SYS_GPGH_MFOS	SYS_BA+0x14C	R/W	GPIOG and GPIOH Multiple Function Output Mode Select Register	0x0000_0000
SYS_GPIJ_MFOS	SYS_BA+0x150	R/W	GPIOI and GPIOJ Multiple Function Output Mode Select Register	0x0000_0000
SYS_GPKL_MFOS	SYS_BA+0x154	R/W	GPIOK and GPIOL Multiple Function Output Mode Select Register	0x0000_0000
SYS_GPMN_MFOS	SYS_BA+0x158	R/W	GPIOM and GPION Multiple Function Output Mode Select Register	0x0000_0000
SYS_UID0	SYS_BA+0x180	R	Unique Identifier Word 0 Register (TZNS)	0x0000_0000
SYS_UID1	SYS_BA+0x184	R	Unique Identifier Word 1 Register (TZNS)	0x0000_0000
SYS_UID2	SYS_BA+0x188	R	Unique Identifier Word 2 Register (TZNS)	0x0000_0000
SYS_UCID0	SYS_BA+0x190	R	Unique Customer Identifier Word 0 Register (TZNS)	0x0000_0000
SYS_UCID1	SYS_BA+0x194	R	Unique Customer Identifier Word 1 Register (TZNS)	0x0000_0000
SYS_UCID2	SYS_BA+0x198	R	Unique Customer Identifier Word 2 Register (TZNS)	0x0000_0000
SYS_RLKTZS	SYS_BA+0x1A0	R/W	TZS Register Lock Control Register	0x0000_0000
SYS_RLKTZNS	SYS_BA+0x1A4	R/W	TZNS Register Lock Control Register (TZNS)	0x0000_0000
SYS_RLKSUBM	SYS_BA+0x1A8	R/W	SUBM Register Lock Control Register (SUBM)	0x0000_0000
SYS_PNUMREVR	SYS_BA+0x1F0	R	Product Revision Register	0xXXXX_XXXX

6.4.8 Register Description

Product and Device Identifier Register (SYS_PDID)

Register	Offset	R/W	Description	Reset Value
SYS_PDID	SYS_BA+0x000	R	Product and Device Identifier Register (TZNS)	0xXXXX_XXXX

31	30	29	28	27	26	25	24
Reserved				DID			
23	22	21	20	19	18	17	16
PID							
15	14	13	12	11	10	9	8
INTVREF3				INTVREF2			
7	6	5	4	3	2	1	0
INTVREF1				INTVREF0			

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:24]	**DID**	**Device ID** This field stores the 4-bit Device ID loaded from OTP memory.
[23:16]	**PID**	**Product ID** This field stores the 8-bit Product ID loaded from OTP memory.
[15:12]	**INTVREF3**	**EADC Internal VREF Level 3 Voltage Trim Value** This field stores the 4-bit EADC internal VREF level 3 voltage trim value loaded from OTP memory.
[11:8]	**INTVREF2**	**EADC Internal VREF Level 2 Voltage Trim Value** This field stores the 4-bit EADC internal VREF level 2 voltage trim value loaded from OTP memory.
[7:4]	**INTVREF1**	**EADC Internal VREF Level 1 Voltage Trim Value** This field stores the 4-bit EADC internal VREF level 1 voltage trim value loaded from OTP memory.
[3:0]	**INTVREF0**	**EADC Internal VREF Level 0 Voltage Trim Value** This field stores the 4-bit EADC internal VREF level 0 voltage trim value loaded from OTP memory.

Power-on Setting OTP Source Register (SYS_PWRONOTP)

Register	Offset	R/W	Description	Reset Value
SYS_PWRONOTP	SYS_BA+0x004	R	Power-on Setting OTP Source Register (TZNS)	0xXXXX_XXXX

31	30	29	28	27	26	25	24
SECBTPSWD							
23	22	21	20	19	18	17	16
Reserved							USBP0ID
15	14	13	12	11	10	9	8
MISCCFG		NPAGESEL		BTSRCSEL		BTSRCVOL	Reserved
7	6	5	4	3	2	1	0
TSISWDIS	TSIMGLDEN	SD0BKEN	UR0DBGDIS	Reserved	WDT1ON	QSPI0CKSEL	PWRONSRC

Bits	Description	
[31:24]	**SECBTPSWD**	**Secure Boot Disable Password (Read Only)** If SECBTPSWD is 0x5A, the secure boot Disabled. **Note 1:** SECBTPSWD didn't take effect and PG[0] used as Secure Boot Disable if PWRONSRC = 0. **Note 2:** In RMA mode, SECBTPSWD didn't take effect and PG[0] used as Secure Boot Disable.
[23:17]	**Reserved**	Reserved.
[16]	**USBP0ID**	**USB Port 0 ID Pin Status** 0= USB port 0 used as a USB device. 1= USB port 0 used as a USB host.

[15:14]	**MISCCFG**	**Miscellaneous Configuration (Read Only)** If BTSRCSEL = 01, boot from SD/eMMC. **MISCCFG[0]:** 0 = SD0/eMMC0 booting. (Default) 1 = SD1/eMMC1 booting. **MISCCFG[1]:** 0 = eMMC 4-bit booting. (Default) 1 = eMMC 8-bit booting. If BTSRCSEL = 10, boot from NAND Flash. 00 = Ignore. 01 = ECC is BCH T12. 10 = ECC is BCH T24. 11 = No ECC If BTSRCSEL = 00, the Boot from SPI Flash. 00 = SPI-NAND Flash with 1-bit mode booting (Default). 10 = SPI-NOR Flash with 1-bit mode booting. If BTSRCSEL = 11, the Boot from USB. X0 = over-current low active detect. X1 = over-current high active detect. **Note:** If PWRONSRC = 0, the value of pin PG[7:6] latched to MISCCFG when pin nRESET transited from low to high. If PWRONSRC = 1, the value of MISCCFG latched from OTP's BTOPTION.
[13:12]	**NPAGESEL**	**NAND Flash Page Size Selection (Read Only)** If BTSRCSEL = 10, the Boot from NAND Flash, these two bits indicates NAND Flash Page Size Selection. 00 = Ignore. 01 = NAND Flash page size is 2 KB. 10 = NAND Flash page size is 4 KB. 11 = NAND Flash page size is 8 KB. If BTSRCSEL = 11, the Boot from USB, these two bits indicates USB Role and Port Selection. 00 = USBD booting. 01 = USBH port 0 boot. 10 = USBD booting. 11 = USBH port 1 boot **Note:** If PWRONSRC = 0, the value of pin PG[5:4] latched to NPAGSEL when pin nRESET transited from low to high. If PWRONSRC = 1, the value of NPAGSEL latched from OTP's BTNANDPS.
[11:10]	**BTSRCSEL**	**Boot Source Selection (Read Only)** 00 = Boot from SPI Flash (Default). 01 = Boot from SD/eMMC. 10 = Boot from NAND Flash. 11 = Boot from USB. **Note:** If PWRONSRC = 0, the value of pin PG[3:2] latched to BTSRCSEL when pin nRESET transited from low to high. If PWRONSRC = 1, the value of BTSRCSEL latched from OTP's BTSRCSEL.

[9]	**BTSRCVOL**	**Boot Source Interface I/O Voltage (Read Only)** 0 = Boot source interface I/O voltage is 3.3V. 1 = Boot source interface I/O voltage is 1.8V. **Note:** If PWRONSRC = 0, the value of pin PG[1] latched to BTSRCVOL when pin nRESET transited from low to high. If PWRONSRC = 1, the value of BTSRCVOL latched from OTP's BTSRCVOL.
[8]	**Reserved**	Reserved.
[7]	**TSISWDIS**	**TSI's Serial Wire Interface Disable Bit (Read Only)** 0 = TSI's serial wire interface Enabled. (Default) 1 = TSI's serial wire interface Disabled.
[6]	**TSIMGLDEN**	**TSI Image Load Control (Read Only)** 0 = Load TSI image from booting storage Disabled. (Default) 1 = Load TSI image from booting storage Enabled.
[5]	**SD0BKEN**	**SD0 Back Up Boot Enable Bit (Read Only)** 0 = SD0 back up boot Disabled (Default). 1 = SD0 back up boot Enabled. **Note:** SD0BKEN didn't take effect if BTSRCSEL= 01 and BTOPTION = 00..
[4]	**UR0DBGDIS**	**UART 0 Debug Message Output Disable Bit (Read Only)** 0= UART 0 debug message output Enabled. 1= UART 0 debug message output Disabled. **Note:** The value of UR0DBGDIS latched from OTP when pin nRESET transited from low to high.
[3]	**Reserved**	Reserved.
[2]	**WDT1ON**	**Watchdog Timer 1 ON/OFF Selection (Read Only)** 0 = After power-on, WDT 1 Disabled. 1 = after power-on WDT 1 Enabled. **Note:** The value of WDT1ON latched from OTP when pin nRESET transited from low to high.
[1]	**QSPI0CKSEL**	**QSPI0_CLK Frequency Selection (Read Only)** 0 = QSPI0_CLK frequency is 30 MHz. 1 = QSPI0_CLK frequency is 50 MHz. **Note:** The value of QSPI0CKSEL latched from OTP when pin nRESET transited from low to high.
[0]	**PWRONSRC**	**Power on Setting Source Control (Read Only)** 0 = Power on setting values come from pin. (Default) 1 = Power on setting values come from OTP.

Power-on Setting Pin Source Register (SYS_PWRONPIN)

Register	Offset	R/W	Description	Reset Value
SYS_PWRONPIN	SYS_BA+0x008	R	Power-on Setting Pin Source Register (TZNS)	0xXXXX_XXXX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MISCCFG		**NPAGESEL**		**BTSRCSEL**		**BTSRCVOL**	**SECBTDIS**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:6]	**MISCCFG**	**Miscellaneous Configuration (Read Only)** If BTSRCSEL = 01, boot from SD/eMMC. **MISCCFG[0]:** 0 = SD0/eMMC0 booting. (Default) 1 = SD1/eMMC1 booting. **MISCCFG[1]:** 0 = eMMC 4-bit booting. (Default) 1 = eMMC 8-bit booting. If BTSRCSEL = 10, boot from NAND Flash. 00 = Ignore. 01 = ECC is BCH T12. 10 = ECC is BCH T24. 11 = No ECC If BTSRCSEL = 00, the Boot from SPI Flash. 00 = SPI-NAND Flash with 1-bit mode booting (Default). 10 = SPI-NOR Flash with 1-bit mode booting. If BTSRCSEL = 11, the Boot from USB. X0 = over-current low active detect. X1 = over-current high active detect. **Note:** If PWRONSRC = 0, the value of pin PG[7:6] latched to MISCCFG when pin nRESET transited from low to high. If PWRONSRC = 1, the value of MISCCFG latched from OTP's BTOPTION.

[5:4]	**NPAGESEL**	**NAND Flash Page Size Selection (Read Only)** If BTSRCSEL = 10, the Boot from NAND Flash, these two bits indicates NAND Flash Page Size Selection. 00 = Ignore. 01 = NAND Flash page size is 2 KB. 10 = NAND Flash page size is 4 KB. 11 = NAND Flash page size is 8 KB. If BTSRCSEL = 11, the Boot from USB, these two bits indicates USB Role and Port Selection. 00 = USBD booting. 01 = USBH port 0 boot. 10 = USBD booting. 11 = USBH port 1 boot **Note:** If PWRONSRC = 0, the value of pin PG[5:4] latched to NPAGSEL when pin nRESET transited from low to high. If PWRONSRC = 1, the value of NPAGSEL latched from OTP's BTNANDPS.
[3:2]	**BTSRCSEL**	**Boot Source Selection (Read Only)** 00 = Boot from SPI Flash (Default). 01 = Boot from SD/eMMC. 10 = Boot from NAND Flash. 11 = Boot from USB. **Note:** If PWRONSRC = 0, the value of pin PG[3:2] latched to BTSRCSEL when pin nRESET transited from low to high. If PWRONSRC = 1, the value of BTSRCSEL latched from OTP's BTSRCSEL.
[1]	**BTSRCVOL**	**Boot Source Interface I/O Voltage (Read Only)** 0 = Boot source interface I/O voltage is 3.3V. 1 = Boot source interface I/O voltage is 1.8V. **Note:** If PWRONSRC = 0, the value of pin PG[1] latched to BTSRCVOL when pin nRESET transited from low to high. If PWRONSRC = 1, the value of BTSRCVOL latched from OTP's BTSRCVOL.
[0]	**SECBTDIS**	**Secure Boot Disable Bit (Read Only)** 0 = Secure Boot Enabled (Default). 1 = Secure Boot Disabled. **Note:** If PWRONSRC = 0, the value of pin PG[0] latched to SECBTDIS when pin nRESET transited from low to high. If PWRONSRC = 1, the value of SECBTDIS latched from OTP's SECBTDIS.

System Reset Status Register (SYS_RSTSTS)

Register	Offset	R/W	Description	Reset Value
SYS_RSTSTS	SYS_BA+0x010	R/W	Reset Source Active Status Register (Shared)	0x0080_X04B

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
RTPM4CPURF	RTPPMUSYSRF	RTPM4SYSRF	RTPM4LKRF	WDT2RF	WDT1RFM	Reserved	
15	14	13	12	11	10	9	8
CPU1RF	Reserved	CPU1WARMRF	CPU1DBGRF	WDT2RFA	WDT1RF	Reserved	
7	6	5	4	3	2	1	0
CPU0RF	HRESETRF	CPU0WARMRF	CPU0DBGRF	LVRF	WDT0RF	PINRF	PORF

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23]	**RTPM4CPURF**	**RTP M4 CPU Reset Flag** The RTP M4 CPU reset flag is set by hardware if software writes CM4RST (SYS_IPRST0[3]) 1 to reset Cortex-M4 Core. 0 = No reset to RTP M4 CPU. 1 = The RTP M4 CPU core is reset by software setting CM4RST (SYS_IPRST0[3]) to 1. **Note:** Write 1 to clear this bit to 0.
[22]	**RTPPMUSYSRF**	**RTP PMU System Reset Flag** The system reset flag is set by the “Reset Signal” from the PMU of Cortex-M4 Core to indicate the previous reset source. 0 = No reset from PMU of Cortex-M4. 1 = The PMU of Cortex-M4 had issued the reset signal (PMURESETREQ) to reset the system. **Note:** Write 1 to clear this bit to 0.
[21]	**RTPM4SYSRF**	**RTP M4 System Reset Flag** The system reset flag is set by the “Reset Signal” from the Cortex-M4 Core to indicate the previous reset source. 0 = No reset from Cortex-M4. 1 = The Cortex-M4 had issued the reset signal to reset the system by writing 1 to the bit SYSRESETREQ(AIRCR[2], Application Interrupt and Reset Control Register, address = 0xE000ED0C) in system control registers of Cortex-M4 core. **Note:** Write 1 to clear this bit to 0.
[20]	**RTPM4LKRF**	**RTP M4 CPU Lockup Reset Flag** 0 = No reset from RTP M4 CPU lockup happened. 1 = The RTP Cortex-M4 lockup happened and chip is reset. **Note 1:** Write 1 to clear this bit to 0. **Note 2:** When CPU lockup happened under ICE is connected, this flag will set to 1 but chip will not reset.

[19]	**WDT2RF**	**WDT 2 Reset Flag for RTP Cortex-M4** The WDT 2 reset flag is set by the "Reset Signal" from the Watchdog Timer 2 or Window Watchdog Timer 2 to indicate the previous reset source. 0 = No reset from watchdog timer 2 or window watchdog timer 2. 1 = The watchdog timer 2 or window watchdog timer 2 had issued the reset signal to reset the system. **Note 1:** Write 1 to clear this bit to 0. **Note 2:** Watchdog Timer 2 register RSTF(WDT2_CTL[2]) bit is set if the system has been reset by WDT 2 time-out reset. Window Watchdog Timer 2 register WWDTRF(WWDT2_STATUS[1]) bit is set if the system has been reset by WWDT 2 time-out reset.
[18]	**WDT1RFM**	**WDT 1 Reset Flag for RTP Cortex-M4** The WDT 1 reset flag is set by the "Reset Signal" from the Watchdog Timer 1 or Window Watchdog Timer 1 to indicate the previous reset source. 0 = No reset from watchdog timer 1 or window watchdog timer 1. 1 = The watchdog timer 1 or window watchdog timer 1 had issued the reset signal to reset the system. **Note 1:** Write 1 to clear this bit to 0. **Note 2:** Watchdog Timer 1 register RSTF(WDT1_CTL[2]) bit is set if the system has been reset by WDT 1 time-out reset. Window Watchdog Timer 1 register WWDTRF(WWDT1_STATUS[1]) bit is set if the system has been reset by WWDT 1 time-out reset. **Note 3:** This flag only take effect when WDT1RSTMEN (SYS_MISCRFCR[18]) is 1.
[17:16]	**Reserved**	Reserved.
[15]	**CPU1RF**	**Cortex-A35 Core 1 Reset Flag** The Cortex-A35 Core 1 reset flag is set by hardware if software writes CA35CR1RST (SYS_IPRST0[2]) 1 to reset Cortex-A35 Core 1. 0 = No reset to Cortex-A35 Core 1. 1 = The Cortex-A35 Core 1 is reset by software setting CA35CR1RST (SYS_IPRST0[2]) to 1. **Note:** Write 1 to clear this bit to 0.
[14]	**Reserved**	Reserved.
[13]	**CPU1WARMRF**	**Cortex-A35 Core 1 Warm Reset Flag** The Cortex-A35 core 1 warm reset flag is set by the "Reset Signal" from WARMRSTREQ of Cortex-A35 core 1 to indicate the previous reset source. The WARMRSTREQ of Cortex-A35 core 1 trigger by writing 1 to the bit RR (RMR[1], Reset Management Register of Cortex-A35 core 1) 0 = No reset from WARMRSTREQ of Cortex-A35 core 1. 1 = The Cortex-A35 core 1 had issued WARMRSTREQ reset signal to reset itself. **Note:** Write 1 to clear this bit to 0.
[12]	**CPU1DBGRF**	**Cortex-A35 Core 1 Debug Reset Flag** The Cortex-A35 core 1 debug reset flag is set by the "Reset Signal" from DBGRSTREQ of Cortex-A35 core 1 to indicate the previous reset source. 0 = No reset from DBGRSTREQ of Cortex-A35 core 1. 1 = The Cortex-A35 core 1 had issued DBGRSTREQ reset signal to reset itself. **Note:** Write 1 to clear this bit to 0.

[11]	**WDT2RFA**	**WDT 2 Reset Flag for Cortex-A35** The WDT 2 reset flag is set by the "Reset Signal" from the Watchdog Timer 2 or Window Watchdog Timer 2 to indicate the previous reset source. 0 = No reset from watchdog timer 2 or window watchdog timer 2. 1 = The watchdog timer 2 or window watchdog timer 2 had issued the reset signal to reset the system. **Note 1:** Write 1 to clear this bit to 0. **Note 2:** Watchdog Timer 2 register RSTF(WDT2_CTL[2]) bit is set if the system has been reset by WDT 2 time-out reset. Window Watchdog Timer 2 register WWDTRF(WWDT2_STATUS[1]) bit is set if the system has been reset by WWDT 2 time-out reset. **Note 3:** This flag only take effect when WDT2RSTAEN (SYS_MISCRFCR[17]) is 1.
[10]	**WDT1RF**	**WDT 1 Reset Flag** The WDT 1 reset flag is set by the "Reset Signal" from the Watchdog Timer 1 or Window Watchdog Timer 1 to indicate the previous reset source. 0 = No reset from watchdog timer 1 or window watchdog timer 1. 1 = The watchdog timer 1 or window watchdog timer 1 had issued the reset signal to reset the system. **Note 1:** Write 1 to clear this bit to 0. **Note 2:** Watchdog Timer 1 register RSTF (WDT1_CTL[2]) bit is set if the system has been reset by WDT 1 time-out reset. Window Watchdog Timer 1 register WWDTRF (WWDT1_STATUS[1]) bit is set if the system has been reset by WWDT 1 time-out reset. **Note 3:** This flag only take effect when WDT1RSTAEN (SYS_MISCRFCR[16]) is 1.
[9:8]	**Reserved**	Reserved.
[7]	**CPU0RF**	**CPU 0 Reset Flag** The CPU 0 reset flag is set by hardware if software writes CA35CR0RST (SYS_IPRST0[1]) 1 to reset Cortex-A35 Core 0. 0 = No reset to CPU. 1 = The Cortex-A35 Core 0 is reset by software setting CA35CR0RST (SYS_IPRST0[1]) to 1. **Note:** Write 1 to clear this bit to 0.
[6]	**HRESETRF**	**HRESET Reset Flag** The HRESET reset flag is set by the "Reset Signal" from the HRESET. 0 = No reset from HRESET. 1 = Reset from HRESET. **Note:** Write 1 to clear this bit to 0.
[5]	**CPU0WARMRF**	**Cortex-A35 Core 0 Warm Reset Flag** The Cortex-A35 core 0 warm reset flag is set by the "Reset Signal" from WARMRSTREQ of Cortex-A35 core 0 to indicate the previous reset source. The WARMRSTREQ of Cortex-A35 core 0 trigger by writing 1 to the bit RR (RMR[1], Reset Management Register of Cortex-A35 core 0) 0 = No reset from WARMRSTREQ of Cortex-A35 core 0. 1 = The Cortex-A35 core 0 had issued WARMRSTREQ reset signal to reset itself. **Note:** Write 1 to clear this bit to 0.
[4]	**CPU0DBGRF**	**Cortex-A35 Core 0 Debug Reset Flag** The Cortex-A35 core 0 debug reset flag is set by the "Reset Signal" from DBGRSTREQ of Cortex-A35 core 0 to indicate the previous reset source. 0 = No reset from DBGRSTREQ of Cortex-A35 core 0. 1 = The Cortex-A35 core 0 had issued DBGRSTREQ reset signal to reset itself. **Note:** Write 1 to clear this bit to 0.

[3]	**LVRF**	**LVR Reset Flag** The LVR reset flag is set by the “Reset Signal” from the Low Voltage Reset Controller to indicate the previous reset source. 0 = No reset from LVR. 1 = LVR had issued reset signal to reset the chip. **Note:** Write 1 to clear this bit to 0.
[2]	**WDT0RF**	**WDT 0 Reset Flag** The WDT 0 reset flag is set by the “Reset Signal” from the Watchdog Timer 0 or Window Watchdog Timer 0 to indicate the previous reset source. 0 = No reset from watchdog timer 0 or window watchdog timer 0. 1 = The watchdog timer 0 or window watchdog timer 0 had issued the reset signal to reset the system. Note 1: Write 1 to clear this bit to 0. Note 2: Watchdog Timer 0 register RSTF(WDT0_CTL[2]) bit is set if the system has been reset by WDT 0 time-out reset. Window Watchdog Timer 0 register WWDTRF(WWDT0_STATUS[1]) bit is set if the system has been reset by WWDT 0 time-out reset.
[1]	**PINRF**	**NRESET Pin Reset Flag** 0 = No reset from nRESET pin. 1 = nRESET pin had issued reset signal to reset the chip. **Note:** Write 1 to clear this bit to 0.
[0]	**PORF**	**POR Reset Flag** 0 = No reset from POR. 1 = POR had issued reset signal to reset the chip. **Note:** Write 1 to clear this bit to 0.

Miscellaneous Reset Function Control Register (SYS_MISCRFCR)

Register	Offset	R/W	Description	Reset Value
SYS_MISCRFCR	SYS_BA+0x014	R/W	Miscellaneous Reset Function Control Register	0x0005_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved					WDT1RSTMEN	WDT2RSTAEN	WDT1RSTAEN
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PORDISCODE							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PORDISCODE							

Bits	Description	
[31:19]	**Reserved**	Reserved.
[18]	**WDT1RSTMEN**	**WDT 1 Reset Real Time Cortex-M4 Sub-system Enable Bit (Write Protect)** 0 = WDT 1 reset real time Cortex-M4 sub-system Disabled. (Default) 1 = WDT 1 reset real time Cortex-M4 sub-system Enabled. **Note 1:** This bit is write protected. Refer to the SYS_RLKTZS register. **Note 2:** This bit keeps high if WDT1ON (SYS_PWRONOTP) is high.
[17]	**WDT2RSTAEN**	**WDT 2 Reset Cortex-A35 Enable Bit (Write Protect)** 0 = WDT 2 reset Cortex-A35 Disabled. (Default) 1 = WDT 2 reset Cortex-A35 Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[16]	**WDT1RSTAEN**	**WDT 1 Reset Cortex-A35 Enable Bit (Write Protect)** 0 = WDT 1 reset Cortex-A35 Disabled. (Default) 1 = WDT 1 reset Cortex-A35 Enabled. **Note 1:** This bit is write protected. Refer to the SYS_RLKTZS register. **Note 2:** This bit keeps high if WDT1ON (SYS_PWRONOTP) is high.
[15:0]	**PORDISCODE**	**Power-on-reset Disable Code (Write Protect)** When powered on, the Power-On-Reset (POR) circuit generates a reset signal to reset whole chip function. However, after power is ready, the POR circuit would consume a few power. To minimize the POR circuit power consumption, user to disable POR circuit by writing 0x5AA5 to this field. The POR circuit will become active again when this field is set to other value or chip is reset by other reset source, including /RESET pin, Watchdog, LVR reset and the software chip reset function. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.

Reset Pin De-bounce Control Register (SYS_RSTDEBCTL)

Register	Offset	R/W	Description	Reset Value
SYS_RSTDEBCTL	SYS_BA+0x018	R/W	Reset Pin De-bounce Control Register	0x8000_04B0

31	30	29	28	27	26	25	24
RSTDEBEN	**Reserved**						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DEBCNT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DEBCNT							

Bits	Description	
[31]	**RSTDEBEN**	**Reset Pin De-bounce Enable Bit (Write Protect)** 0 = Reset pin de-bounce Disabled. 1 = Reset pin de-bounce Enabled. (Default) **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[30:16]	**Reserved**	Reserved.
[15:0]	**DEBCNT**	**Power-on-reset Disable Code (Write Protect)** This 16-bit external RESET De-bounce Counter can specify the external RESET de-bounce time up to around 5.46ms (0xFFFF) @ XIN=12 MHz. The default external RESET de-bounce time is 0.1ms (0x04B0) @ XIN = 12 MHz. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.

Low Voltage Reset and Detect Control Register (SYS_LVRDCR)

Register	Offset	R/W	Description	Reset Value
SYS_LVRDCR	SYS_BA+0x01C	R/W	Low Voltage Reset and Detect Control Register	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**LVDODGSEL**			**LVDWKRTPEN**	**LVDWKA35EN**	**LVDSEL**	**LVDEN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**LVRDGSEL**			**LVREN**

Bits	Description	
[31:15]	**Reserved**	Reserved.
[14:12]	**LVDODGSEL**	**LVD Output De-glitch Time Select (Write Protect)** 000 = Without de-glitch function. 001 = 4 system clock (LVRDGCLK). 010 = 8 system clock (LVRDGCLK). 011 = 16 system clock (LVRDGCLK). 100 = 32 system clock (LVRDGCLK). 101 = 64 system clock (LVRDGCLK). 110 = 128 system clock (LVRDGCLK). 111 = 256 system clock (LVRDGCLK). **Note 1:** These bits are write protected. Refer to the SYS_RLKTZS register. **Note 2:** Refer to LVRDBSEL (CLK_CLKSEL0[3]) for LVRDGCLK clock source selection.
[11]	**LVDWKRTPEN**	**Low Voltage Detect Wake-up RTP Cortex-M4 Enable Control Bit (Write Protect)** 0 = Low voltage detection wakeup RTP Cortex-M4 Disabled. 1 = Low voltage detection wakeup RTP Cortex-M4 Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[10]	**LVDWKA35EN**	**Low Voltage Detect Wake-up Cortex-A35 Enable Control Bit (Write Protect)** 0 = Low voltage detection wakeup A35 Disabled. 1 = Low voltage detection wakeup A35 Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[9]	**LVDSEL**	**Low Voltage Detect Threshold Selection (Write Protect)** 0 = Low voltage detection level is 2.8V. 1 = Low voltage detection level is 2.6V. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.

[8]	**LVDEN**	**Low Voltage Detect Enable Bit (Write Protect)** 0 = Low voltage detect function Disabled. 1 = Low voltage detect function Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[7:4]	**Reserved**	Reserved.
[3:1]	**LVRDGSEL**	**LVR Output De-glitch Time Select (Write Protect)** 000 = Without de-glitch function. 001 = 4 system clock (LVRDGCLK). 010 = 8 system clock (LVRDGCLK). 011 = 16 system clock (LVRDGCLK). 100 = 32 system clock (LVRDGCLK). 101 = 64 system clock (LVRDGCLK). 110 = 128 system clock (LVRDGCLK). 111 = 256 system clock (LVRDGCLK). **Note 1:** These bits are write protected. Refer to the SYS_RLKTZS register. **Note 2:** Refer to LVRDBSEL (CLK_CLKSEL0[3]) for LVRDGCLK clock source selection.
[0]	**LVREN**	**Low Voltage Reset Enable Bit (Write Protect)** 0 = Low voltage reset function Disabled. 1 = Low voltage reset function Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.

Reset Control Register 0 (SYS_IPRST0)

This register is the reset enable of different IPs. Write 1 to any bit of this register trigger reset controller to reset the IP.

The right of write 1 to bit of this register controls by register SSPCC_PSSETx (x=0~11). If the security attribute of peripheral set to SUBM, its corresponding reset bit only can be written by RTP Cortex-M4 core. If the security attribute of peripheral set to TZNS, its corresponding reset bit only can be written by Cortex-A35 core in Non-Secure state. If the security attribute of peripheral set to TZS, its corresponding reset bit only can be written by Cortex-A35 core in Secure state.

Register	Offset	R/W	Description	Reset Value
SYS_IPRST0	SYS_BA+0x020	R/W	Reset Control Register 0 (Shared)	0x7000_0008

31	30	29	28	27	26	25	24
Reserved	DDRPUBRST	MCTLCRST	MCTLPRST	GPIORST	NANDRST	SDH1RST	SDH0RST
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved	HSUSBDRST	HSUSBH1RST	HSUSBH0RST	EBIRST	HWSEMRST	GMAC1RST	GMAC0RST
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
WRHO1RST	WRHO0RST	VDECRST	GFXRST	CCAP1RST	CCAP0RST	DISPCRST	Reserved
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PDMA3RST	PDMA2RST	PDMA1RST	PDMA0RST	CM4RST	CA35CR1RST	CA35CR0RST	CHIPRST

Bits	Description	
[31]	**Reserved**	Reserved.
[30]	**DDRPUBRST**	**DDR PHY PUB Reset Enable Bit (Write Protect)** 0 = DDR PHY PUB reset Disabled. 1 = DDR PHY PUB reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[29]	**MCTLCRST**	**DDR Memory Controller MCTL2 Core Reset Enable Bit (Write Protect)** 0 = DDR Memory Controller MCTL2 core reset Disabled. 1 = DDR Memory Controller MCTL2 core reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[28]	**MCTLPRST**	**DDR Memory Controller MCTL2 APB Interface Reset Enable Bit (Write Protect)** Write 1 to enable MCTL2 APB reset to reset APB interface logic of MCTL2. Write 0 to trigger a reset disable procedure and this bit cleared automatically after 128 pclk. 0 = DDR Memory Controller MCTL2 APB interface reset Disabled. 1 = DDR Memory Controller MCTL2 APB interface reset Enabled. **Note 1:** Once trigger reset disable procedure, it's necessary to poll MCTLPRST till its 0. **Note 2:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.

[27]	**GPIORST**	**GPIO Reset Enable Bit (Write Protect)** 0 = GPIO reset Disabled. 1 = GPIO reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[26]	**NANDRST**	**NAND Flash Controller Reset Enable Bit (Write Protect)** 0 = NAND Flash controller reset Disabled. 1 = NAND Flash controller reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[25]	**SDH1RST**	**SDIO Controller 1 Reset Enable Bit (Write Protect)** 0 = SDIO controller 1 reset Disabled. 1 = SDIO controller 1 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[24]	**SDH0RST**	**SDIO Controller 0 Reset Enable Bit (Write Protect)** 0 = SDIO controller 0 reset Disabled. 1 = SDIO controller 0 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[23]	**Reserved**	Reserved.
[22]	**HSUSBDRST**	**High-speed USB Device Controller Reset Enable Bit (Write Protect)** 0 = High-Speed USB device controller reset Disabled. 1 = High-Speed USB device controller reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[21]	**HSUSBH1RST**	**High-speed USB Host Controller 1 Reset Enable Bit (Write Protect)** 0 = High-Speed USB host controller 1 reset Disabled. 1 = High-Speed USB host controller 1 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[20]	**HSUSBH0RST**	**High-speed USB Host Controller 0 Reset Enable Bit (Write Protect)** 0 = High-Speed USB host controller 0 reset Disabled. 1 = High-Speed USB host controller 0 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[19]	**EBIRST**	**EBI Controller Reset (Write Protect)** 0 = EBI controller reset Disabled. 1 = EBI controller reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[18]	**HWSEMRST**	**Hardware Semaphore Reset Enable Bit (Write Protect)** 0 = Hardware Semaphore reset Disabled. 1 = Hardware Semaphore reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.

[17]	**GMAC1RST**	**Gigabit Ethernet MAC 1 Reset Enable Bit (Write Protect)** 0 = Gigabit Ethernet MAC 1 reset Disabled. 1 = Gigabit Ethernet MAC 1 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[16]	**GMAC0RST**	**Gigabit Ethernet MAC 0 Reset Enable Bit (Write Protect)** 0 = Gigabit Ethernet MAC 0 reset Disabled. 1 = Gigabit Ethernet MAC 0 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[15]	**WRHO1RST**	**Wormhole 1 Reset Enable Bit (Write Protect)** 0 = Wormhole 1 reset Disabled. 1 = Wormhole 1 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[14]	**WRHO0RST**	**Wormhole 0 Reset Enable Bit (Write Protect)** 0 = Wormhole 0 reset Disabled. 1 = Wormhole 0 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[13]	**VDECRST**	**Video Decoder Reset Enable Bit (Write Protect)** 0 = Video Decoder (H.264/JPEG) reset Disabled. 1 = Video Decoder (H.264/JPEG) reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[12]	**GFXRST**	**Graphic Engine Reset Enable Bit (Write Protect)** 0 = Graphic Engine reset Disabled. 1 = Graphic Engine reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[11]	**CCAP1RST**	**Camera Capture Sensor Interface 1 Reset Enable Bit (Write Protect)** 0 = Camera Capture sensor interface 1 reset Disabled. 1 = Camera Capture sensor interface 1 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[10]	**CCAP0RST**	**Camera Capture Sensor Interface 0 Reset Enable Bit (Write Protect)** 0 = Camera Capture sensor interface 0 reset Disabled. 1 = Camera Capture sensor interface 0 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[9]	**DISPCRST**	**LCD Display Controller Reset Enable Bit (Write Protect)** 0 = LCD Display Controller reset Disabled. 1 = LCD Display Controller reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[8]	**Reserved**	Reserved.

[7]	**PDMA3RST**	**PDMA 3 Reset Enable Bit (Write Protect)** 0 = PDMA 3 reset Disabled. 1 = PDMA 3 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[6]	**PDMA2RST**	**PDMA2 Reset Enable Bit (Write Protect)** 0 = PDMA 2 reset Disabled. 1 = PDMA 2 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[5]	**PDMA1RST**	**PDMA 1 Reset Enable Bit (Write Protect)** 0 = PDMA 1 reset Disabled. 1 = PDMA 1 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[4]	**PDMA0RST**	**PDMA 0 Reset Enable Bit (Write Protect)** 0 = PDMA 0 reset Disabled. 1 = PDMA 0 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[3]	**CM4RST**	**Cortex-M4 Core Reset (Write Protect)** 0 = Cortex-M4 core reset Disabled. 1 = Cortex-M4 core reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[2]	**CA35CR1RST**	**Cortex-A35 Core 1 One-shot Reset (Write Protect)** 0 = Cortex-A35 core 1 one-shot reset Disabled. 1 = Cortex-A35 core 1 one-shot reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[1]	**CA35CR0RST**	**Cortex-A35 Core 0 One-shot Reset (Write Protect)** 0 = Cortex-A35 core 0 one-shot reset Disabled. 1 = Cortex-A35 core 0 one-shot reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[0]	**CHIPRST**	**Chip One-shot Reset Enable Bit (Write Protect)** 0 = Chip one-shot reset Disabled. 1 = Chip one-shot reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.

Reset Control Register 1 (SYS_IPRST1)

Register	Offset	R/W	Description	Reset Value
SYS_IPRST1	SYS_BA+0x024	R/W	Reset Control Register 1 (Shared)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		I2S0RST	EADC0RST	Reserved		CANFD1RST	CANFD0RST
23	22	21	20	19	18	17	16
UART7RST	UART6RST	UART5RST	UART4RST	UART3RST	UART2RST	UART1RST	UART0RST
15	14	13	12	11	10	9	8
SPI2RST	SPI1RST	SPI0RST	QSPI0RST	I2C3RST	I2C2RST	I2C1RST	I2C0RST
7	6	5	4	3	2	1	0
Reserved		TMR3RST	TMR2RST	TMR1RST	TMR0RST	Reserved	

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29]	**I2S0RST**	**I^2S 0 Reset Enable Bit (Write Protect)** 0 = I^2S 0 reset Disabled. 1 = I^2S 0 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[28]	**EADC0RST**	**EADC 0 Reset Enable Bit (Write Protect)** 0 = EADC 0 reset Disabled. 1 = EADC 0 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[27:26]	**Reserved**	Reserved.
[25]	**CANFD1RST**	**CAN FD 1 Reset Enable Bit (Write Protect)** 0 = CAN FD 1 reset Disabled. 1 = CAN FD 1 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[24]	**CANFD0RST**	**CAN FD 0 Reset Enable Bit (Write Protect)** 0 = CAN FD 0 reset Disabled. 1 = CAN FD 0 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[23]	**UART7RST**	**UART 7 Reset Enable Bit (Write Protect)** 0 = UART 7 reset Disabled. 1 = UART 7 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.

[22]	**UART6RST**	**UART 6 Reset Enable Bit (Write Protect)** 0 = UART 6 reset Disabled. 1 = UART 6 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[21]	**UART5RST**	**UART 5 Reset Enable Bit (Write Protect)** 0 = UART 5 reset Disabled. 1 = UART 5 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[20]	**UART4RST**	**UART 4 Reset Enable Bit (Write Protect)** 0 = UART 4 reset Disabled. 1 = UART 4 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[19]	**UART3RST**	**UART 3 Reset Enable Bit (Write Protect)** 0 = UART 3 reset Disabled. 1 = UART 3 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[18]	**UART2RST**	**UART 2 Reset Enable Bit (Write Protect)** 0 = UART 2 reset Disabled. 1 = UART 2 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[17]	**UART1RST**	**UART 1 Reset Enable Bit (Write Protect)** 0 = UART 1 reset Disabled. 1 = UART 1 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[16]	**UART0RST**	**UART 0 Reset Enable Bit (Write Protect)** 0 = UART 0 reset Disabled. 1 = UART 0 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[15]	**SPI2RST**	**SPI 2 Reset Enable Bit (Write Protect)** 0 = SPI 2 reset Disabled. 1 = SPI 2 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[14]	**SPI1RST**	**SPI 1 Reset Enable Bit (Write Protect)** 0 = SPI 1 reset Disabled. 1 = SPI 1 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.

[13]	**SPI0RST**	**SPI 0 Reset Enable Bit (Write Protect)** 0 = SPI 0 reset Disabled. 1 = SPI 0 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[12]	**QSPI0RST**	**QSPI 0 Reset Enable Bit (Write Protect)** 0 = QSPI 0 reset Disabled. 1 = QSPI 0 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[11]	**I2C3RST**	**I^2C 3 Reset Enable Bit (Write Protect)** 0 = I^2C 3 reset Disabled. 1 = I^2C 3 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[10]	**I2C2RST**	**I^2C 2 Reset Enable Bit (Write Protect)** 0 = I^2C 2 reset Disabled. 1 = I^2C 2 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[9]	**I2C1RST**	**I^2C 1 Reset Enable Bit (Write Protect)** 0 = I^2C 1 reset Disabled. 1 = I^2C 1 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[8]	**I2C0RST**	**I^2C 0 Reset Enable Bit (Write Protect)** 0 = I^2C 0 reset Disabled. 1 = I^2C 0 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[7:6]	**Reserved**	Reserved.
[5]	**TMR3RST**	**TIMER 3 Reset Enable Bit (Write Protect)** 0 = TIMER 3 reset Disabled. 1 = TIMER 3 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[4]	**TMR2RST**	**TIMER 2 Reset Enable Bit (Write Protect)** 0 = TIMER 2 reset Disabled. 1 = TIMER 2 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[3]	**TMR1RST**	**TIMER 1 Reset Enable Bit (Write Protect)** 0 = TIMER 1 reset Disabled. 1 = TIMER 1 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.

[2]	**TMR0RST**	**TIMER 0 Reset Enable Bit (Write Protect)** 0 = TIMER 0 reset Disabled. 1 = TIMER 0 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[1:0]	**Reserved**	Reserved.

Reset Control Register 2 (SYS_IPRST2)

Register	Offset	R/W	Description	Reset Value
SYS_IPRST2	SYS_BA+0x028	R/W	Reset Control Register 2 (Shared)	0x0000_0000

31	30	29	28	27	26	25	24
ADC0RST	Reserved		CANFD2RST	ECAP1RST	ECAP0RST	Reserved	
23	22	21	20	19	18	17	16
QEI1RST	QEI0RST	Reserved				EPWM1RST	EPWM0RST
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	SPI3RST	Reserved	QSPI1RST	Reserved		SC1RST	SC0RST

Bits	Description	
[31]	**ADC0RST**	**ADC 0 Reset Enable Bit (Write Protect)** 0 = ADC 0 reset Disabled. 1 = ADC 0 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[30:29]	**Reserved**	Reserved.
[28]	**CANFD2RST**	**CAN FD 2 Reset Enable Bit (Write Protect)** 0 = CAN FD 2 reset Disabled. 1 = CAN FD 2 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[27]	**ECAP1RST**	**ECAP 1 Reset Enable Bit (Write Protect)** 0 = ECAP 1 reset Disabled. 1 = ECAP 1 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[26]	**ECAP0RST**	**ECAP 0 Reset Enable Bit (Write Protect)** 0 = ECAP 0 reset Disabled. 1 = ECAP 0 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[25:24]	**Reserved**	Reserved.
[23]	**QEI1RST**	**QEI 1 Reset Enable Bit (Write Protect)** 0 = QEI 1 reset Disabled. 1 = QEI 1 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.

[22]	**QEI0RST**	**QEI 0 Reset Enable Bit (Write Protect)** 0 = QEI 0 reset Disabled. 1 = QEI 0 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[21:18]	**Reserved**	Reserved.
[17]	**EPWM1RST**	**EPWM 1 Reset Enable Bit (Write Protect)** 0 = EPWM 1 reset Disabled. 1 = EPWM 1 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[16]	**EPWM0RST**	**EPWM 0 Reset Enable Bit (Write Protect)** 0 = EPWM 0 reset Disabled. 1 = EPWM 0 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[15:7]	**Reserved**	Reserved.
[6]	**SPI3RST**	**SPI 3 Reset Enable Bit (Write Protect)** 0 = SPI 3 reset Disabled. 1 = SPI 3 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[5]	**Reserved**	Reserved.
[4]	**QSPI1RST**	**QSPI 1 Reset Enable Bit (Write Protect)** 0 = QSPI 1 reset Disabled. 1 = QSPI 1 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[3:2]	**Reserved**	Reserved.
[1]	**SC1RST**	**SC 1 Reset Enable Bit (Write Protect)** 0 = SC 1 reset Disabled. 1 = SC 1 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[0]	**SC0RST**	**SC 0 Reset Enable Bit (Write Protect)** 0 = SC 0 reset Disabled. 1 = SC 0 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.

Reset Control Register 3 (SYS_IPRST3)

Register	Offset	R/W	Description	Reset Value
SYS_IPRST3	SYS_BA+0x02C	R/W	Reset Control Register 3 (Shared)	0x0000_0000

31	30	29	28	27	26	25	24
SSPCCRST	SSMCCRST	Reserved	GICRST	Reserved			KPIRST
23	22	21	20	19	18	17	16
CANFD3RST	QEI2RST	ECAP2RST	EPWM2RST	I2C5RST	I2C4RST	I2S1RST	UART16RST
15	14	13	12	11	10	9	8
UART15RST	UART14RST	UART13RST	UART12RST	UART11RST	UART10RST	UART9RST	UART8RST
7	6	5	4	3	2	1	0
TMR11RST	TMR10RST	TMR9RST	TMR8RST	TMR7RST	TMR6RST	TMR5RST	TMR4RST

Bits	Description	
[31]	**SSPCCRST**	**SSPCC Reset Enable Bit (Write Protect)** 0 = SSPCC reset Disabled. 1 = SSPCC reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[30]	**SSMCCRST**	**SSMCC Reset Enable Bit (Write Protect)** 0 = SSMCC reset Disabled. 1 = SSMCC reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[29]	**Reserved**	Reserved.
[28]	**GICRST**	**GIC Reset Enable Bit (Write Protect)** 0 = GIC reset Disabled. 1 = GIC reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[27:25]	**Reserved**	Reserved.
[24]	**KPIRST**	**KPI Reset Enable Bit (Write Protect)** 0 = KPI reset Disabled. 1 = KPI reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[23]	**CANFD3RST**	**CAN FD 3 Reset Enable Bit (Write Protect)** 0 = CAN FD 3 reset Disabled. 1 = CAN FD 3 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.

[22]	**QEI2RST**	**QEI 2 Reset Enable Bit (Write Protect)** 0 = QEI 2 reset Disabled. 1 = QEI 2 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[21]	**ECAP2RST**	**ECAP 2 Reset Enable Bit (Write Protect)** 0 = ECAP 2 reset Disabled. 1 = ECAP 2 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[20]	**EPWM2RST**	**EPWM 2 Reset Enable Bit (Write Protect)** 0 = EPWM 2 reset Disabled. 1 = EPWM 2 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[19]	**I2C5RST**	**I^2C 5 Reset Enable Bit (Write Protect)** 0 = I^2C 5 reset Disabled. 1 = I^2C 5 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[18]	**I2C4RST**	**I^2C 4 Reset Enable Bit (Write Protect)** 0 = I^2C 4 reset Disabled. 1 = I^2C 4reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[17]	**I2S1RST**	**I^2S 1 Reset Enable Bit (Write Protect)** 0 = I^2S 1 reset Disabled. 1 = I^2S 1 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[16]	**UART16RST**	**UART 16 Reset Enable Bit (Write Protect)** 0 = UART 16 reset Disabled. 1 = UART 16 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[15]	**UART15RST**	**UART 15 Reset Enable Bit (Write Protect)** 0 = UART 15 reset Disabled. 1 = UART 15 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[14]	**UART14RST**	**UART 14 Reset Enable Bit (Write Protect)** 0 = UART 14 reset Disabled. 1 = UART 14 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.

[13]	**UART13RST**	**UART 13 Reset Enable Bit (Write Protect)** 0 = UART 13 reset Disabled. 1 = UART 13 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[12]	**UART12RST**	**UART 12 Reset Enable Bit (Write Protect)** 0 = UART 12 reset Disabled. 1 = UART 12 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[11]	**UART11RST**	**UART 11 Reset Enable Bit (Write Protect)** 0 = UART 11 reset Disabled. 1 = UART 11 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[10]	**UART10RST**	**UART 10 Reset Enable Bit (Write Protect)** 0 = UART 10 reset Disabled. 1 = UART 10 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[9]	**UART9RST**	**UART 9 Reset Enable Bit (Write Protect)** 0 = UART 9 reset Disabled. 1 = UART 9 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[8]	**UART8RST**	**UART 8 Reset Enable Bit (Write Protect)** 0 = UART 8 reset Disabled. 1 = UART 8 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[7]	**TMR11RST**	**TIMER 11 Reset Enable Bit (Write Protect)** 0 = TIMER 11 reset Disabled. 1 = TIMER 11 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[6]	**TMR10RST**	**TIMER 10 Reset Enable Bit (Write Protect)** 0 = TIMER 10 reset Disabled. 1 = TIMER 10 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[5]	**TMR9RST**	**TIMER 9 Reset Enable Bit (Write Protect)** 0 = TIMER 9 reset Disabled. 1 = TIMER 9 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.

[4]	**TMR8RST**	**TIMER 8 Reset Enable Bit (Write Protect)** 0 = TIMER 8 reset Disabled. 1 = TIMER 8 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[3]	**TMR7RST**	**TIMER 7 Reset Enable Bit (Write Protect)** 0 = TIMER 7 reset Disabled. 1 = TIMER 7 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[2]	**TMR6RST**	**TIMER 6 Reset Enable Bit (Write Protect)** 0 = TIMER 6 reset Disabled. 1 = TIMER 6 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[1]	**TMR5RST**	**TIMER 5 Reset Enable Bit (Write Protect)** 0 = TIMER 5 reset Disabled. 1 = TIMER 5 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[0]	**TMR4RST**	**TIMER 4 Reset Enable Bit (Write Protect)** 0 = TIMER 4 reset Disabled. 1 = TIMER 4 reset Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.

Power Management Unit Control Register (SYS_PMUCR)

Register	Offset	R/W	Description	Reset Value
SYS_PMUCR	SYS_BA+0x030	R/W	Power Management Unit Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					RTPDBPDEN	Reserved	RTPPDEN
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved					A35DBPDEN	Reserved	A35PDEN
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PWRACKTO				PWRSTBTM			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	PDWKDLY	Reserved	AUTOL2FDIS	Reserved			A35PGEN

Bits	Description	
[31:27]	**Reserved**	Reserved.
[26]	**RTPDBPDEN**	**RTP M4 Entering Power-down Even ICE Connected (Write Protect, SubM)** 0 = RTP M4 does not enter Power-down during Debug mode. 1 = RTP M4 enters power-down in Debug mode. **Note:** This bit is write protected. Refer to the SYS_RLKSUBM register.
[25]	**Reserved**	Reserved.
[24]	**RTPPDEN**	**RTP M4 Power Down Enable (Write Protect, SubM)** When this bit is set to 1, Power-down mode is enabled and the chip keeps active till the CPU sleep mode is also active and then the chip enters Power-down mode. When chip wakes up from Power-down mode, this bit is auto cleared. Users need to set this bit again for next Power-down. In Power-down mode, HXT, HIRC, PLL and system clock will be disabled and ignored the clock source selection. The clocks of peripheral are not controlled by Power-down mode, if the peripheral clock source is from LXT or LIRC. 0 = Chip operating normally or chip in idle mode because of WFI command. 1 = Chip waits CPU sleep command WFI and then enters Power-down mode. **Note:** This bit is write protected. Refer to the SYS_RLKSUBM register.
[23:19]	**Reserved**	Reserved.
[18]	**A35DBPDEN**	**Cortex-A35 Entering Power-down Even ICE Connected (Write Protect)** 0 = Cortex-A35 does not enter Power-down during Debug mode. 1 = Cortex-A35 enters power-down in Debug mode. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[17]	**Reserved**	Reserved.

[16]	**A35PDEN**	**Cortex-A35 Power Down Enable Bit (Write Protect)** When this bit is set to 1, Power-down mode is enabled and the chip keeps active till the CPU sleep mode is also active and then the chip enters Power-down mode. When chip wakes up from Power-down mode, this bit is auto cleared. Users need to set this bit again for next Power-down. In Power-down mode, HXT, HIRC, PLL and system clock will be disabled and ignored the clock source selection. The clocks of peripheral are not controlled by Power-down mode, if the peripheral clock source is from LXT or LIRC. 0 = Chip operating normally or chip in idle mode because of WFI command. 1 = Chip waits CPU sleep command WFI and then enters Power-down mode. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[15:12]	**PWRACKTO**	**Power Gating Acknowledgement Time Out Selection (Write Protect)** 0000 = 20us (Default). 0001 = 30us. 0010 = 40us. 0011 = 50us. 0100 = 60us. 0101 = 80us. 0110 = 100us. 0111 = 200us. 1000 = 400us. 1001 = 600us. 1010 = 800us. 1011 = 1ms. 1100 = 2ms. 1101 = 4ms. 1110 = 5.4ms. 1111 = 0us. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[11:8]	**PWRSTBTM**	**Power Gating Acknowledgement Stable Time (Write Protect)** The PWRSTBTM indicates the stable time after receiving power gating acknowledgement. 0000 = 5us (Default). 0001 = 10us. 0010 = 20us. 0011 = 40us. 0100 = 60us. 0101 = 80us. 0110 = 100us. 0111 = 200us. 1000 = 400us. 1001 = 600us. 1010 = 800us. 1011 = 1ms. 1100 = 2ms. 1101 = 4ms. 1110 = 5.4ms. 1111 = 0us. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[7]	**Reserved**	Reserved.

[6]	**PDWKDLY**	**Wake-up Delay Counter Enable Bit (Write Protect)** When the Cortex-A35 wakes up from Power-down mode, the clock control will delay certain clock cycles to wait system clock stable. The delayed clock cycle is 4096 clock cycles when chip works at 24 Mhz external high speed crystal oscillator (HXT), and 256 clock cycles when chip works at 12 MHz internal high speed RC oscillator (HIRC). 0 = Wake-up delay counter Disabled. 1 = Wake-up delay counter Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[5]	**Reserved**	Reserved.
[4]	**AUTOL2FDIS**	**Automatic L2 Cache Flush Disable (Write Protect)** 0 = Automatic L2 cache flush Enabled. 1 = Automatic L2 cache flush Disabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[3:1]	**Reserved**	Reserved.
[0]	**A35PGEN**	**Cortex-A35 Power Gating Enable (Write Protect)** 0 = Cortex-A35 dual core power gating Disabled. 1 = Cortex-A35 dual core power gating Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.

DDR Controller Q Channel Control and Status Register (SYS_DDRCQCSR)

Register	Offset	R/W	Description	Reset Value
SYS_DDRCQCSR	SYS_BA+0x034	R/W	DDR Controller Q Channel Control and Status Register	0x0001_00FF

31	30	29	28	27	26	25	24
DDRQACKTO				DDRQREQDLY			
23	22	21	20	19	18	17	16
Reserved						DDRCQDENYIF	DDRCQBYPAS
15	14	13	12	11	10	9	8
AXIQDENYIF							
7	6	5	4	3	2	1	0
AXIQBYPAS							

Bits	Description	
[31:28]	**DDRQACKTO**	**DDR Controller Q Channel Acknowledgement Time Out Selection (Write Protect)** 0000 = 20us (Default). 0001 = 30us. 0010 = 40us. 0011 = 50us. 0100 = 60us. 0101 = 80us. 0110 = 100us. 0111 = 200us. 1000 = 400us. 1001 = 600us. 1010 = 800us. 1011 = 1ms. 1100 = 2ms. 1101 = 3ms. 1110 = 5.4ms. 1111 = 0us. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.

[27:24]	**DDRQREQDLY**	**DDR Controller Q Channel Request Delay Time Selection (Write Protect)** 0000 = 20us (Default). 0001 = 30us. 0010 = 40us. 0011 = 50us. 0100 = 60us. 0101 = 80us. 0110 = 100us. 0111 = 200us. 1000 = 400us. 1001 = 600us. 1010 = 800us. 1011 = 1ms. 1100 = 2ms. 1101 = 3ms. 1110 = 5.4ms. 1111 = 0us. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[23:18]	**Reserved**	Reserved.
[17]	**DDRCQDENYIF**	**DDR Controller Core Q Channel Deny Interrupt Flag** 0 = Q channel power down request accept by DDR controller core. 1 = Q channel power down request reject by DDR controller core and DDR controller wouldn't enter self-refresh mode.
[16]	**DDRCQBYPAS**	**DDR Controller Core Q Channel Handshake Bypass (Write Protect)** 0 = Q channel handshake of DDR controller core Not Bypassed. 1 = Q channel handshake of DDR controller core Bypassed. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[15:8]	**AXIQDENYIF**	**DDR Controller AXI Port 7 ~ Port 0 Q Channel Interrupt Flag** 0 = Q channel power down request accept by DDR controller AXI port x (x=0, 1, …, 7). 1 = Q channel power down request reject by DDR controller AXI port x, AXI port x wouldn't enter low power mode and clock of AXI port x keep clocking (x=0, 1, …, 7).
[7:0]	**AXIQBYPAS**	**DDR Controller AXI Port 7 ~ Port 0 Q Channel Handshake Bypass (Write Protect)** The AXIQBYPAS indicates to bypass DDR controller's AXI port Q channel handshake mechanism. The each bit of AXIQBYPAS is for corresponding AXI port of DDR controller. AXIQBYPAS[x] 0 = Q channel handshake mechanism of AXI port x Not Bypassed (x=0, 1, …, 7). 1 = Q channel handshake mechanism of AXI port x Bypassed (x=0, 1, …, 7). **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.

Power Management Unit Interrupt Enable Register (SYS_PMUIEN)

Register	Offset	R/W	Description	Reset Value
SYS_PMUIEN	SYS_BA+0x038	R/W	Power Management Unit Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			RTPPDWKIEN	Reserved			A35PDWKIEN
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							PMUIEN

Bits	Description	
[31:13]	**Reserved**	Reserved.
[12]	**RTPPDWKIEN**	**RTP M4 Power-down Wake-up Interrupt Enable Control Bit (Write Protect, SubM)** 0 = RTP M4 wake-up interrupt Disabled. 1 = RTP M4 wake-up interrupt Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKSUBM register.
[11:9]	**Reserved**	Reserved.
[8]	**A35PDWKIEN**	**Cortex-A35 Power-down Wake-up Interrupt Enable Control Bit (Write Protect)** 0 = Cortex-a35 wake-up interrupt Disabled. 1 = Cortex-a35 wake-up interrupt Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[7:1]	**Reserved**	Reserved.
[0]	**PMUIEN**	**PMU Interrupt Enable Control Bit (Write Protect)** 0 = PMU interrupt Disabled. 1 = PMU interrupt Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.

Power Management Unit Status Register (SYS_PMUSTS)

Register	Offset	R/W	Description	Reset Value
SYS_PMUSTS	SYS_BA+0x03C	R/W	Power Management Unit Status Register	0x0000_0000

31	30	29	28	27	26	25	24
PWRACKCNT							
23	22	21	20	19	18	17	16
PWRACKCNT							
15	14	13	12	11	10	9	8
Reserved			RTPPDWKIF	Reserved			A35PDWKIF
7	6	5	4	3	2	1	0
Reserved		L2FDONE	Reserved			PGTOIF	PMUIF

Bits	Description	
[31:16]	**PWRACKCNT**	**Power Gating Acknowledgement Timing Counter Value (Read Only)** The PWRACKCNT show the value of power gating acknowledgement time-out counter.
[15:13]	**Reserved**	Reserved.
[12]	**RTPPDWKIF**	**RTP M4 Power-down Wake-up Interrupt Flag** 0 = RTP M4 didn't wake-up from power-down mode. 1 = RTP M4 receive a wake-up event and wake-up from power-down mode.
[11:9]	**Reserved**	Reserved.
[8]	**A35PDWKIF**	**Cortex-A35 Power-down Wake-up Interrupt Flag** 0 = Cortex-A35 didn't wake-up from power-down mode. 1 = Cortex-A35 receive a wake-up event and wake-up from power-down mode.
[7:6]	**Reserved**	Reserved.
[5]	**L2FDONE**	**Cortex-A35 L2 Cache Flush Done Status** 0 = Cortex-A35 L2 cache flush didn't finish yet. 1 = Cortex-A35 L2 cache flush done.
[4:2]	**Reserved**	Reserved.
[1]	**PGTOIF**	**Power Gating Time Out Interrupt Flag** 0 = Power gating acknowledgement counter is not time-out yet. 1 = Power gating acknowledgement counter is time-out.
[0]	**PMUIF**	**PMU Interrupt Flag** When PMUIEN high, this bit high to indicate that PGTOIF, DDRCQDENYIF or AXIQDENYIF is active. When PMUIEN low, this bit didn't take effect. 0 = PGTOIF, DDRCQDENYIF and AXIQDENYIF are not active. 1 = PGTOIF, DDRCQDENYIF or AXIQDENYIF is active.

Cortex-A35 Core 0 Warm-boot Address Register (SYS_CA35WRBADR0)

Register	Offset	R/W	Description	Reset Value
SYS_CA35WRBADR0	SYS_BA+0x040	R/W	Cortex-A35 Core 0 Warm-boot Address Register	0x0000_0000

31	30	29	28	27	26	25	24
WRMBTADDR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
WRMBTADDR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
WRMBTADDR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WRMBTADDR							

Bits	Description	
[31:0]	**WRMBTADDR**	**Warm Boot Address** The WRMBTADDR indicates the warm boot run address for Cortex-A35 Core 0.

Cortex-A35 Core 0 Warm-boot Parameter Register (SYS_CA35WRBPAR0)

Register	Offset	R/W	Description	Reset Value
SYS_CA35WRBPAR0	SYS_BA+0x044	R/W	Cortex-A35 Core 0 Warm-boot Parameter Register	0x0000_0000

31	30	29	28	27	26	25	24
WRMBTPARA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
WRMBTPARA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
WRMBTPARA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WRMBTPARA							

Bits	Description	
[31:0]	**WRMBTPARA**	**Warm Boot Parameter** The WRMBTPARA indicates the warm boot parameters for Cortex-A35 Core 0.

Cortex-A35 Core 1 Warm-boot Address Register (SYS_CA35WRBADR1)

Register	Offset	R/W	Description	Reset Value
SYS_CA35WRBADR1	SYS_BA+0x048	R/W	Cortex-A35 Core 1 Warm-boot Address Register	0x0000_0000

31	30	29	28	27	26	25	24
WRMBTADDR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
WRMBTADDR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
WRMBTADDR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WRMBTADDR							

Bits	Description	
[31:0]	**WRMBTADDR**	**Warm Boot Address** The WRMBTADDR indicates the warm boot run address for Cortex-A35 Core 1.

Cortex-A35 Core 1 Warm-boot Parameter Register (SYS_CA35WRBPAR1)

Register	Offset	R/W	Description	Reset Value
SYS_CA35WRBPAR1	SYS_BA+0x04C	R/W	Cortex-A35 Core 1 Warm-boot Parameter Register	0x0000_0000

31	30	29	28	27	26	25	24
WRMBTPARA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
WRMBTPARA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
WRMBTPARA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WRMBTPARA							

Bits	Description	
[31:0]	**WRMBTPARA**	**Warm Boot Parameter** The WRMBTPARA indicates the warm boot parameters for Cortex-A35 Core 1.

USB PHY Miscellaneous Control Register (SYS_USBPMISCR)

Register	Offset	R/W	Description	Reset Value
SYS_USBPMISCR	SYS_BA+0x060	R/W	USB PHY Miscellaneous Control Register (TZNS)	0x7885_7885

31	30	29	28	27	26	25	24
RCALCODE1				RTUNESEL1	Reserved	PHY1CK12MSTB	PHY1HSTCKSTB
23	22	21	20	19	18	17	16
EFUSESEL1	Reserved				PHY1COMN	PHY1SUSPEND	PHY1POR
15	14	13	12	11	10	9	8
RCALCODE0				RTUNESEL0	PHY0DEVCKSTB	PHY0CK12MSTB	PHY0HSTCKSTB
7	6	5	4	3	2	1	0
EFUSESEL0	VBUSDGSEL			Reserved	PHY0COMN	PHY0SUSPEND	PHY0POR

Bits	Description	
[31:28]	**RCALCODE1**	**USB 2.0 Port 1 High-Speed PHY Internal Resistor Trim Code** If RTUNESEL1 = 1, RCALCODE1 provides the tuning code for high-speed source impedance directly. If RTUNESEL1 = 0, RCALCODE1 provides the tuning code for on-chip resistor within +/- 18% resistance tuning range. 0000 = +18% (236 Ω). 0001 = +15.6%. 0010 = +13.2%. 0011 = +10.8%. 0100 = +8.4%. 0101 = +6%. 0110 = +3.6%. 0111 = +1.2%. 1000 = -1.2%. 1001 = -3.6%. 1010 = -6%. 1011 = -8.4%. 1100 = -10.8%. 1101 = -13.2%. 1110 = -15.6%. 1111 = -18% (164 Ω).
[27]	**RTUNESEL1**	**USB 2.0 Port 1 High-speed PHY Source Impedance Tuning Method Selection** 0 = Internal digital calibration codes are used for tuning the high-speed source impedance. 1 = The RCALCODE1 value is used for tuning the high-speed source impedance.
[26]	**Reserved**	Reserved.
[25]	**PHY1CK12MSTB**	**USB 2.0 Port 1 High-speed PHY 12 MHz Clock Stable Flag** 0 = USB 2.0 port 1 high-speed PHY 12 MHz clock is not stable. 1 = USB 2.0 port 1 high-speed PHY 12 MHz clock is stable.

[24]	**PHY1HSTCKSTB**	**USB 2.0 Port 1 High-speed PHY 60 MHz UTMI Interface Clock for Host Stable Flag** 0 = USB 2.0 port 1 high-speed PHY UTMI Interface clock for Host is not stable. 1 = USB 2.0 port 1 high-speed PHY UTMI Interface clock for Host is stable.
[23]	**EFUSESEL1**	**USB 2.0 Port 1 High-speed PHY Resistor Calibration with External Resistor Control Bit** 0 = An external resistor (REXT) is needed and internal digital calibration code is based on REXT. 1 = The internal resistor method with RCALCODE1 resistance control up to +/- 18% Enabled, allowing the removal of the REXT resistor.
[22:19]	**Reserved**	Reserved.
[18]	**PHY1COMN**	**USB 2.0 Port 1 High-speed PHY Common Block Power-Down Control Bit** 0 = In Suspend or Sleep modes, the REFCLK_LOGIC, Bias and PLL blocks of USB 2.0 port 1 high-speed PHY remain powered. With this setting, the input reference clock must remain on and valid during suspend or sleep. 1 = In Suspend mode, the REFCLK_LOGIC, Bias and PLL blocks of USB 2.0 port 1 high-speed PHY are powered down. In Sleep mode, the Bias and PLL blocks of USB 2.0 port 1 high-speed PHY are powered down.
[17]	**PHY1SUSPEND**	**USB 2.0 Port 1 High-speed PHY Suspend Control Bit** 0 = USB 2.0 port 1 high-speed PHY is in Suspend mode. 1 = USB 2.0 port 1 high-speed PHY is in Normal operating mode.
[16]	**PHY1POR**	**USB 2.0 Port 1 High-speed PHY Power-On Reset Control Bit** 0 = All test registers and state machines in the USB 2.0 port 1 high-speed PHY are not in reset state. 1 = All test registers and state machines in the USB 2.0 port 1 high-speed PHY are in reset state.
[15:12]	**RCALCODE0**	**USB 2.0 Port 0 High-Speed PHY Internal Resistor Trim Code** If RTUNESEL0 = 1, RCALCODE0 provides the tuning code for high-speed source impedance directly. If RTUNESEL0 = 0, RCALCODE0 provides the tuning code for on-chip resistor within +/- 18% resistance tuning range. 0000 = +18% (236 Ω). 0001 = +15.6%. 0010 = +13.2%. 0011 = +10.8%. 0100 = +8.4%. 0101 = +6%. 0110 = +3.6%. 0111 = +1.2%. 1000 = -1.2%. 1001 = -3.6%. 1010 = -6%. 1011 = -8.4%. 1100 = -10.8%. 1101 = -13.2%. 1110 = -15.6%. 1111 = -18% (164 Ω).
[11]	**RTUNESEL0**	**USB 2.0 Port 0 High-speed PHY Source Impedance Tuning Method Selection** 0 = Internal digital calibration codes are used for tuning the high-speed source impedance. 1 = The RCALCODE0 value is used for tuning the high-speed source impedance.

[10]	**PHY0DEVCKSTB**	**USB 2.0 Port 0 High-speed PHY 60 MHz UTMI Interface Clock for Device Stable Flag** 0 = USB 2.0 port 0 high-speed PHY UTMI Interface clock for Device is not stable. 1 = USB 2.0 port 0 high-speed PHY UTMI Interface clock for Device is stable.
[9]	**PHY0CK12MSTB**	**USB 2.0 Port 0 High-speed PHY 12 MHz Clock Stable Flag** 0 = USB 2.0 port 0 high-speed PHY 12 MHz clock is not stable. 1 = USB 2.0 port 0 high-speed PHY 12 MHz clock is stable.
[8]	**PHY0HSTCKSTB**	**USB 2.0 Port 0 High-speed PHY 60 MHz UTMI Interface Clock for Host Stable Flag** 0 = USB 2.0 port 0 high-speed PHY UTMI Interface clock for Host is not stable. 1 = USB 2.0 port 0 high-speed PHY UTMI Interface clock for Host is stable.
[7]	**EFUSESEL0**	**USB 2.0 Port 0 High-speed PHY Resistor Calibration with External Resistor Control Bit** 0 = An external resistor (REXT) is needed and internal digital calibration code is based on REXT. 1 = The internal resistor method with RCALCODE0 resistance control up to +/- 18% Enabled, allowing the removal of the REXT resistor.
[6:4]	**VBUSDGSEL**	**V_{BUS} Detect De-glitch Time Select** 000 = Without de-glitch function. 001 = 4 HIRC clock. 010 = 8 HIRC clock. 011 = 16 HIRC clock. 100 = 32 HIRC clock. 101 = 64 HIRC clock. 110 = 128 HIRC clock. 111 = 256 HIRC clock.
[3]	**Reserved**	Reserved.
[2]	**PHY0COMN**	**USB 2.0 Port 0 High-speed PHY Common Block Power-Down Control Bit** 0 = In Suspend or Sleep modes, the REFCLK_LOGIC, Bias and PLL blocks of USB 2.0 port 0 high-speed PHY remain powered. With this setting, the input reference clock must remain on and valid during suspend or sleep. 1 = In Suspend mode, the REFCLK_LOGIC, Bias and PLL blocks of USB 2.0 port 0 high-speed PHY are powered down. In Sleep mode, the Bias and PLL blocks of USB 2.0 port 0 high-speed PHY are powered down.
[1]	**PHY0SUSPEND**	**USB 2.0 Port 0 High-speed PHY Suspend Control Bit** 0 = USB 2.0 port 0 high-speed PHY is in Suspend mode. 1 = USB 2.0 port 0 high-speed PHY is in Normal operating mode.
[0]	**PHY0POR**	**USB 2.0 Port 0 High-speed PHY Power-On Reset Control Bit** 0 = All test registers and state machines in the USB 2.0 port 0 high-speed PHY are not in reset state. 1 = All test registers and state machines in the USB 2.0 port 0 high-speed PHY are in reset state.

USB Port x PHY Control Register (SYS_USBPxPCR, x = 0, 1)

Register	Offset	R/W	Description	Reset Value
SYS_USBP0PCR	SYS_BA+0x064	R/W	USB Port 0 PHY Control Register	0x7344_3C31
SYS_USBP1PCR	SYS_BA+0x068	R/W	USB Port 1 PHY Control Register (TZNS)	0x7344_3C31

31	30	29	28	27	26	25	24
TXRESTUNE		TXHSXVTUNE		TXVREFTUNE			
23	22	21	20	19	18	17	16
TXRISETUNE		TXPREEMPAMPTUNE		Reserved		PLLITUNE	
15	14	13	12	11	10	9	8
TXFSLSTUNE				PLLPTUNE			
7	6	5	4	3	2	1	0
TXPREEMPPULSETUNE	SQRXTUNE			EQBYPASSENB	COMPDISTUNE		

Bits	Description	
[31:30]	**TXRESTUNE**	**USB 2.0 High-speed PHY USB Source Impedance Adjustment** 00 = Source Impedance is increased by approximately 3.03 Ω. 01 = 0 (Default). 10 = Source Impedance is increased by approximately 2.11 Ω. 11 = Source Impedance is increased by approximately 4.51 Ω.
[29:28]	**TXHSXVTUNE**	**USB 2.0 High-speed PHY Transmitter High-Speed Crossover Adjustment** The TXHSXVTUNE adjusts the voltage at which the DP and DM signals cross while transmitting in HS mode. 00 = Reserved. 01 = -9.16mV. 10 = +9.42mV. 11 = 0 (Default).

[27:24]	**TXVREFTUNE**	**USB 2.0 High-speed PHY HS DC Voltage Level Adjustment** The TXVREFTUNE adjusts the high-speed DC level voltage. 0000 = -9.37%. 0001 = -6.24%. 0010 = -3.12%. 0011 = 0 (Default). 0100 = +2.75%. 0101 = +5.87%. 0110 = +8.99%. 0111 = +12.11%. 1000 = +14.71%. 1001 = +17.82%. 1010 = +20.94%. 1011 = +24.06%. 1100 = +26.81%. 1101 = +29.94%. 1110 = +33.06%. 1111 = +36.18%.
[23:22]	**TXRISETUNE**	**USB 2.0 High-speed PHY HS Transmitter Rise/Fall Time Adjustment** The TXRISETUNE adjusts the rise/fall times of the high-speed waveform. 00 = +7.34%. 01 = 0 (Default). 10 = -5.98%. 11 = -7.49%.
[21:20]	**TXPREEMPAMPTUNE**	**USB 2.0 High-speed PHY HS Transmitter Pre-Emphasis Current Control** 00 = HS Transmitter pre-emphasis is disabled. (Default) 01 = HS Transmitter pre-emphasis circuit sources 1x pre-emphasis current. 10 = HS Transmitter pre-emphasis circuit sources 2x pre-emphasis current. 11 = HS Transmitter pre-emphasis circuit sources 3x pre-emphasis current.
[19:18]	**Reserved**	Reserved.
[17:16]	**PLLITUNE**	**USB 2.0 High-speed PHY Integral Path Tune** The value of PLLITUNE should be keep in default. 00 = 1.0x (Default). 01 = 2.0x. 10 = 3.0x. 11 = 4.0x.
[15:12]	**TXFSLSTUNE**	**USB 2.0 High-speed PHY FS/LS Source Impedance Adjustment** The TXFSLSTUNE adjusts the low- and full-speed single-ended source impedance while driving high. 0000 = +14.2%. 0001 = +6.60%. 0011 = 0 (Default). 0111 = -5.48%. 1111 = -10.29%.

[11:8]	**PLLPTUNE**	**USB 2.0 High-speed PHY PLL Proportional Path Tune** The value of PLLPTUNE should be keep in default. 0000 = 4.0x. 0001 = 4.5x. 0010 = 5.0x. 0011 = 5.5x. 0100 = 6.0x. 0101 = 6.5x. 0110 = 7.0x. 0111 = 7.5x. 1000 = 8.0x. 1001 = 8.5x. 1010 = 9.0x. 1011 = 9.5x. 1100 = 10.0x (Default). 1101 = 10.5x. 1110 = 11.0x. 1111 = 11.5x.
[7]	**TXPREEMPPULSETUNE**	**USB 2.0 High-speed PHY Squelch Equalizer Bypass Control Bit** 0 = Equalizer is Enabled. 1 = Equalizer is bypassed and acts as a simple differential input amplifier.
[6:4]	**SQRXTUNE**	**USB 2.0 High-speed PHY Squelch Threshold Adjustment** The SQRXTUNE adjusts the voltage level for the threshold used to detect valid high-speed data. 000 = +15.5%. 001 = +10.87%. 010 = +5.86%. 011 = 0 (Default). 100 = -5.86%. 101 = -13.33%. 110 = -21.56%. 111 = -31.54%.
[3]	**EQBYPASSENB**	**USB 2.0 High-speed PHY Squelch Equalizer Bypass Control Bit** 0 = Equalizer is Enabled. 1 = Equalizer is bypassed and acts as a simple differential input amplifier.
[2:0]	**COMPDISTUNE**	**USB 2.0 High-speed Disconnect Threshold Adjustment** The SQRXTUNE adjusts the voltage level for the threshold used to detect a disconnect event at the host. 000 = -5.48%. 001 = 0. (Default). 010 = +6.04%. 011 = +12.75%. 100 = +19.66%. 101 = +28.24%. 110 = +38.31%. 111 = +50.28%.

Miscellaneous Function Control Register 0 (SYS_MISCFCR0)

Register	Offset	R/W	Description	Reset Value
SYS_MISCFCR0	SYS_BA+0x070	R/W	Miscellaneous Function Control Register 0 (Shared)	0x2800_0230

31	30	29	28	27	26	25	24
DRATSRDLY							
23	22	21	20	19	18	17	16
DDRCGDIS	Reserved			LNSTWKEN	VBUSWKEN	SDH1VSTB	SDH0VSTB
15	14	13	12	11	10	9	8
WDT2RSTEN	WDT1RSTEN	SELFTEST	UHOVRCURH	Reserved	Reserved	HDSPUEN	WDT0RSTEN
7	6	5	4	3	2	1	0
Reserved					RTPDRMAEN	RTPDCACHEN	RTPICACHEN

Bits	Description	
[31:24]	**DRATSRDLY**	**DDR Auto Self Refresh Delay Count (TZNS)** This register uses to set the delay cycles of DDR memory controller before the core clock of DDR memory controller is gating. It allows for the self-refresh status to propagate to the APB domain so the STAT.selfref_type register field also reflects the status. It is sufficient to set this delay cycles to 31 cycles normally. In particular, if the AXI frequency is much less than the DDRC frequency, a higher value may be required in order to ensure that all read data is synchronized to the AXI domain before the clock is removed. **Note:** User can set this register value during core reset of DDR memory controller is asserted (i.e. MCTLCRST (SYS_IPRST0[29]) set to 1'b1).
[23]	**DDRCGDIS**	**DDR Controller Core Clock Gating Disable Bit (TZNS)** 0 = DDR controller core clock gating in auto self-refresh mode Enabled. 1 = DDR controller core clock gating in auto self-refresh mode Disabled. **Note:** This register needs to be set to 1'b1 to bypass clock gating function of DDR core clock before user writes/reads control registers or status registers of DDR memory controller. Therefore the signals in PCLK domain of DDR memory controller can be synchronous to core clock domain of DDR memory controller, and the signals in core clock domain of DDR memory controller can be synchronous to PCLK domain of DDR memory controller correctly. After user writes/reads control registers or status registers of DDR memory controller, this register can be set to 1'b0 to enable clock gating function of DDR core clock.
[22:20]	**Reserved**	Reserved.
[19]	**LNSTWKEN**	**HSUSBD Line State Wakeup Enable Control Bit (TZNS)** 0 = HSUSBD line state wakeup system from Power-down mode Disabled. 1 = HSUSBD line state wakeup system from Power-down mode Enabled.
[18]	**VBUSWKEN**	**HSUSBD V_{BUS} Detect Wakeup Enable Control Bit (TZNS)** 0 = HSUSBD V_{BUS} detect wakeup system from Power-down mode Disabled. 1 = HSUSBD V_{BUS} detect wakeup system from Power-down mode Enabled.
[17]	**SDH1VSTB**	**Voltage Stable Indicator to SDH 1 (TZNS)** Set this bit high to indicate SDH 1 that I/O voltage is stable. 0 = Voltage of I/O used as SDH 1 is not stable. 1 = Voltage of I/O used as SDH 1 is stable.

[16]	**SDH0VSTB**	**Voltage Stable Indicator to SDH 0 (TZNS)** Set this bit high to indicate SDH 0 that I/O voltage is stable. 0 = Voltage of I/O used as SDH 0 is not stable. 1 = Voltage of I/O used as SDH 0 is stable.
[15]	**WDT2RSTEN**	**WatchDog Timer 2 Reset Connection Enable Bit (TZNS)** This bit is use to enable the function that connect watch-dog timer 2 reset to nRESET pin. If this bit is enabled, the watch-dog timer 2 reset is connected to nRESET pin internally 0 = Watch-dog timer 2 reset not connected to nRESET pin internally. 1 = Watch-dog timer 2 reset connected to nRESET pin internally.
[14]	**WDT1RSTEN**	**WatchDog Timer 1 Reset Connection Enable Bit (TZNS)** This bit is use to enable the function that connect watch-dog timer 1 reset to nRESET pin. If this bit is enabled, the watch-dog timer 1 reset is connected to nRESET pin internally 0 = Watch-dog timer 1 reset not connected to nRESET pin internally. 1 = Watch-dog timer 1 reset connected to nRESET pin internally.
[13]	**SELFTEST**	**Self-test Mode Enable Bit (TZNS)** 0 = Self-Test mode Disabled. 1 = Self-Test mode Enabled.
[12]	**UHOVRCURH**	**USB Host Overcurrent Detection High Active (TZNS)** 0 = USB host overcurrent detection signal is low active. 1 = USB host overcurrent detection signal is high active.
[11:10]	**Reserved**	Reserved.
[9]	**HDSPUEN**	**HDS Pin Internal Pull-up Enable Bit (TZNS)** 0 = HDS pin internal pull-up resister Disabled. 1 = HDS pin internal pull-up resister Enabled.
[8]	**WDT0RSTEN**	**WatchDog Timer 0 Reset Connection Enable Bit** This bit is use to enable the function that connect watch-dog timer 0 reset to nRESET pin. If this bit is enabled, the watch-dog timer 0 reset is connected to nRESET pin internally 0 = Watch-dog timer 0 reset not connected to nRESET pin internally. 1 = Watch-dog timer 0 reset connected to nRESET pin internally.
[7:3]	**Reserved**	Reserved.
[2]	**RTPDRMAEN**	**Real-time Cortex-M4 Processor DRAM Access Enable Bit (SUBM)** 0 = Real-Time Cortex-M4 processor DRAM access Disabled. 1 = Real-Time Cortex-M4 processor DRAM access Enabled.
[1]	**RTPDCACHEN**	**Real-time Cortex-M4 Processor Data Cache Enable Bit (SUBM)** 0 = Real-Time Cortex-M4 processor data cache Disabled. 1 = Real-Time Cortex-M4 processor data cache Enabled.
[0]	**RTPICACHEN**	**Real-time Cortex-M4 Processor Instruction Cache Enable Bit (SUBM)** 0 = Real-Time Cortex-M4 processor instruction cache Disabled. 1 = Real-Time Cortex-M4 processor instruction cache Enabled.

Miscellaneous Function Control Register 1 (SYS_MISCFCR1)

The right of write 1 to bit 3~0 of this register controls by register SSPCC_PSSET10. If the security attribute of peripheral set to SUBM, its corresponding reset bit only can be written by RTP Cortex-M4 core. If the security attribute of peripheral set to TZNS, its corresponding reset bit only can be written by Cortex-A35 core in Non-Secure state. If the security attribute of peripheral set to TZS, its corresponding reset bit only can be written by Cortex-A35 core in Secure state.

Register	Offset	R/W	Description	Reset Value
SYS_MISCFCR1	SYS_BA+0x074	R/W	Miscellaneous Function Control Register 1 (Shared)	0x0073_0100

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved	DPICFG0EXT			Reserved		RMESYSRAM	RMEL1RAM
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						HXTDS	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CANFD3CKSTP	CANFD2CKSTP	CANFD1CKSTP	CANFD0CKSTP	CANFD3PDEN	CANFD2PDEN	CANFD1PDEN	CANFD0PDEN

Bits	Description	
[31:23]	**Reserved**	Reserved.
[22:20]	**DPICFG0EXT**	**DPI Data Format Configuration Extension** 000 = DPI 16 bit data format configuration 1 extension 0. 001 = Reserved. 010 = DPI 18 bit data format configuration 1 extension 0. 011 = Reserved. 100 = DPI 24 bit data format configuration extension 0. 101 = Reserved. 110 = Reserved. 111 = DPI data format configuration extension Disabled. **Note 1:** If DPICFG0EXT is 000, the DPI_DATA_FORMAT(dcregDpiConfig0 [2:0]) must be 000. **Note 2:** If DPICFG0EXT is 010, the DPI_DATA_FORMAT(dcregDpiConfig0 [2:0]) must be 011. **Note 3:** If DPICFG0EXT is 100, the DPI_DATA_FORMAT(dcregDpiConfig0 [2:0]) must be 101.
[19:18]	**Reserved**	Reserved.
[17]	**RMESYSRAM**	**System SRAM Macro RME Control Bit (Write Protect)** 0 = Default read-write margin of system SRAM selected. 1 = High speed read-write margin of system SRAM selected. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.

[16]	**RMEL1RAM**	**Cortex-A35 L1 Cache SRAM Macro RME Control Bit (Write Protect)** 0 = Default read-write margin of Cortex-A35 L1 and L2 cache SRAM selected. 1 = High speed read-write margin of Cortex-A35 L1 and L2 cache SRAM selected. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[15:10]	**Reserved**	Reserved.
[9:8]	**HXTDS**	**HXT Driving Current Selection (Write Protect)** 00 = Low power consumption mode for 2.5V~3.3V operating voltage. 01 = High noise immunity mode for 2.5V~3.3V operating voltage. 10 = Low power consumption mode for 1.8V~2.5V operating voltage. 11 = High noise immunity mode for 1.8V~2.5V operating voltage. **Note:** This bit is write protected. Refer to the SYS_RLKTZS, SYS_RLKTZNS or SYS_RLKSUBM register according to security attribute of each circuit.
[7]	**CANFD3CKSTP**	**CAN FD 3 Clock Stop Acknowledgement (Read Only)** 0 = CAN FD 3 clock didn't stop. 1 = CAN FD 3 clock stop.
[6]	**CANFD2CKSTP**	**CAN FD 2 Clock Stop Acknowledgement (Read Only)** 0 = CAN FD 2 clock didn't stop. 1 = CAN FD 2 clock stop.
[5]	**CANFD1CKSTP**	**CAN FD 1 Clock Stop Acknowledgement (Read Only)** 0 = CAN FD 1 clock didn't stop. 1 = CAN FD 1 clock stop.
[4]	**CANFD0CKSTP**	**CAN FD 0 Clock Stop Acknowledgement (Read Only)** 0 = CAN FD 0 clock didn't stop. 1 = CAN FD 0 clock stop.
[3]	**CANFD3PDEN**	**CAN FD 3 Power Down Enable Bit** 0 = CAN FD 3 Power-down mode Disabled. 1 = CAN FD 3 Power-down mode Enabled.
[2]	**CANFD2PDEN**	**CAN FD 2 Power Down Enable Bit** 0 = CAN FD 2 Power-down mode Disabled. 1 = CAN FD 2 Power-down mode Enabled.
[1]	**CANFD1PDEN**	**CAN FD 1 Power Down Enable Bit** 0 = CAN FD 1 Power-down mode Disabled. 1 = CAN FD 1 Power-down mode Enabled.
[0]	**0PDEN**	**CAN FD 0 Power Down Enable Bit** 0 = CAN FD 0 Power-down mode Disabled. 1 = CAN FD 0 Power-down mode Enabled.

Miscellaneous Interrupt Enable Register (SYS_MISCIER)

Register	Offset	R/W	Description	Reset Value
SYS_MISCIER	SYS_BA+0x078	R/W	Miscellaneous Interrupt Enable Register (TZNS)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**VBUSCHGIEN**	**USB0IDCHGIEN**	**LVDIEN**

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**VBUSCHGIEN**	**USUSB0_VBUSVLD Pin Status Change Interrupt Enable Bit** 0 = USUSB0_VBUSVLD pin status change interrupt Disabled. 1 = USUSB0_VBUSVLD pin status change interrupt Enabled.
[1]	**USB0IDCHGIEN**	**USB0_ID Pin Status Change Interrupt Enable Bit** 0 = USB0_ID pin status change interrupt Disabled. 1 = USB0_ID pin status change interrupt Enabled.
[0]	**LVDIEN**	**Low Voltage Detect Interrupt Enable Bit** 0 = Low voltage detect interrupt Disabled. 1 = Low voltage detect interrupt Enabled.

Miscellaneous Interrupt Status Register (SYS_MISCISR)

Register	Offset	R/W	Description	Reset Value
SYS_MISCISR	SYS_BA+0x07C	R/W	Miscellaneous Interrupt Status Register (TZNS)	0x0001_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved					**VBUSSTS**	**USB0IDSTS**	**LVDSTS**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**VBUSCHGIF**	**USB0IDCHGIF**	**LVDIF**

Bits	Description	
[31:19]	**Reserved**	Reserved.
[18]	**VBUSSTS**	**V_{BUS} Detect Pin State** 0 = V_{BUS} detect pin state is low. 1 = V_{BUS} detect pin state is high.
[17]	**USB0IDSTS**	**USB0_ID Pin State** 0 = USB port 0 used as a USB device port. 1 = USB port 0 used as a USB host port.
[16]	**LVDSTS**	**Low Voltage Detect State** 0 = Low voltage detect state is low. 1 = Low voltage detect state is high.
[15:3]	**Reserved**	Reserved.
[2]	**VBUSCHGIF**	**USUSB0_VBUSVLD Pin State Change Interrupt Flag** 0 = USUSB0_VBUSVLD pin state didn't change. 1 = USUSB0_VBUSVLD pin state changed from low to high or from high to low.
[1]	**USB0IDCHGIF**	**USB0_ID Pin State Change Interrupt Flag** 0 = USB0_ID state didn't change. 1 = USB0_ID state changed from low to high or from high to low.
[0]	**LVDIF**	**Low Voltage Detect Interrupt Flag** 0 = No low voltage event. 1 = Low voltage event detected.

GPIOA Low Byte Multiple Function Control Register (SYS_GPA_MFPL)

Register	Offset	R/W	Description	Reset Value
SYS_GPA_MFPL	SYS_BA+0x080	R/W	GPIOA Low Byte Multiple Function Control Register	0x0XXX_XX00

31	30	29	28	27	26	25	24
MFPGPA7				MFPGPA6			
23	22	21	20	19	18	17	16
MFPGPA5				MFPGPA4			
15	14	13	12	11	10	9	8
MFPGPA3				MFPGPA2			
7	6	5	4	3	2	1	0
MFPGPA1				MFPGPA0			

Bits	Description	
[31:28]	**MFPGPA7**	**Pin PA.7 Multi-function Pin Selection**
[27:24]	**MFPGPA6**	**Pin PA.6 Multi-function Pin Selection**
[23:20]	**MFPGPA5**	**Pin PA.5 Multi-function Pin Selection**
[19:16]	**MFPGPA4**	**Pin PA.4 Multi-function Pin Selection**
[15:12]	**MFPGPA3**	**Pin PA.3 Multi-function Pin Selection**
[11:8]	**MFPGPA2**	**Pin PA.2 Multi-function Pin Selection**
[7:4]	**MFPGPA1**	**Pin PA.1 Multi-function Pin Selection**
[3:0]	**MFPGPA0**	**Pin PA.0 Multi-function Pin Selection**

GPIOA High Byte Multiple Function Control Register (SYS_GPA_MFPH)

Register	Offset	R/W	Description	Reset Value
SYS_GPA_MFPH	SYS_BA+0x084	R/W	GPIOA High Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MFPGPA15				MFPGPA14			
23	22	21	20	19	18	17	16
MFPGPA13				MFPGPA12			
15	14	13	12	11	10	9	8
MFPGPA11				MFPGPA10			
7	6	5	4	3	2	1	0
MFPGPA9				MFPGPA8			

Bits	Description	
[31:28]	**MFPGPA15**	**Pin PA.15 Multi-function Pin Selection**
[27:24]	**MFPGPA14**	**Pin PA.14 Multi-function Pin Selection**
[23:20]	**MFPGPA13**	**Pin PA.13 Multi-function Pin Selection**
[19:16]	**MFPGPA12**	**Pin PA.12 Multi-function Pin Selection**
[15:12]	**MFPGPA11**	**Pin PA.11 Multi-function Pin Selection**
[11:8]	**MFPGPA10**	**Pin PA.10 Multi-function Pin Selection**
[7:4]	**MFPGPA9**	**Pin PA.9 Multi-function Pin Selection**
[3:0]	**MFPGPA8**	**Pin PA.8 Multi-function Pin Selection**

GPIOB Low Byte Multiple Function Control Register (SYS_GPB_MFPL)

Register	Offset	R/W	Description	Reset Value
SYS_GPB_MFPL	SYS_BA+0x088	R/W	GPIOB Low Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MFPGPB7				MFPGPB6			
23	22	21	20	19	18	17	16
MFPGPB5				MFPGPB4			
15	14	13	12	11	10	9	8
MFPGPB3				MFPGPB2			
7	6	5	4	3	2	1	0
MFPGPB1				MFPGPB0			

Bits	Description	
[31:28]	**MFPGPB7**	**Pin PB.7 Multi-function Pin Selection**
[27:24]	**MFPGPB6**	**Pin PB.6 Multi-function Pin Selection**
[23:20]	**MFPGPB5**	**Pin PB.5 Multi-function Pin Selection**
[19:16]	**MFPGPB4**	**Pin PB.4 Multi-function Pin Selection**
[15:12]	**MFPGPB3**	**Pin PB.3 Multi-function Pin Selection**
[11:8]	**MFPGPB2**	**Pin PB.2 Multi-function Pin Selection**
[7:4]	**MFPGPB1**	**Pin PB.1 Multi-function Pin Selection**
[3:0]	**MFPGPB0**	**Pin PB.0 Multi-function Pin Selection**

GPIOB High Byte Multiple Function Control Register (SYS_GPB_MFPH)

Register	Offset	R/W	Description	Reset Value
SYS_GPB_MFPH	SYS_BA+0x08C	R/W	GPIOB High Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MFPGPB15				MFPGPB14			
23	22	21	20	19	18	17	16
MFPGPB13				MFPGPB12			
15	14	13	12	11	10	9	8
MFPGPB11				MFPGPB10			
7	6	5	4	3	2	1	0
MFPGPB9				MFPGPB8			

Bits	Description	
[31:28]	**MFPGPB15**	**Pin PB.15 Multi-function Pin Selection**
[27:24]	**MFPGPB14**	**Pin PB.14 Multi-function Pin Selection**
[23:20]	**MFPGPB13**	**Pin PB.13 Multi-function Pin Selection**
[19:16]	**MFPGPB12**	**Pin PB.12 Multi-function Pin Selection**
[15:12]	**MFPGPB11**	**Pin PB.11 Multi-function Pin Selection**
[11:8]	**MFPGPB10**	**Pin PB.10 Multi-function Pin Selection**
[7:4]	**MFPGPB9**	**Pin PB.9 Multi-function Pin Selection**
[3:0]	**MFPGPB8**	**Pin PB.8 Multi-function Pin Selection**

GPIOC Low Byte Multiple Function Control Register (SYS_GPC_MFPL)

Register	Offset	R/W	Description	Reset Value
SYS_GPC_MFPL	SYS_BA+0x090	R/W	GPIOC Low Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MFPGPC7				MFPGPC6			
23	22	21	20	19	18	17	16
MFPGPC5				MFPGPC4			
15	14	13	12	11	10	9	8
MFPGPC3				MFPGPC2			
7	6	5	4	3	2	1	0
MFPGPC1				MFPGPC0			

Bits	Description	
[31:28]	**MFPGPC7**	**Pin PC.7 Multi-function Pin Selection**
[27:24]	**MFPGPC6**	**Pin PC.6 Multi-function Pin Selection**
[23:20]	**MFPGPC5**	**Pin PC.5 Multi-function Pin Selection**
[19:16]	**MFPGPC4**	**Pin PC.4 Multi-function Pin Selection**
[15:12]	**MFPGPC3**	**Pin PC.3 Multi-function Pin Selection**
[11:8]	**MFPGPC2**	**Pin PC.2 Multi-function Pin Selection**
[7:4]	**MFPGPC1**	**Pin PC.1 Multi-function Pin Selection**
[3:0]	**MFPGPC0**	**Pin PC.0 Multi-function Pin Selection**

GPIOC High Byte Multiple Function Control Register (SYS_GPC_MFPH)

Register	Offset	R/W	Description	Reset Value
SYS_GPC_MFPH	SYS_BA+0x094	R/W	GPIOC High Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MFPGPC15				MFPGPC14			
23	22	21	20	19	18	17	16
MFPGPC13				MFPGPC12			
15	14	13	12	11	10	9	8
MFPGPC11				MFPGPC10			
7	6	5	4	3	2	1	0
MFPGPC9				MFPGPC8			

Bits	Description	
[31:28]	**MFPGPC15**	**Pin PC.15 Multi-function Pin Selection**
[27:24]	**MFPGPC14**	**Pin PC.14 Multi-function Pin Selection**
[23:20]	**MFPGPC13**	**Pin PC.13 Multi-function Pin Selection**
[19:16]	**MFPGPC12**	**Pin PC.12 Multi-function Pin Selection**
[15:12]	**MFPGPC11**	**Pin PC.11 Multi-function Pin Selection**
[11:8]	**MFPGPC10**	**Pin PC.10 Multi-function Pin Selection**
[7:4]	**MFPGPC9**	**Pin PC.9 Multi-function Pin Selection**
[3:0]	**MFPGPC8**	**Pin PC.8 Multi-function Pin Selection**

GPIOD Low Byte Multiple Function Control Register (SYS_GPD_MFPL)

Register	Offset	R/W	Description	Reset Value
SYS_GPD_MFPL	SYS_BA+0x098	R/W	GPIOD Low Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MFPGPD7				MFPGPD6			
23	22	21	20	19	18	17	16
MFPGPD5				MFPGPD4			
15	14	13	12	11	10	9	8
MFPGPD3				MFPGPD2			
7	6	5	4	3	2	1	0
MFPGPD1				MFPGPD0			

Bits	Description	
[31:28]	**MFPGPD7**	**Pin PD.7 Multi-function Pin Selection**
[27:24]	**MFPGPD6**	**Pin PD.6 Multi-function Pin Selection**
[23:20]	**MFPGPD5**	**Pin PD.5 Multi-function Pin Selection**
[19:16]	**MFPGPD4**	**Pin PD.4 Multi-function Pin Selection**
[15:12]	**MFPGPD3**	**Pin PD.3 Multi-function Pin Selection**
[11:8]	**MFPGPD2**	**Pin PD.2 Multi-function Pin Selection**
[7:4]	**MFPGPD1**	**Pin PD.1 Multi-function Pin Selection**
[3:0]	**MFPGPD0**	**Pin PD.0 Multi-function Pin Selection**

GPIOD High Byte Multiple Function Control Register (SYS_GPD_MFPH)

Register	Offset	R/W	Description	Reset Value
SYS_GPD_MFPH	SYS_BA+0x09C	R/W	GPIOD High Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MFPGPD15				MFPGPD14			
23	22	21	20	19	18	17	16
MFPGPD13				MFPGPD12			
15	14	13	12	11	10	9	8
MFPGPD11				MFPGPD10			
7	6	5	4	3	2	1	0
MFPGPD9				MFPGPD8			

Bits	Description	
[31:28]	**MFPGPD15**	**Pin PD.15 Multi-function Pin Selection**
[27:24]	**MFPGPD14**	**Pin PD.14 Multi-function Pin Selection**
[23:20]	**MFPGPD13**	**Pin PD.13 Multi-function Pin Selection**
[19:16]	**MFPGPD12**	**Pin PD.12 Multi-function Pin Selection**
[15:12]	**MFPGPD11**	**Pin PD.11 Multi-function Pin Selection**
[11:8]	**MFPGPD10**	**Pin PD.10 Multi-function Pin Selection**
[7:4]	**MFPGPD9**	**Pin PD.9 Multi-function Pin Selection**
[3:0]	**MFPGPD8**	**Pin PD.8 Multi-function Pin Selection**

GPIOE Low Byte Multiple Function Control Register (SYS_GPE_MFPL)

Register	Offset	R/W	Description	Reset Value
SYS_GPE_MFPL	SYS_BA+0x0A0	R/W	GPIOE Low Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MFPGPE7				MFPGPE6			
23	22	21	20	19	18	17	16
MFPGPE5				MFPGPE4			
15	14	13	12	11	10	9	8
MFPGPE3				MFPGPE2			
7	6	5	4	3	2	1	0
MFPGPE1				MFPGPE0			

Bits	Description	
[31:28]	**MFPGPE7**	**Pin PE.7 Multi-function Pin Selection**
[27:24]	**MFPGPE6**	**Pin PE.6 Multi-function Pin Selection**
[23:20]	**MFPGPE5**	**Pin PE.5 Multi-function Pin Selection**
[19:16]	**MFPGPE4**	**Pin PE.4 Multi-function Pin Selection**
[15:12]	**MFPGPE3**	**Pin PE.3 Multi-function Pin Selection**
[11:8]	**MFPGPE2**	**Pin PE.2 Multi-function Pin Selection**
[7:4]	**MFPGPE1**	**Pin PE.1 Multi-function Pin Selection**
[3:0]	**MFPGPE0**	**Pin PE.0 Multi-function Pin Selection**

GPIOE High Byte Multiple Function Control Register (SYS_GPE_MFPH)

Register	Offset	R/W	Description	Reset Value
SYS_GPE_MFPH	SYS_BA+0x0A4	R/W	GPIOE High Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MFPGPE15				MFPGPE14			
23	22	21	20	19	18	17	16
MFPGPE13				MFPGPE12			
15	14	13	12	11	10	9	8
MFPGPE11				MFPGPE10			
7	6	5	4	3	2	1	0
MFPGPE9				MFPGPE8			

Bits	Description	
[31:28]	**MFPGPE15**	**Pin PE.15 Multi-function Pin Selection**
[27:24]	**MFPGPE14**	**Pin PE.14 Multi-function Pin Selection**
[23:20]	**MFPGPE13**	**Pin PE.13 Multi-function Pin Selection**
[19:16]	**MFPGPE12**	**Pin PE.12 Multi-function Pin Selection**
[15:12]	**MFPGPE11**	**Pin PE.11 Multi-function Pin Selection**
[11:8]	**MFPGPE10**	**Pin PE.10 Multi-function Pin Selection**
[7:4]	**MFPGPE9**	**Pin PE.9 Multi-function Pin Selection**
[3:0]	**MFPGPE8**	**Pin PE.8 Multi-function Pin Selection**

GPIOF Low Byte Multiple Function Control Register (SYS_GPF_MFPL)

Register	Offset	R/W	Description	Reset Value
SYS_GPF_MFPL	SYS_BA+0x0A8	R/W	GPIOF Low Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MFPGPF7				MFPGPF6			
23	22	21	20	19	18	17	16
MFPGPF5				MFPGPF4			
15	14	13	12	11	10	9	8
MFPGPF3				MFPGPF2			
7	6	5	4	3	2	1	0
MFPGPF1				MFPGPF0			

Bits	Description	
[31:28]	**MFPGPF7**	**Pin PF.7 Multi-function Pin Selection**
[27:24]	**MFPGPF6**	**Pin PF.6 Multi-function Pin Selection**
[23:20]	**MFPGPF5**	**Pin PF.5 Multi-function Pin Selection**
[19:16]	**MFPGPF4**	**Pin PF.4 Multi-function Pin Selection**
[15:12]	**MFPGPF3**	**Pin PF.3 Multi-function Pin Selection**
[11:8]	**MFPGPF2**	**Pin PF.2 Multi-function Pin Selection**
[7:4]	**MFPGPF1**	**Pin PF.1 Multi-function Pin Selection**
[3:0]	**MFPGPF0**	**Pin PF.0 Multi-function Pin Selection**

GPIOF High Byte Multiple Function Control Register (SYS_GPF_MFPH)

Register	Offset	R/W	Description	Reset Value
SYS_GPF_MFPH	SYS_BA+0x0AC	R/W	GPIOF High Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MFPGPF15				MFPGPF14			
23	22	21	20	19	18	17	16
MFPGPF13				MFPGPF12			
15	14	13	12	11	10	9	8
MFPGPF11				MFPGPF10			
7	6	5	4	3	2	1	0
MFPGPF9				MFPGPF8			

Bits	Description	
[31:28]	**MFPGPF15**	**Pin PF.15 Multi-function Pin Selection**
[27:24]	**MFPGPF14**	**Pin PF.14 Multi-function Pin Selection**
[23:20]	**MFPGPF13**	**Pin PF.13 Multi-function Pin Selection**
[19:16]	**MFPGPF12**	**Pin PF.12 Multi-function Pin Selection**
[15:12]	**MFPGPF11**	**Pin PF.11 Multi-function Pin Selection**
[11:8]	**MFPGPF10**	**Pin PF.10 Multi-function Pin Selection**
[7:4]	**MFPGPF9**	**Pin PF.9 Multi-function Pin Selection**
[3:0]	**MFPGPF8**	**Pin PF.8 Multi-function Pin Selection**

GPIOG Low Byte Multiple Function Control Register (SYS_GPG_MFPL)

Register	Offset	R/W	Description	Reset Value
SYS_GPG_MFPL	SYS_BA+0x0B0	R/W	GPIOG Low Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MFPGPG7				MFPGPG6			
23	22	21	20	19	18	17	16
MFPGPG5				MFPGPG4			
15	14	13	12	11	10	9	8
MFPGPG3				MFPGPG2			
7	6	5	4	3	2	1	0
MFPGPG1				MFPGPG0			

Bits	Description	
[31:28]	**MFPGPG7**	**Pin PG.7 Multi-function Pin Selection**
[27:24]	**MFPGPG6**	**Pin PG.6 Multi-function Pin Selection**
[23:20]	**MFPGPG5**	**Pin PG.5 Multi-function Pin Selection**
[19:16]	**MFPGPG4**	**Pin PG.4 Multi-function Pin Selection**
[15:12]	**MFPGPG3**	**Pin PG.3 Multi-function Pin Selection**
[11:8]	**MFPGPG2**	**Pin PG.2 Multi-function Pin Selection**
[7:4]	**MFPGPG1**	**Pin PG.1 Multi-function Pin Selection**
[3:0]	**MFPGPG0**	**Pin PG.0 Multi-function Pin Selection**

GPIOG High Byte Multiple Function Control Register (SYS_GPG_MFPH)

Register	Offset	R/W	Description	Reset Value
SYS_GPG_MFPH	SYS_BA+0x0B4	R/W	GPIOG High Byte Multiple Function Control Register	0xXXXX_X000

31	30	29	28	27	26	25	24
MFPGPG15				MFPGPG14			
23	22	21	20	19	18	17	16
MFPGPG13				MFPGPG12			
15	14	13	12	11	10	9	8
MFPGPG11				MFPGPG10			
7	6	5	4	3	2	1	0
MFPGPG9				MFPGPG8			

Bits	Description	
[31:28]	**MFPGPG15**	**Pin PG.15 Multi-function Pin Selection**
[27:24]	**MFPGPG14**	**Pin PG.14 Multi-function Pin Selection**
[23:20]	**MFPGPG13**	**Pin PG.13 Multi-function Pin Selection**
[19:16]	**MFPGPG12**	**Pin PG.12 Multi-function Pin Selection**
[15:12]	**MFPGPG11**	**Pin PG.11 Multi-function Pin Selection**
[11:8]	**MFPGPG10**	**Pin PG.10 Multi-function Pin Selection**
[7:4]	**MFPGPG9**	**Pin PG.9 Multi-function Pin Selection**
[3:0]	**MFPGPG8**	**Pin PG.8 Multi-function Pin Selection**

GPIOH Low Byte Multiple Function Control Register (SYS_GPH_MFPL)

Register	Offset	R/W	Description	Reset Value
SYS_GPH_MFPL	SYS_BA+0x0B8	R/W	GPIOH Low Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MFPGPH7				MFPGPH6			
23	22	21	20	19	18	17	16
MFPGPH5				MFPGPH4			
15	14	13	12	11	10	9	8
MFPGPH3				MFPGPH2			
7	6	5	4	3	2	1	0
MFPGPH1				MFPGPH0			

Bits	Description	
[31:28]	**MFPGPH7**	**Pin PH.7 Multi-function Pin Selection**
[27:24]	**MFPGPH6**	**Pin PH.6 Multi-function Pin Selection**
[23:20]	**MFPGPH5**	**Pin PH.5 Multi-function Pin Selection**
[19:16]	**MFPGPH4**	**Pin PH.4 Multi-function Pin Selection**
[15:12]	**MFPGPH3**	**Pin PH.3 Multi-function Pin Selection**
[11:8]	**MFPGPH2**	**Pin PH.2 Multi-function Pin Selection**
[7:4]	**MFPGPH1**	**Pin PH.1 Multi-function Pin Selection**
[3:0]	**MFPGPH0**	**Pin PH.0 Multi-function Pin Selection**

GPIOH High Byte Multiple Function Control Register (SYS_GPH_MFPH)

Register	Offset	R/W	Description	Reset Value
SYS_GPH_MFPH	SYS_BA+0x0BC	R/W	GPIOH High Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MFPGPH15				MFPGPH14			
23	22	21	20	19	18	17	16
MFPGPH13				MFPGPH12			
15	14	13	12	11	10	9	8
MFPGPH11				MFPGPH10			
7	6	5	4	3	2	1	0
MFPGPH9				MFPGPH8			

Bits	Description	
[31:28]	**MFPGPH15**	**Pin PH.15 Multi-function Pin Selection**
[27:24]	**MFPGPH14**	**Pin PH.14 Multi-function Pin Selection**
[23:20]	**MFPGPH13**	**Pin PH.13 Multi-function Pin Selection**
[19:16]	**MFPGPH12**	**Pin PH.12 Multi-function Pin Selection**
[15:12]	**MFPGPH11**	**Pin PH.11 Multi-function Pin Selection**
[11:8]	**MFPGPH10**	**Pin PH.10 Multi-function Pin Selection**
[7:4]	**MFPGPH9**	**Pin PH.9 Multi-function Pin Selection**
[3:0]	**MFPGPH8**	**Pin PH.8 Multi-function Pin Selection**

GPIOI Low Byte Multiple Function Control Register (SYS_GPI_MFPL)

Register	Offset	R/W	Description	Reset Value
SYS_GPI_MFPL	SYS_BA+0x0C0	R/W	GPIOI Low Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MFPGPI7				MFPGPI6			
23	22	21	20	19	18	17	16
MFPGPI5				MFPGPI4			
15	14	13	12	11	10	9	8
MFPGPI3				MFPGPI2			
7	6	5	4	3	2	1	0
MFPGPI1				MFPGPI0			

Bits	Description	
[31:28]	**MFPGPI7**	**Pin PI.7 Multi-function Pin Selection**
[27:24]	**MFPGPI6**	**Pin PI.6 Multi-function Pin Selection**
[23:20]	**MFPGPI5**	**Pin PI.5 Multi-function Pin Selection**
[19:16]	**MFPGPI4**	**Pin PI.4 Multi-function Pin Selection**
[15:12]	**MFPGPI3**	**Pin PI.3 Multi-function Pin Selection**
[11:8]	**MFPGPI2**	**Pin PI.2 Multi-function Pin Selection**
[7:4]	**MFPGPI1**	**Pin PI.1 Multi-function Pin Selection**
[3:0]	**MFPGPI0**	**Pin PI.0 Multi-function Pin Selection**

GPIOI High Byte Multiple Function Control Register (SYS_GPI_MFPH)

Register	Offset	R/W	Description	Reset Value
SYS_GPI_MFPH	SYS_BA+0x0C4	R/W	GPIOI High Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MFPGPI15				MFPGPI14			
23	22	21	20	19	18	17	16
MFPGPI13				MFPGPI12			
15	14	13	12	11	10	9	8
MFPGPI11				MFPGPI10			
7	6	5	4	3	2	1	0
MFPGPI9				MFPGPI8			

Bits	Description	
[31:28]	**MFPGPI15**	**Pin PI.15 Multi-function Pin Selection**
[27:24]	**MFPGPI14**	**Pin PI.14 Multi-function Pin Selection**
[23:20]	**MFPGPI13**	**Pin PI.13 Multi-function Pin Selection**
[19:16]	**MFPGPI12**	**Pin PI.12 Multi-function Pin Selection**
[15:12]	**MFPGPI11**	**Pin PI.11 Multi-function Pin Selection**
[11:8]	**MFPGPI10**	**Pin PI.10 Multi-function Pin Selection**
[7:4]	**MFPGPI9**	**Pin PI.9 Multi-function Pin Selection**
[3:0]	**MFPGPI8**	**Pin PI.8 Multi-function Pin Selection**

GPIOJ Low Byte Multiple Function Control Register (SYS_GPJ_MFPL)

Register	Offset	R/W	Description	Reset Value
SYS_GPJ_MFPL	SYS_BA+0x0C8	R/W	GPIOJ Low Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MFPGPJ7				MFPGPJ6			
23	22	21	20	19	18	17	16
MFPGPJ5				MFPGPJ4			
15	14	13	12	11	10	9	8
MFPGPJ3				MFPGPJ2			
7	6	5	4	3	2	1	0
MFPGPJ1				MFPGPJ0			

Bits	Description	
[31:28]	**MFPGPJ7**	**Pin PJ.7 Multi-function Pin Selection**
[27:24]	**MFPGPJ6**	**Pin PJ.6 Multi-function Pin Selection**
[23:20]	**MFPGPJ5**	**Pin PJ.5 Multi-function Pin Selection**
[19:16]	**MFPGPJ4**	**Pin PJ.4 Multi-function Pin Selection**
[15:12]	**MFPGPJ3**	**Pin PJ.3 Multi-function Pin Selection**
[11:8]	**MFPGPJ2**	**Pin PJ.2 Multi-function Pin Selection**
[7:4]	**MFPGPJ1**	**Pin PJ.1 Multi-function Pin Selection**
[3:0]	**MFPGPJ0**	**Pin PJ.0 Multi-function Pin Selection**

GPIOJ High Byte Multiple Function Control Register (SYS_GPJ_MFPH)

Register	Offset	R/W	Description	Reset Value
SYS_GPJ_MFPH	SYS_BA+0x0CC	R/W	GPIOJ High Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MFPGPJ15				MFPGPJ14			
23	22	21	20	19	18	17	16
MFPGPJ13				MFPGPJ12			
15	14	13	12	11	10	9	8
MFPGPJ11				MFPGPJ10			
7	6	5	4	3	2	1	0
MFPGPJ9				MFPGPJ8			

Bits	Description	
[31:28]	**MFPGPJ15**	**Pin PJ.15 Multi-function Pin Selection**
[27:24]	**MFPGPJ14**	**Pin PJ.14 Multi-function Pin Selection**
[23:20]	**MFPGPJ13**	**Pin PJ.13 Multi-function Pin Selection**
[19:16]	**MFPGPJ12**	**Pin PJ.12 Multi-function Pin Selection**
[15:12]	**MFPGPJ11**	**Pin PJ.11 Multi-function Pin Selection**
[11:8]	**MFPGPJ10**	**Pin PJ.10 Multi-function Pin Selection**
[7:4]	**MFPGPJ9**	**Pin PJ.9 Multi-function Pin Selection**
[3:0]	**MFPGPJ8**	**Pin PJ.8 Multi-function Pin Selection**

GPIOK Low Byte Multiple Function Control Register (SYS_GPK_MFPL)

Register	Offset	R/W	Description	Reset Value
SYS_GPK_MFPL	SYS_BA+0x0D0	R/W	GPIOK Low Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MFPGPK7				MFPGPK6			
23	22	21	20	19	18	17	16
MFPGPK5				MFPGPK4			
15	14	13	12	11	10	9	8
MFPGPK3				MFPGPK2			
7	6	5	4	3	2	1	0
MFPGPK1				MFPGPK0			

Bits	Description	
[31:28]	**MFPGPK7**	**Pin PK.7 Multi-function Pin Selection**
[27:24]	**MFPGPK6**	**Pin PK.6 Multi-function Pin Selection**
[23:20]	**MFPGPK5**	**Pin PK.5 Multi-function Pin Selection**
[19:16]	**MFPGPK4**	**Pin PK.4 Multi-function Pin Selection**
[15:12]	**MFPGPK3**	**Pin PK.3 Multi-function Pin Selection**
[11:8]	**MFPGPK2**	**Pin PK.2 Multi-function Pin Selection**
[7:4]	**MFPGPK1**	**Pin PK.1 Multi-function Pin Selection**
[3:0]	**MFPGPK0**	**Pin PK.0 Multi-function Pin Selection**

GPIOK High Byte Multiple Function Control Register (SYS_GPK_MFPH)

Register	Offset	R/W	Description	Reset Value
SYS_GPK_MFPH	SYS_BA+0x0D4	R/W	GPIOK High Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MFPGPK15				MFPGPK14			
23	22	21	20	19	18	17	16
MFPGPK13				MFPGPK12			
15	14	13	12	11	10	9	8
MFPGPK11				MFPGPK10			
7	6	5	4	3	2	1	0
MFPGPK9				MFPGPK8			

Bits	Description	
[31:28]	**MFPGPK15**	**Pin PK.15 Multi-function Pin Selection**
[27:24]	**MFPGPK14**	**Pin PK.14 Multi-function Pin Selection**
[23:20]	**MFPGPK13**	**Pin PK.13 Multi-function Pin Selection**
[19:16]	**MFPGPK12**	**Pin PK.12 Multi-function Pin Selection**
[15:12]	**MFPGPK11**	**Pin PK.11 Multi-function Pin Selection**
[11:8]	**MFPGPK10**	**Pin PK.10 Multi-function Pin Selection**
[7:4]	**MFPGPK9**	**Pin PK.9 Multi-function Pin Selection**
[3:0]	**MFPGPK8**	**Pin PK.8 Multi-function Pin Selection**

GPIOL Low Byte Multiple Function Control Register (SYS_GPL_MFPL)

Register	Offset	R/W	Description	Reset Value
SYS_GPL_MFPL	SYS_BA+0x0D8	R/W	GPIOL Low Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MFPGPL7				MFPGPL6			
23	22	21	20	19	18	17	16
MFPGPL5				MFPGPL4			
15	14	13	12	11	10	9	8
MFPGPL3				MFPGPL2			
7	6	5	4	3	2	1	0
MFPGPL1				MFPGPL0			

Bits	Description	
[31:28]	**MFPGPL7**	**Pin PL.7 Multi-function Pin Selection**
[27:24]	**MFPGPL6**	**Pin PL.6 Multi-function Pin Selection**
[23:20]	**MFPGPL5**	**Pin PL.5 Multi-function Pin Selection**
[19:16]	**MFPGPL4**	**Pin PL.4 Multi-function Pin Selection**
[15:12]	**MFPGPL3**	**Pin PL.3 Multi-function Pin Selection**
[11:8]	**MFPGPL2**	**Pin PL.2 Multi-function Pin Selection**
[7:4]	**MFPGPL1**	**Pin PL.1 Multi-function Pin Selection**
[3:0]	**MFPGPL0**	**Pin PL.0 Multi-function Pin Selection**

GPIOL High Byte Multiple Function Control Register (SYS_GPL_MFPH)

Register	Offset	R/W	Description	Reset Value
SYS_GPL_MFPH	SYS_BA+0x0DC	R/W	GPIOL High Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MFPGPL15				MFPGPL14			
23	22	21	20	19	18	17	16
MFPGPL13				MFPGPL12			
15	14	13	12	11	10	9	8
MFPGPL11				MFPGPL10			
7	6	5	4	3	2	1	0
MFPGPL9				MFPGPL8			

Bits	Description	
[31:28]	**MFPGPL15**	**Pin PL.15 Multi-function Pin Selection**
[27:24]	**MFPGPL14**	**Pin PL.14 Multi-function Pin Selection**
[23:20]	**MFPGPL13**	**Pin PL.13 Multi-function Pin Selection**
[19:16]	**MFPGPL12**	**Pin PL.12 Multi-function Pin Selection**
[15:12]	**MFPGPL11**	**Pin PL.11 Multi-function Pin Selection**
[11:8]	**MFPGPL10**	**Pin PL.10 Multi-function Pin Selection**
[7:4]	**MFPGPL9**	**Pin PL.9 Multi-function Pin Selection**
[3:0]	**MFPGPL8**	**Pin PL.8 Multi-function Pin Selection**

GPIOM Low Byte Multiple Function Control Register (SYS_GPM_MFPL)

Register	Offset	R/W	Description	Reset Value
SYS_GPM_MFPL	SYS_BA+0x0E0	R/W	GPIOM Low Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MFPGPM7				MFPGPM6			
23	22	21	20	19	18	17	16
MFPGPM5				MFPGPM4			
15	14	13	12	11	10	9	8
MFPGPM3				MFPGPM2			
7	6	5	4	3	2	1	0
MFPGPM1				MFPGPM0			

Bits	Description	
[31:28]	**MFPGPM7**	**Pin PM.7 Multi-function Pin Selection**
[27:24]	**MFPGPM6**	**Pin PM.6 Multi-function Pin Selection**
[23:20]	**MFPGPM5**	**Pin PM.5 Multi-function Pin Selection**
[19:16]	**MFPGPM4**	**Pin PM.4 Multi-function Pin Selection**
[15:12]	**MFPGPM3**	**Pin PM.3 Multi-function Pin Selection**
[11:8]	**MFPGPM2**	**Pin PM.2 Multi-function Pin Selection**
[7:4]	**MFPGPM1**	**Pin PM.1 Multi-function Pin Selection**
[3:0]	**MFPGPM0**	**Pin PM.0 Multi-function Pin Selection**

GPIOM High Byte Multiple Function Control Register (SYS_GPM_MFPH)

Register	Offset	R/W	Description	Reset Value
SYS_GPM_MFPH	SYS_BA+0x0E4	R/W	GPIOM High Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MFPGPM15				MFPGPM14			
23	22	21	20	19	18	17	16
MFPGPM13				MFPGPM12			
15	14	13	12	11	10	9	8
MFPGPM11				MFPGPM10			
7	6	5	4	3	2	1	0
MFPGPM9				MFPGPM8			

Bits	Description	
[31:28]	**MFPGPM15**	**Pin PM.15 Multi-function Pin Selection**
[27:24]	**MFPGPM14**	**Pin PM.14 Multi-function Pin Selection**
[23:20]	**MFPGPM13**	**Pin PM.13 Multi-function Pin Selection**
[19:16]	**MFPGPM12**	**Pin PM.12 Multi-function Pin Selection**
[15:12]	**MFPGPM11**	**Pin PM.11 Multi-function Pin Selection**
[11:8]	**MFPGPM10**	**Pin PM.10 Multi-function Pin Selection**
[7:4]	**MFPGPM9**	**Pin PM.9 Multi-function Pin Selection**
[3:0]	**MFPGPM8**	**Pin PM.8 Multi-function Pin Selection**

GPION Low Byte Multiple Function Control Register (SYS_GPN_MFPL)

Register	Offset	R/W	Description	Reset Value
SYS_GPN_MFPL	SYS_BA+0x0E8	R/W	GPION Low Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MFPGPN7				MFPGPN6			
23	22	21	20	19	18	17	16
MFPGPN5				MFPGPN4			
15	14	13	12	11	10	9	8
MFPGPN3				MFPGPN2			
7	6	5	4	3	2	1	0
MFPGPN1				MFPGPN0			

Bits	Description	
[31:28]	**MFPGPN7**	**Pin PN.7 Multi-function Pin Selection**
[27:24]	**MFPGPN6**	**Pin PN.6 Multi-function Pin Selection**
[23:20]	**MFPGPN5**	**Pin PN.5 Multi-function Pin Selection**
[19:16]	**MFPGPN4**	**Pin PN.4 Multi-function Pin Selection**
[15:12]	**MFPGPN3**	**Pin PN.3 Multi-function Pin Selection**
[11:8]	**MFPGPN2**	**Pin PN.2 Multi-function Pin Selection**
[7:4]	**MFPGPN1**	**Pin PN.1 Multi-function Pin Selection**
[3:0]	**MFPGPN0**	**Pin PN.0 Multi-function Pin Selection**

GPION High Byte Multiple Function Control Register (SYS_GPN_MFPH)

Register	Offset	R/W	Description	Reset Value
SYS_GPN_MFPH	SYS_BA+0x0EC	R/W	GPION High Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MFPGPN15				MFPGPN14			
23	22	21	20	19	18	17	16
MFPGPN13				MFPGPN12			
15	14	13	12	11	10	9	8
MFPGPN11				MFPGPN10			
7	6	5	4	3	2	1	0
MFPGPN9				MFPGPN8			

Bits	Description	
[31:28]	**MFPGPN15**	**Pin PN.15 Multi-function Pin Selection**
[27:24]	**MFPGPN14**	**Pin PN.14 Multi-function Pin Selection**
[23:20]	**MFPGPN13**	**Pin PN.13 Multi-function Pin Selection**
[19:16]	**MFPGPN12**	**Pin PN.12 Multi-function Pin Selection**
[15:12]	**MFPGPN11**	**Pin PN.11 Multi-function Pin Selection**
[11:8]	**MFPGPN10**	**Pin PN.10 Multi-function Pin Selection**
[7:4]	**MFPGPN9**	**Pin PN.9 Multi-function Pin Selection**
[3:0]	**MFPGPN8**	**Pin PN.8 Multi-function Pin Selection**

Temperature Sensor Function Control Register (SYS_TSENSRFCR)

Register	Offset	R/W	Description	Reset Value
SYS_TSENSRFCR	SYS_BA+0x104	R/W	Temperature Sensor Function Control Register	0x1000_XXXX

31	30	29	28	27	26	25	24
DATAVALID	TSTCON	REFUDEN	PD	TSENSRDATA			
23	22	21	20	19	18	17	16
TSENSRDATA							
15	14	13	12	11	10	9	8
TSENSRREF1							
7	6	5	4	3	2	1	0
TSENSRREF0							

Bits	Description	
[31]	**DATAVALID**	**Temperature Sensor Data Valid** 0 = Temperature sensor data in TSENSRDATA is not valid. 1 = Temperature sensor data in TSENSRDATA is valid.
[30]	**TSTCON**	**Temperature Sensor Test Mode Control** 0 = Temperature sensor test mode Disabled. 1 = Temperature sensor test mode Enabled.
[29]	**REFUDEN**	**Temperature Sensor Reference Data Update Enable Bit** 0 = Write to update TSENSRREF0 and TSENSRREF1 is Disabled. 1 = Write to update TSENSRREF0 and TSENSRREF1 is Enabled.
[28]	**PD**	**Temperature Sensor Power Down** 0 = Temperature sensor data is in normal operation. 1 = Temperature sensor data is in power down.
[27:16]	**TSENSRDATA**	**Temperature Sensor Data** TSENSRDATA keeps 12-bit value measured by temperature sensor.
[15:8]	**TSENSRREF1**	**Temperature Sensor Reference Value 1** TSENSRREF1 keeps 8-bit value measured at 25C for temperature conversion formula variable B calibration.
[7:0]	**TSENSRREF0**	**Temperature Sensor Reference Value 0** TSENSRREF0 keeps 8-bit value measured at 25C for temperature conversion formula variable A calibration.

GMAC 0 Miscellaneous Control Register (SYS_GMAC0MISCR)

Register	Offset	R/W	Description	Reset Value
SYS_GMAC0MISCR	SYS_BA+0x108	R/W	GMAC 0 Miscellaneous Control Register (TZNS)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
RXCLKDLY				TXCLKDLY			
15	14	13	12	11	10	9	8
Reserved			RXCLKINV	Reserved		TXCLKGEN	TXCLKINV
7	6	5	4	3	2	1	0
Reserved						PFRMTXEN	RMIIEN

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:20]	**RXCLKDLY**	**Receive Clock Path Delay Control** 0000 = 0.00ns (Default). 0001 = 0.13ns. 0010 = 0.27ns. 0011 = 0.40ns. 0100 = 0.53ns. 0101 = 0.67ns. 0110 = 0.80ns. 0111 = 0.93ns. 1000 = 1.07ns. 1001 = 1.20ns. 1010 = 1.33ns. 1011 = 1.47ns. 1100 = 1.60ns. 1101 = 1.73ns. 1110 = 1.87ns. 1111 = 2.00ns. **Note:** These bits are reserved when GMAC 0 is in RMII mode.

[19:16]	**TXCLKDLY**	**Transmit Clock Path Delay Control** 0000 = 0.00ns (Default). 0001 = 0.13ns. 0010 = 0.27ns. 0011 = 0.40ns. 0100 = 0.53ns. 0101 = 0.67ns. 0110 = 0.80ns. 0111 = 0.93ns. 1000 = 1.07ns. 1001 = 1.20ns. 1010 = 1.33ns. 1011 = 1.47ns. 1100 = 1.60ns. 1101 = 1.73ns. 1110 = 1.87ns. 1111 = 2.00ns. **Note:** These bits are reserved when GMAC 0 is in RMII mode.
[15:13]	**Reserved**	Reserved.
[12]	**RXCLKINV**	**Receive Clock Inverter Enable Bit** 0 = Receive clock (input) inverter Disabled. 1 = Receive clock (input) inverter Enabled. **Note:** This bit is reserved when GMAC 0 is in RMII mode.
[11:10]	**Reserved**	Reserved.
[9]	**TXCLKGEN**	**Transmit Clock Gating Enable Bit** 0 = Transmit clock (output) gating when entered LPI mode Disabled. 1 = Transmit clock (output) gating when entered LPI mode Enabled. **Note:** This bit is reserved when GMAC 0 is in RMII mode.
[8]	**TXCLKINV**	**Transmit Clock Inverter Enable Bit** 0 = Transmit clock (output) inverter Disabled. 1 = Transmit clock (output) inverter Enabled. **Note:** This bit is reserved when GMAC 0 is in RMII mode.
[7:2]	**Reserved**	Reserved.
[1]	**PFRMTXEN**	**Pause Frame Transmit Enable Bit** 0 = Pause frame transmit Disabled. 1 = Pause frame transmit Enabled.
[0]	**RMIIEN**	**RMII Mode Enable Bit** 0 = GMAC 0 is in RGMII mode. 1 = GMAC 0 is in RMII mode.

GMAC 1 Miscellaneous Control Register (SYS_GMAC1MISCR)

Register	Offset	R/W	Description	Reset Value
SYS_GMAC1MISCR	SYS_BA+0x10C	R/W	GMAC 1 Miscellaneous Control Register (TZNS)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
RXCLKDLY				TXCLKDLY			
15	14	13	12	11	10	9	8
Reserved			RXCLKINV	Reserved		TXCLKGEN	TXCLKINV
7	6	5	4	3	2	1	0
Reserved						PFRMTXEN	RMIIEN

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:20]	**RXCLKDLY**	**Receive Clock Path Delay Control** 0000 = 0.00ns (Default). 0001 = 0.13ns. 0010 = 0.27ns. 0011 = 0.40ns. 0100 = 0.53ns. 0101 = 0.67ns. 0110 = 0.80ns. 0111 = 0.93ns. 1000 = 1.07ns. 1001 = 1.20ns. 1010 = 1.33ns. 1011 = 1.47ns. 1100 = 1.60ns. 1101 = 1.73ns. 1110 = 1.87ns. 1111 = 2.00ns. **Note:** These bits are reserved when GMAC 1 is in RMII mode.

[19:16]	**TXCLKDLY**	**Transmit Clock Path Delay Control** 0000 = 0.00ns (Default). 0001 = 0.13ns. 0010 = 0.27ns. 0011 = 0.40ns. 0100 = 0.53ns. 0101 = 0.67ns. 0110 = 0.80ns. 0111 = 0.93ns. 1000 = 1.07ns. 1001 = 1.20ns. 1010 = 1.33ns. 1011 = 1.47ns. 1100 = 1.60ns. 1101 = 1.73ns. 1110 = 1.87ns. 1111 = 2.00ns. **Note:** These bits are reserved when GMAC 1 is in RMII mode.
[15:13]	**Reserved**	Reserved.
[12]	**RXCLKINV**	**Receive Clock Inverter Enable Bit** 0 = Receive clock (input) inverter Disabled. 1 = Receive clock (input) inverter Enabled. **Note:** This bit is reserved when GMAC 1 is in RMII mode.
[11:10]	**Reserved**	Reserved.
[9]	**TXCLKGEN**	**Transmit Clock Gating Enable Bit** 0 = Transmit clock (output) gating when entered LPI mode Disabled. 1 = Transmit clock (output) gating when entered LPI mode Enabled. **Note:** This bit is reserved when GMAC 1 is in RMII mode.
[8]	**TXCLKINV**	**Transmit Clock Inverter Enable Bit** 0 = Transmit clock (output) inverter Disabled. 1 = Transmit clock (output) inverter Enabled. **Note:** This bit is reserved when GMAC 1 is in RMII mode.
[7:2]	**Reserved**	Reserved.
[1]	**PFRMTXEN**	**Pause Frame Transmit Enable Bit** 0 = Pause frame transmit Disabled. 1 = Pause frame transmit Enabled.
[0]	**RMIIEN**	**RMII Mode Enable Bit** 0 = GMAC 1 is in RGMII mode. 1 = GMAC 1 is in RMII mode.

MAC Address 0 Low Significant Word Register (SYS_MACAD0LSR)

Register	Offset	R/W	Description	Reset Value
SYS_MACAD0LSR	SYS_BA+0x110	R	MAC Address 0 Low Significant Word Register (TZNS)	0xXXXX_XXXX

31	30	29	28	27	26	25	24
MACADRLSR							
23	22	21	20	19	18	17	16
MACADRLSR							
15	14	13	12	11	10	9	8
MACADRLSR							
7	6	5	4	3	2	1	0
MACADRLSR							

Bits	Description	
[31:0]	**MACADRLSR**	**MAC Address Low Significant Word Register**

MAC Address 0 High Significant Word Register (SYS_MACAD0HSR)

Register	Offset	R/W	Description	Reset Value
SYS_MACAD0HSR	SYS_BA+0x114	R	MAC Address 0 High Significant Word Register (TZNS)	0xXXXX_XXXX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
MACADRHSR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MACADRHSR							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**MACADRHSR**	**MAC Address High Significant Word Register**

MAC Address 1 Low Significant Word Register (SYS_MACAD1LSR)

Register	Offset	R/W	Description	Reset Value
SYS_MACAD1LSR	SYS_BA+0x118	R	MAC Address 1 Low Significant Word Register (TZNS)	0xXXXX_XXXX

31	30	29	28	27	26	25	24
MACADRLSR							
23	22	21	20	19	18	17	16
MACADRLSR							
15	14	13	12	11	10	9	8
MACADRLSR							
7	6	5	4	3	2	1	0
MACADRLSR							

Bits	Description	
[31:0]	**MACADRLSR**	**MAC Address Low Significant Word Register**

MAC Address 1 High Significant Word Register (SYS_MACAD1HSR)

Register	Offset	R/W	Description	Reset Value
SYS_MACAD1HSR	SYS_BA+0x11C	R	MAC Address 1 High Significant Word Register (TZNS)	0xXXXX_XXXX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
MACADRHSR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MACADRHSR							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**MACADRHSR**	**MAC Address High Significant Word Register**

CoreSight Debug Control Register (SYS_CSDBGCTL)

Register	Offset	R/W	Description	Reset Value
SYS_CSDBGCTL	SYS_BA+0x120	R/W	CoreSight Debug Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**LPEMU**	**DBGPWRUPACK**	**DBGPWRUPREQ**	**DBGRST**

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3]	**LPEMU**	**Low Power Emulation Enable Bit** 0 = Low power Emulation Enabled. 1 = Low power Emulation Disabled. When this bit is on, CA35 and RTP's clock and power will be maintained even the SOC in power-down mode.
[2]	**DBGPWRUPACK**	**Debug Power Up Acknowledge Bit** 0 = Debug power-up request is not ready. 1 = Debug power-up request is ready.
[1]	**DBGPWRUPREQ**	**Debug Power Up Request Bit** 0 = The power-up request Disabled. 1 = The power-up request Enabled. **Note:** If user wants to do self-hosted debug, it has to write the DBGPWRUPREQ to 1, and check the DBGPWRUPACK to 1 before self-hosted debug start. This bit will enable A35 power and clock and RTP clock.
[0]	**DBGRST**	**Debug Reset Bit** 0 = Release the reset for all debug component including A35, RTP and Coresight. 1 = Reset all debug component including A35, RTP and Coresight.

GPIOA and GPIOB Multiple Function Output Mode Select Register (SYS_GPAB_MFOS)

Register	Offset	R/W	Description	Reset Value
SYS_GPAB_MFOS	SYS_BA+0x140	R/W	GPIOA and GPIOB Multiple Function Output Mode Select Register	0x0000_0000
SYS_GPCD_MFOS	SYS_BA+0x144	R/W	GPIOC and GPIOD Multiple Function Output Mode Select Register	0x0000_0000
SYS_GPEF_MFOS	SYS_BA+0x148	R/W	GPIOE and GPIOF Multiple Function Output Mode Select Register	0x0000_0000
SYS_GPGH_MFOS	SYS_BA+0x14C	R/W	GPIOG and GPIOH Multiple Function Output Mode Select Register	0x0000_0000
SYS_GPIJ_MFOS	SYS_BA+0x150	R/W	GPIOI and GPIOJ Multiple Function Output Mode Select Register	0x0000_0000
SYS_GPKL_MFOS	SYS_BA+0x154	R/W	GPIOK and GPIOL Multiple Function Output Mode Select Register	0x0000_0000
SYS_GPMN_MFOS	SYS_BA+0x158	R/W	GPIOM and GPION Multiple Function Output Mode Select Register	0x0000_0000

31	30	29	28	27	26	25	24
GPIOyMFOS							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
GPIOyMFOS							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
GPIOxMFOS							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPIOxMFOS							

Bits	Description	
[i] i=16, 17..31	**GPIOyMFOS**	**GPIOy Pin[n] Multiple Function Pin Output Mode Select** This bit used to select multiple function pin output mode type for Py.n pin 0 = Multiple function pin output mode type is Push-pull mode. 1 = Multiple function pin output mode type is Open-drain mode. **Note:** n=0, 1..15, Max. n=15. **Note:** y= B, D, F, H, J, L, N.
[m] m=0,1..15	**GPIOxMFOS**	**GPIOx Pin[m] Multiple Function Pin Output Mode Select** This bit used to select multiple function pin output mode type for Px.m pin 0 = Multiple function pin output mode type is Push-pull mode. 1 = Multiple function pin output mode type is Open-drain mode. **Note:** Max. m=15. **Note:** y= A, C, E, F, I, K, M.

Unique Identifier Word 0~2 Register (SYS_UID0~2)

Register	Offset	R/W	Description	Reset Value
SYS_UID0	SYS_BA+0x180	R	Unique Identifier Word 0 Register (TZNS)	0x0000_0000
SYS_UID1	SYS_BA+0x184	R	Unique Identifier Word 1 Register (TZNS)	0x0000_0000
SYS_UID2	SYS_BA+0x188	R	Unique Identifier Word 2 Register (TZNS)	0x0000_0000

31	30	29	28	27	26	25	24
UID							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
UID							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
UID							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UID							

Bits	Description	
[31:0]	**UID**	**Unique ID** Unique identify number of the chip. Loaded from OTP automatically during chip power on.

Unique Customer Identifier Word 0~2 Register (SYS_UCID0~2)

Register	Offset	R/W	Description	Reset Value
SYS_UCID0	SYS_BA+0x190	R	Unique Customer Identifier Word 0 Register (TZNS)	0x0000_0000
SYS_UCID1	SYS_BA+0x194	R	Unique Customer Identifier Word 1 Register (TZNS)	0x0000_0000
SYS_UCID2	SYS_BA+0x198	R	Unique Customer Identifier Word 2 Register (TZNS)	0x0000_0000

31	30	29	28	27	26	25	24
UCID							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
UCID							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
UCID							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UCID							

Bits	Description	
[31:0]	**UCID**	**Unique Customer ID** Unique customer identifier number of the chip. Loaded from OTP automatically during chip power on.

TZS Register Lock Control Register (SYS_RLKTZS)

Some of the system control registers need to be protected to avoid inadvertent write and disturb the chip operation. These system control registers are protected after the power-on reset till user to disable register protection. For user to program these protected registers, a register protection disable sequence needs to be followed by a special programming. The register protection disable sequence is writing the data “59h”, “16h” “88h” to the register SYS_RLKTZS address at 0x4046_01A0 continuously. Any different data value, different sequence or any other write to other address during these three data writing will abort the whole sequence.

After the protection is disabled, user can check the protection disable bit at address 0x4046_01A0 bit0, 1 is protection disable, and 0 is protection enable. Then user can update the target protected register value and then write any data to the address “0x4046_01A0” to enable register protection.

This register is write for disable/enable register protection and read for the REGLCTL status

Register	Offset	R/W	Description	Reset Value
SYS_RLKTZS	SYS_BA+0x1A0	R/W	TZS Register Lock Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
REGLCTL							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**REGLCTL**	**Register Lock Control Code (Write Only)** Some registers have write-protection function. Writing these registers have to disable the protected function by writing the sequence value “59h”, “16h”, “88h” to this field. After this sequence is completed, the REGLCTL bit will be set to 1 and write-protection registers can be normal write. **REGLCTL[0]** **Register Lock Control Disable Index (Read Only)** 0 = Write-protection Enabled for writing protected registers. Any write to the protected register is ignored. 1 = Write-protection Disabled for writing protected registers.

TZNS Register Lock Control Register (SYS_RLKTZNS)

Some of the system control registers need to be protected to avoid inadvertent write and disturb the chip operation. These system control registers are protected after the power-on reset till user to disable register protection. For user to program these protected registers, a register protection disable sequence needs to be followed by a special programming. The register protection disable sequence is writing the data “59h”, “16h” “88h” to the register SYS_RLKTZNS address at 0x4046_01A4 continuously. Any different data value, different sequence or any other write to other address during these three data writing will abort the whole sequence.

After the protection is disabled, user can check the protection disable bit at address 0x4046_01A4 bit0, 1 is protection disable, and 0 is protection enable. Then user can update the target protected register value and then write any data to the address “0x4046_01A4” to enable register protection.

This register is write for disable/enable register protection and read for the REGLCTL status

Register	Offset	R/W	Description	Reset Value
SYS_RLKTZNS	SYS_BA+0x1A4	R/W	TZNS Register Lock Control Register (TZNS)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
REGLCTL							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**REGLCTL**	**Register Lock Control Code (Write Only)** Some registers have write-protection function. Writing these registers have to disable the protected function by writing the sequence value “59h”, “16h”, “88h” to this field. After this sequence is completed, the REGLCTL bit will be set to 1 and write-protection registers can be normal write. **REGLCTL[0]** **Register Lock Control Disable Index (Read Only)** 0 = Write-protection Enabled for writing protected registers. Any write to the protected register is ignored. 1 = Write-protection Disabled for writing protected registers.

SUBM Register Lock Control Register (SYS_RLKSUBM)

Some of the system control registers need to be protected to avoid inadvertent write and disturb the chip operation. These system control registers are protected after the power-on reset till user to disable register protection. For user to program these protected registers, a register protection disable sequence needs to be followed by a special programming. The register protection disable sequence is writing the data “59h”, “16h” “88h” to the register SYS_RLKSUBM address at 0x4046_01A8 continuously. Any different data value, different sequence or any other write to other address during these three data writing will abort the whole sequence.

After the protection is disabled, user can check the protection disable bit at address 0x4046_01A8 bit0, 1 is protection disable, and 0 is protection enable. Then user can update the target protected register value and then write any data to the address “0x4046_01A8” to enable register protection.

This register is write for disable/enable register protection and read for the REGLCTL status

Register	Offset	R/W	Description	Reset Value
SYS_RLKSUBM	SYS_BA+0x1A8	R/W	SUBM Register Lock Control Register (SUBM)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
REGLCTL							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**REGLCTL**	**Register Lock Control Code (Write Only)** Some registers have write-protection function. Writing these registers have to disable the protected function by writing the sequence value “59h”, “16h”, “88h” to this field. After this sequence is completed, the REGLCTL bit will be set to 1 and write-protection registers can be normal write. **REGLCTL[0]** **Register Lock Control Disable Index (Read Only)** 0 = Write-protection Enabled for writing protected registers. Any write to the protected register is ignored. 1 = Write-protection Disabled for writing protected registers.

Product Revision Register (SYS_PNUMREVR)

Register	Offset	R/W	Description	Reset Value
SYS_PNUMREVR	SYS_BA+0x1F0	R	Product Revision Register	0xXXXX_XXXX

31	30	29	28	27	26	25	24
Reserved				REV			
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:24]	**REV**	**Chip Revision** 0 = Revision A. 1 = Revision B.
[23:0]	**Reserved**	Reserved.

6.4.9 NMI Control Register

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
NMI Base Address: **NMI_BA = 0x4046_0300**				
SYS_NMIIEN	NMI_BA+0x000	R/W	NMI Source Interrupt Enable Register (SUBM)	0x0000_0000
SYS_NMISTS	NMI_BA+0x004	R	NMI Source Interrupt Status Register (SUBM)	0x0000_0000

NMI Source Interrupt Enable Register (SYS_NMIIEN)

Register	Offset	R/W	Description	Reset Value
SYS_NMIIEN	NMI_BA+0x000	R/W	NMI Source Interrupt Enable Register (SUBM)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
UART1IEN	**Reserved**	**WRHO0IEN**	**HSEM0IEN**	**EINT3IEN**	**EINT2IEN**	**EINT1IEN**	**EINT0IEN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TAMPERIEN	**RTCIEN**	**Reserved**	**CLKFAILIEN**	**Reserved**	**PWRWUIEN**	**Reserved**	**LVDIEN**

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15]	**UART1IEN**	**UART1 NMI Source Enable (Write Protect)** 0 = UART1 NMI source Disabled. 1 = UART1 NMI source Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKSUBM register.
[14]	**Reserved**	Reserved.
[13]	**WRHO0IEN**	**Wormhole 0 NMI Source Enable (Write Protect)** 0 = Wormhole 0 NMI source Disabled. 1 = Wormhole 0 NMI source Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKSUBM register.
[12]	**HSEM0IEN**	**Hardware Semaphore 0 NMI Source Enable (Write Protect)** 0 = Wormhole 0 NMI source Disabled. 1 = Wormhole 0 NMI source Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKSUBM register.
[11]	**EINT3IEN**	**External Interrupt 3 NMI Source Enable (Write Protect)** 0 = External interrupt 3 NMI source Disabled. 1 = External interrupt 3 NMI source Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKSUBM register.
[10]	**EINT2IEN**	**External Interrupt 2 NMI Source Enable (Write Protect)** 0 = External interrupt 2 NMI source Disabled. 1 = External interrupt 2 NMI source Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKSUBM register.
[9]	**EINT1IEN**	**External Interrupt 1 NMI Source Enable (Write Protect)** 0 = External interrupt 1 NMI source Disabled. 1 = External interrupt 1 NMI source Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKSUBM register.

[8]	**EINT0IEN**	**External Interrupt 0 NMI Source Enable (Write Protect)** 0 = External interrupt 0 NMI source Disabled. 1 = External interrupt 0 NMI source Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKSUBM register.
[7]	**TAMPERIEN**	**Tamper Interrupt NMI Source Enable (Write Protect)** 0 = Backup register tamper detected interrupt NMI source Disabled. 1 = Backup register tamper detected interrupt NMI source Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKSUBM register.
[6]	**RTCIEN**	**RTC NMI Source Enable (Write Protect)** 0 = RTC NMI source Disabled. 1 = RTC NMI source Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKSUBM register.
[5]	**Reserved**	Reserved.
[4]	**CLKFAILIEN**	**Clock Fail Detected NMI Source Enable (Write Protect)** 0 = Clock fail detected interrupt NMI source Disabled. 1 = Clock fail detected interrupt NMI source Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKSUBM register.
[3]	**Reserved**	Reserved.
[2]	**PWRWUIEN**	**Power-down Mode Wake-up NMI Source Enable (Write Protect)** 0 = Power-down mode wake-up NMI source Disabled. 1 = Power-down mode wake-up NMI source Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKSUBM register.
[1]	**Reserved**	Reserved.
[0]	**LVDIEN**	**LVD NMI Source Enable (Write Protect)** 0 = LVD NMI source Disabled. 1 = LVD NMI source Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKSUBM register.

NMI Source Interrupt Status Register (SYS_NMISTS)

Register	Offset	R/W	Description	Reset Value
SYS_NMISTS	NMI_BA+0x004	R	NMI Source Interrupt Status Register (SUBM)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
UART1STS	**Reserved**	**WRHO0STS**	**HSEM0STS**	**EINT3STS**	**EINT2STS**	**EINT1STS**	**EINT0STS**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TAMPERSTS	**RTCSTS**	**Reserved**	**CLKFAILSTS**	**Reserved**	**PWRWUSTS**	**Reserved**	**LVDSTS**

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15]	**UART1STS**	**UART1 NMI Status (Read Only)** 0 = UART1 interrupt is de-asserted. 1 = UART1 interrupt is asserted.
[14]	**Reserved**	Reserved.
[13]	**WRHO0STS**	**Wormhole 0 NMI Status (Read Only)** 0 = Wormhole 0 interrupt is de-asserted. 1 = Wormhole 0 interrupt is asserted.
[12]	**HSEM0STS**	**Hardware Semaphore 0 NMI Status (Read Only)** 0 = Wormhole 0 interrupt is de-asserted. 1 = Wormhole 0 interrupt is asserted.
[11]	**EINT3STS**	**External Interrupt 3 NMI Status (Read Only)** 0 = External interrupt 3 interrupt is de-asserted. 1 = External interrupt 3 interrupt is asserted.
[10]	**EINT2STS**	**External Interrupt 2 NMI Status (Read Only)** 0 = External interrupt 2 interrupt is de-asserted. 1 = External interrupt 2 interrupt is asserted.
[9]	**EINT1STS**	**External Interrupt 1 NMI Status (Read Only)** 0 = External interrupt 1 interrupt is de-asserted. 1 = External interrupt 1 interrupt is asserted.
[8]	**EINT0STS**	**External Interrupt 0 NMI Status (Read Only)** 0 = External interrupt 0 interrupt is de-asserted. 1 = External interrupt 0 interrupt is asserted.

[7]	**TAMPERSTS**	**Tamper Interrupt NMI Status (Read Only)** 0 = Backup register tamper detected interrupt is de-asserted. 1 = Backup register tamper detected interrupt is asserted.
[6]	**RTCSTS**	**RTC NMI Status (Read Only)** 0 = RTC interrupt is de-asserted. 1 = RTC interrupt is asserted.
[5]	**Reserved**	Reserved.
[4]	**CLKFAILSTS**	**Clock Fail Detected NMI Status (Read Only)** 0 = Clock fail detected interrupt is de-asserted. 1 = Clock fail detected interrupt is asserted.
[3]	**Reserved**	Reserved.
[2]	**PWRWUSTS**	**Power-down Mode Wake-up NMI Status (Read Only)** 0 = Power-down mode wake-up interrupt is de-asserted. 1 = Power-down mode wake-up interrupt is asserted.
[1]	**Reserved**	Reserved.
[0]	**LVDSTS**	**LVD NMI Status (Read Only)** 0 = LVD interrupt is de-asserted. 1 = LVD interrupt is asserted.

6.5 Clock Controller

6.5.1 Overview

The clock controller generates clocks for the whole chip, including system clocks and all peripheral clocks. The clock controller also implements the power control function with the individual clock ON/OFF control, clock source selection and a clock divider. The chip will not enter RTP-M4 Power-down mode until RTP-M4 or CA35 sets the Power-down enable bit RTPPDEN (SYS_PMUCR [24]) and the core executes the WFI instruction. The chip will not enter CA35 Power-down mode until CA35 sets the Power-down enable bit CA35PDEN (SYS_PMUCR [16]) and the core executes the WFI instruction. After that, the chip enters CA35 Power-down mode or RTP-M4 Power-down mode and waits for wake-up interrupt source triggered to leave Power-down mode. In CA35 or RTP-M4 Power-down mode, the clock controller can turn off some clock sources or PLL automatically by setting the control bit in CLK_PWRCTL. Figure 6.5-1, Figure 6.5-2 and Figure 6.5-3 show the clock generator and the overview of the clock source control.

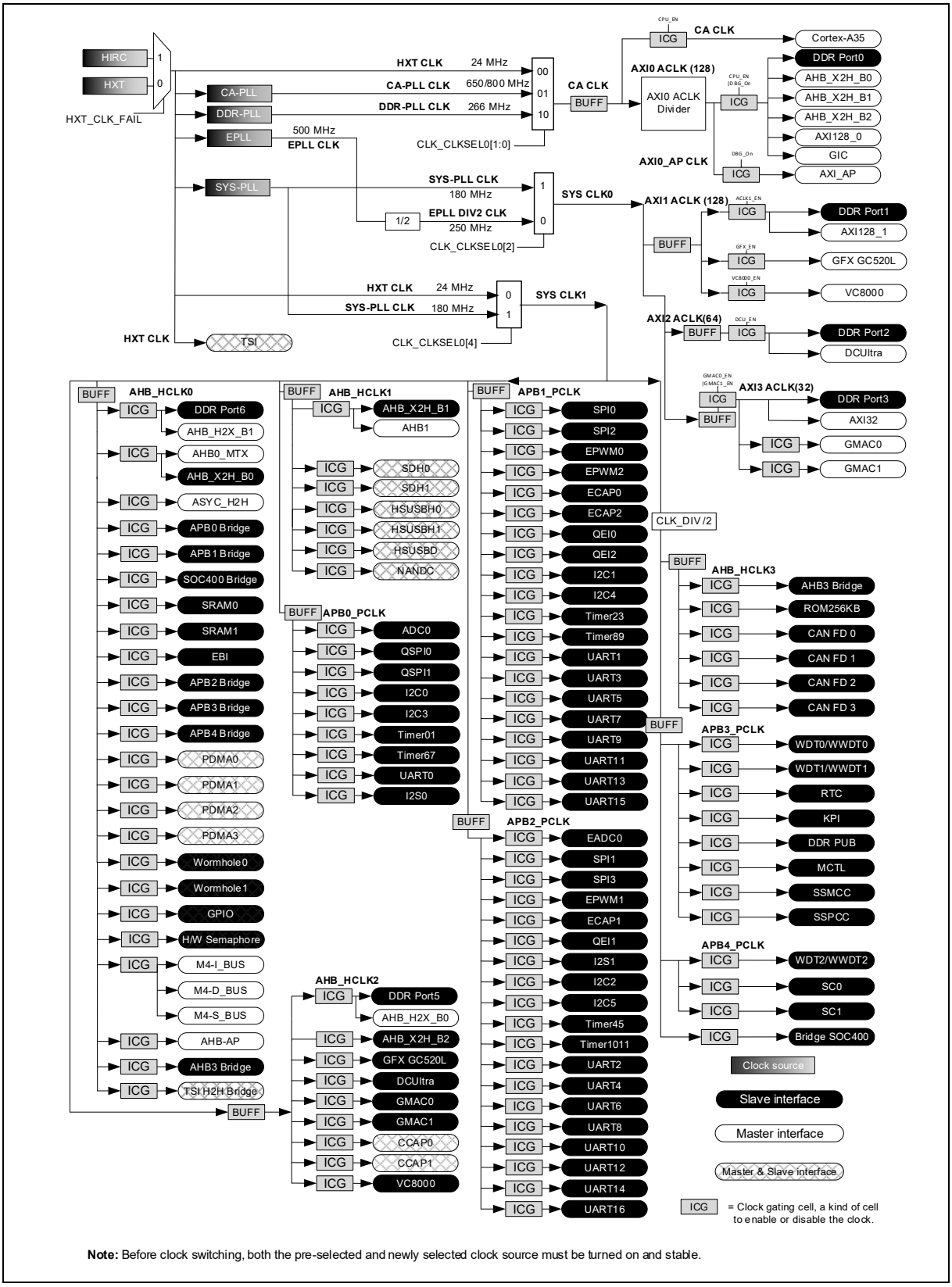


Figure 6.5-1 Clock Generator Global View Diagram (1/3)

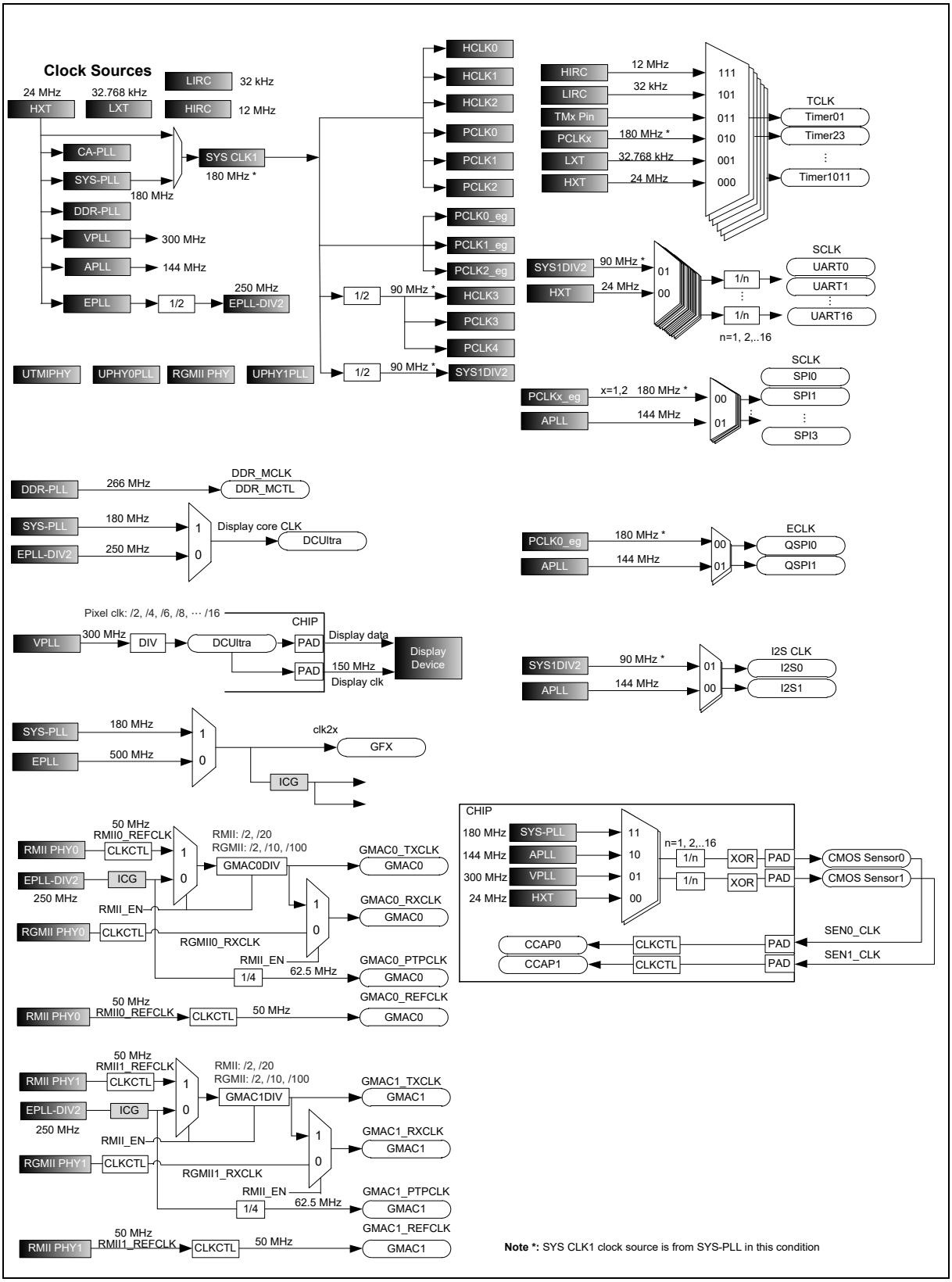


Figure 6.5-2 Clock Generator Global View Diagram (2/3)

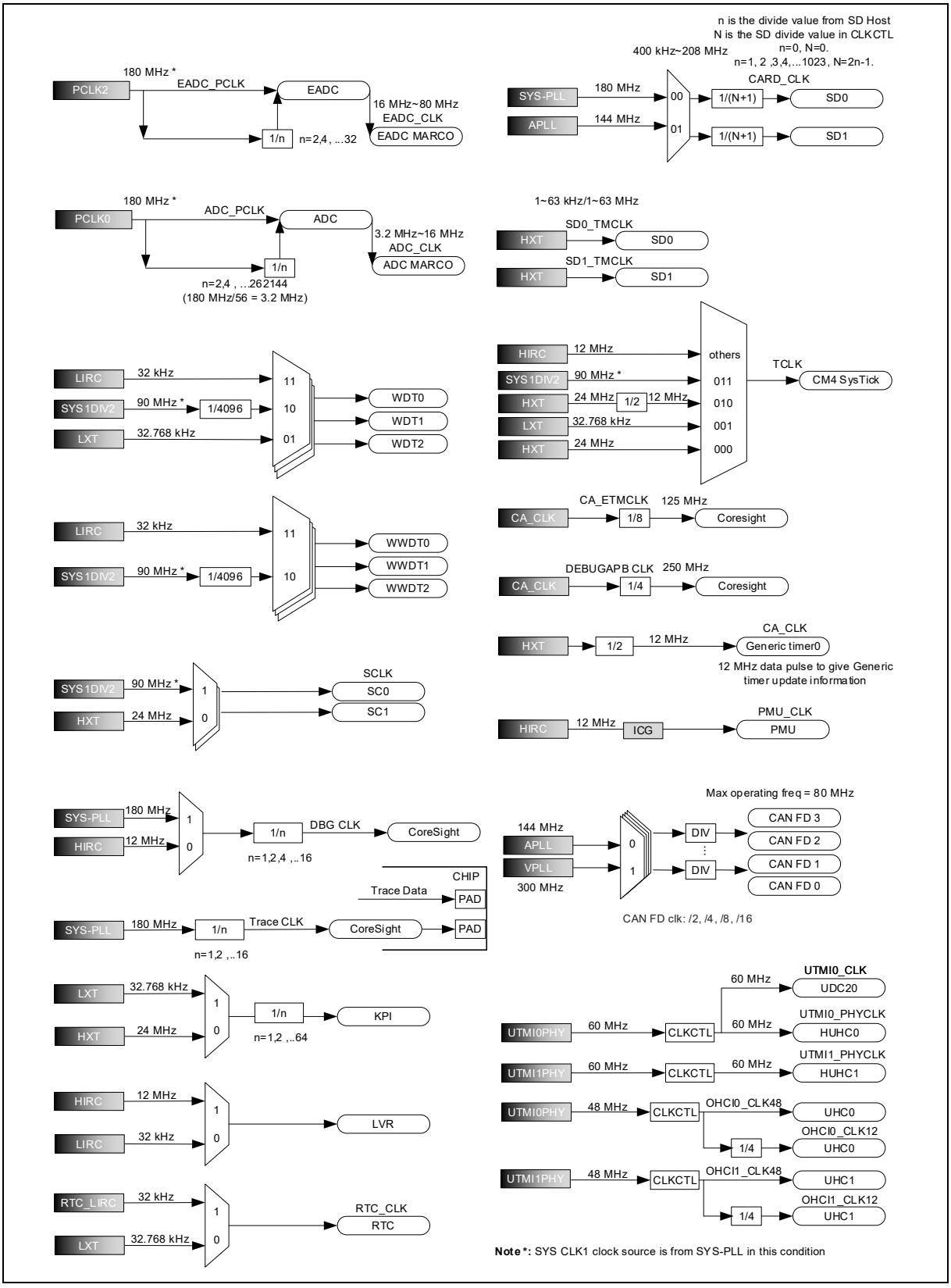


Figure 6.5-3 Clock Generator Global View Diagram (3/3)

6.5.2 Clock Generator

The clock generator consists of 10 clock sources, which are listed below:

- Programmable PLL for Cortex-CA35 output clock frequency (CA-PLL), PLL source can be selected from external 24 MHz external high speed crystal (HXT).
- Programmable PLL for system output clock frequency (SYS-PLL), PLL source can be selected from external 24 MHz external high speed crystal (HXT).
- Programmable PLL for DDR output clock frequency (DDR-PLL), PLL source can be selected from external 24 MHz external high speed crystal (HXT).
- Programmable PLL for Audio output clock frequency (APLL), PLL source can be selected from external 24 MHz external high speed crystal (HXT).
- Programmable PLL for Ethernet output clock frequency (EPLL), PLL source can be selected from external 24 MHz external high speed crystal (HXT).
- Programmable PLL for video output clock frequency (VPLL), PLL source can be selected from external 24 MHz external high speed crystal (HXT).
- 24 MHz external high speed crystal oscillator (HXT)
- 32.768 kHz external low speed crystal oscillator (LXT)
- 12 MHz internal high speed RC oscillator (HIRC)
- 32 kHz internal low speed RC oscillator (LIRC)

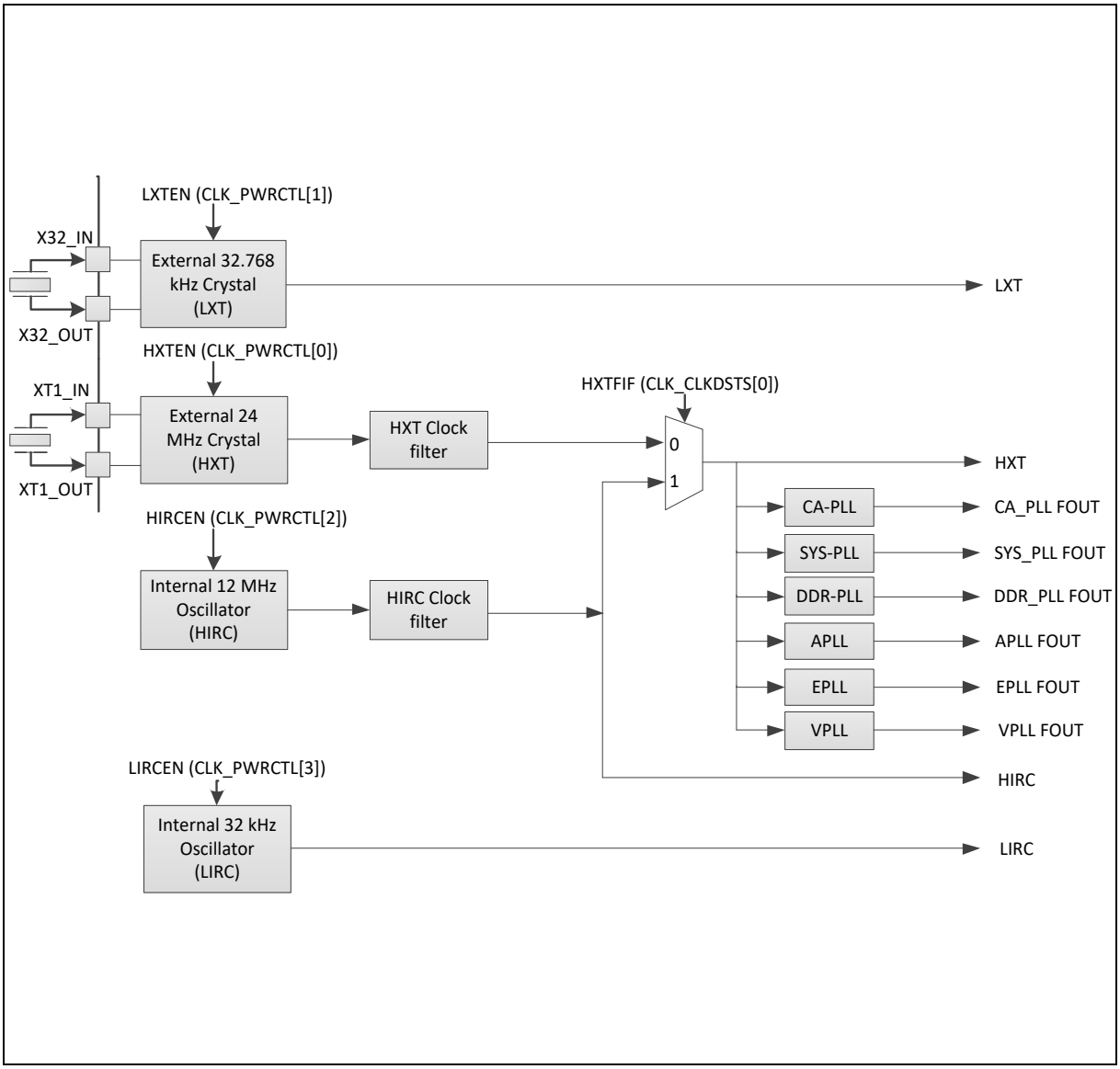


Figure 6.5-4 Clock Generator Block Diagram

Each of these clock sources has certain stable time to wait for clock operating at stable frequency. When clock source is enabled, a stable counter start counting and correlated clock stable index.

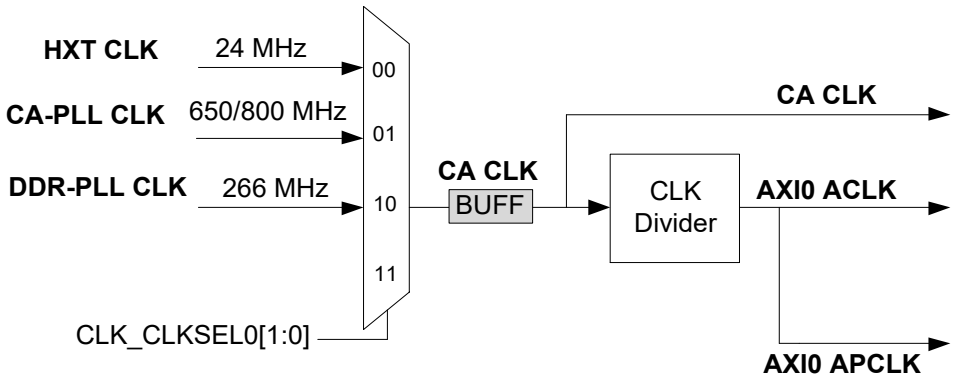
That is, HXTSTB (CLK_STATUS[0]), LXTSTB (CLK_STATUS[1]), SYSPLLSTB (CLK_STATUS[2]), LIRCSTB (CLK_STATUS[3]), HIRCSTB (CLK_STATUS[4]), CAPLLSTB (CLK_STATUS[6]), DDRPLLSTB (CLK_STATUS[8]), EPLLSTB (CLK_STATUS[9]), APLLSTB (CLK_STATUS[10]) and VPLLSTB (CLK_STATUS[11]) these bits are set to 1 after stable counter value reach a define value.

System and peripheral can use the clock as its operating clock only when correlate clock stable index is set to 1. The clock stable index will be automatically cleared when user disables the clock source (HXTEN (CLK_PWRCTL[0]), LXTEN (CLK_PWRCTL[1]), HIRCEN (CLK_PWRCTL[2]), LIRCEN (CLK_PWRCTL[3]), and PD (CLK_PLL0CTL0[16], CLK_PLL2CTL1[0], CLK_PLL3CTL1[0], CLK_PLL4CTL1[0], and CLK_PLL5CTL1[0])).

Besides, the clock stable index of HXT, HIRC, and PLL will be automatically cleared when chip enters power-down and clock stable counter will re-count after chip wake-up if correlate clock is enabled.

6.5.3 CA35 CPU Clock, System Clock and SysTick Clock

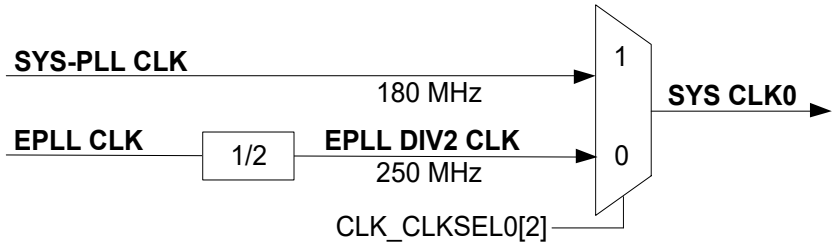
The CA35 CPU clock has 3 clock sources, which are generated from clock generator block. The clock source switch depends on the register CA35CKSEL (CLK_CLKSEL0 [1:0]). There is a clock divider before AXI0 ACLK and AXI0 APCLK. The clock of AXI0 ACLK and AXI0 APCLK can divide 4 or divide 2 by setting ACLK0DIV (CLK_CLKDIV0 [26]). The block diagram is shown in Figure 6.5-5.



Note: Before clock switching, both the pre-selected and newly selected clock source must be turned on and stable.

Figure 6.5-5 CA35 CPU Clock Block Diagram

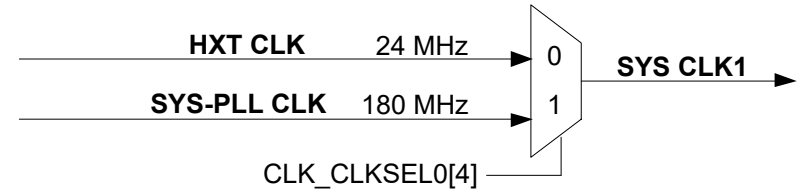
This chip has 2 system clocks, SYSCLK0, and SYSCLK1. The SYSCLK0 has 2 clock sources, which are generated from SYS-PLL and EPLL. There is a frequency divider fixed at 2 before the MUX of SYSCLK0 EPLL source. The SYSCLK0 is the bus clock source of the GFX, VC8000, DCUltra and GMAC. The block diagram is shown in Figure 6.5-6.



Note: Before clock switching, both the pre-selected and newly selected clock source must be turned on and stable.

Figure 6.5-6 SYSCLK0 Block Diagram

The SYSCLK1 has 2 clock sources, which are generated from SYS-PLL, and HXT. The SYSCLK1 is the bus clock source of Real time processor and most of all the peripherals. The block diagram is shown in Figure 6.5-7.



Note: Before clock switching, both the pre-selected and newly selected clock source must be turned on and stable.

Figure 6.5-7 SYSCLK1 Block Diagram

There are two clock fail detectors to observe HXT and LXT clock source and they have individual enable and interrupt control. When HXT detector is enabled, the HIRC clock is enabled automatically. When LXT detector is enabled, the LIRC clock is enabled automatically.

When HXT clock detector is enabled, the HXT clock source will automatically switch to HIRC if HXT clock stop is detected. If HXT clock stop condition is detected, the HXTFIF (CLK_CLKDSTS[0]) is set to 1 and chip will enter interrupt if HXTFIE (CLK_CLKDCTL[5]) is set to 1. HXT clock source stable flag, HXTSTB (CLK_STATUS[0]), will be cleared if HXT stops when using HXT fail detector function. User can try to recover HXT by disabling HXT, and enable HXT again to check if the clock stable bit is set to 1 or not. If HXT clock stable bit is set to 1, it means HXT is recovered to oscillate after re-enable action and user can switch system clock to HXT again.

The HXT clock stop detect and system clock switch to HIRC procedure is shown in Figure 6.5-8.

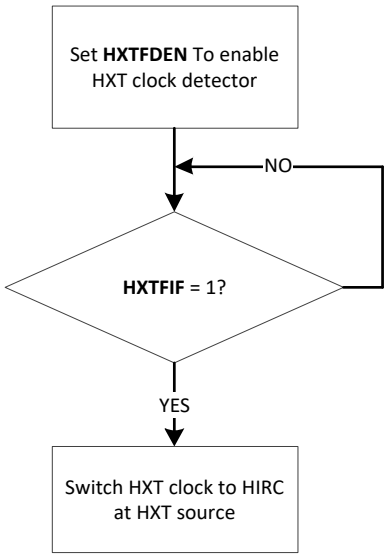
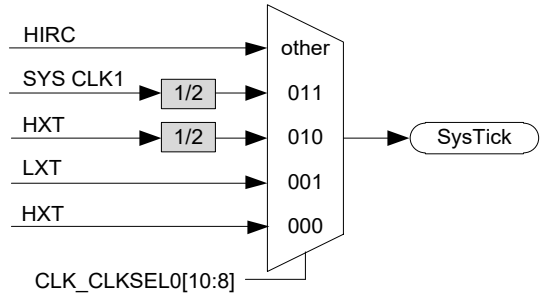


Figure 6.5-8 HXT Stop Protect Procedure



Note: Before clock switching, both the pre-selected and newly selected clock source must be turned on and stable.

Figure 6.5-9 SysTick Clock Control Block Diagram

The clock source of SysTick in processor can use CPU clock or external clock (SYST_CTRL[2]). If using external clock, the SysTick clock (STCLK) has 5 clock sources. The clock source switch depends on the setting of the register RTPSTSEL (CLK_CLKSEL0 [10:8]). The block diagram is shown in Figure 6.5-9.

6.5.4 Peripherals Clock

Each peripheral clock has its own clock source selection. Refer to the CLK_CLKSEL0, CLK_CLKSEL1, CLK_CLKSEL2, CLK_CLKSEL3 and CLK_CLKSEL4 register.

6.5.5 PLL Settings and Operation Modes

This chip is embedded with 2 PLLs and 4 Advanced PLLs. The PLL has only 1 operation mode. The Advanced PLL has 3 operation modes for use, including integer mode, fraction mode and spread spectrum mode. The integer mode is used for normal PLL operations. The fraction mode is used for the specific clock output frequency, such like 147.456 MHz. The spread spectrum mode is used for the application with EMI concern.

Output Clock Frequency Formula in different modes is shown below.

6.5.5.1 PLL Control Mode

$$\frac{F_{CLK}}{N} = \frac{F_{REF}}{M}$$

$$F_{CLKO} = \frac{F_{CLK}}{P}$$

Equation 6.5-1 PLL Control Mode Equation

Feedback divider factor, N = FBDIV[7:0] = PLL0CTL0[7:0]

Input divider, M = INDIV[5:0] = PLL0CTL0[11:8]

Output divider, P = 2^ OUTDIV[1:0] = PLL0CTL0[13:12],

EX: If OUTDIV = 00, P=1. If OUTDIV = 01, P=2. If OUTDIV = 10, P=4. If OUTDIV = 11, P=8.

For proper operation in normal mode, the following constraints must be satisfied:

1 MHz ≤ F_{REF} ≤ 50 MHz

500 MHz ≤ F_{CLK} ≤ 1500 MHz

62.5 MHz ≤ F_{CLKO} ≤ 1500 MHz

M >=2; N >=1

6.5.5.2 Advanced PLL Integer Mode

$$\frac{F_{CLK}}{N} = \frac{F_{REF}}{M}$$

$$F_{CLKO} = \frac{F_{CLK}}{P}$$

Equation 6.5-2 Advanced PLL Integer Mode Equation

Feedback divider factor, N = FBDIV[10:0] = PLLxCTL0[10:0]

Input divider, M = INDIV[5:0] = PLLxCTL0[17:12]

Output divider, P = OUTDIV[2:0] = PLLxCTL1[6:4]

Wherein F_{REF} is the input frequency, F_{CLK} is the output frequency of VCO, F_{CLKO} is the output frequency after output divider.

For proper operation in integer mode, the following constraints must be satisfied:

1 MHz ≤ F_{REF} ≤ 200 MHz

1 MHz ≤ F_{REF} /M ≤ 40 MHz

600 MHz ≤ F_{CLK} ≤ 2400 MHz

85.7 MHz ≤ F_{CLKO} ≤ 2400 MHz

6.5.5.3 Advanced PLL Fractional Mode

$$\frac{F_{CLK}}{N.X} = \frac{F_{REF}}{M}$$

$$F_{CLKO} = \frac{F_{CLK}}{P}$$

Equation 6.5-3 Advanced PLL Fractional Mode Equation

Feedback divider factor, N = FBDIV[10:0] = PLLxCTL0[10:0]

Input divider, M = INDIV[5:0] = PLLxCTL0[17:12]

Output divider, P = OUTDIV[2:0] = PLLxCTL1[6:4]

Fractional number, X=FRAC[23:0]/2^{24}, FRAC[23:0] = PLLxCTL1[31:8]

Fractional divider, N.X = N+X

For proper operation in Fraction mode, the following constraints must be satisfied:

1 MHz ≤ F_{REF} ≤ 200 MHz

10 MHz ≤ F_{REF} /M ≤ 40 MHz

600 MHz ≤ F_{CLK} ≤ 2400 MHz

85.7 MHz ≤ F_{CLKO} ≤ 2400 MHz

147.456 MHz, FCLKO= 18432/125,

let P = 5, FCLK= 18432/25,

18432/(25*N.X)=(24/M), 18432/(25*24) = (N.X)/M, 30.72 = (N.X)/M

For 24/M > 10, M = 1,

30.72 = N.X, N= 30, X = 0.72, FRAC = 12079595.52 ~= 12079596

FBDIV[10:0] = N = 30 = 0x1E

INDIV[4:0] = M = 1

OUTDIV[2:0] = P = 5

FRAC[23:0] = FRAC= 12079596 = 0xB851EC

6.5.5.4 Advanced PLL Spread Spectrum Mode

The Advanced PLL Spread Spectrum mode only supports down spreading.

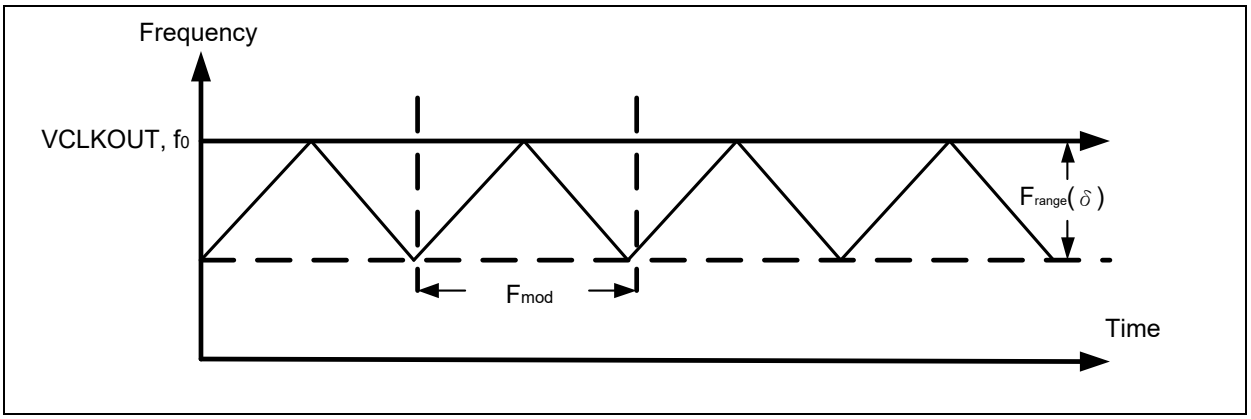


Figure 6.5-10 Advanced PLL Down Spreading in Spectrum Mode

$$\frac{F_{CLK}}{N.X} = \frac{F_{REF}}{M}$$

$$F_{CLKO} = \frac{F_{CLK}}{P}$$

$$F_{MOD} = \frac{F_{REF}}{M * SS * 2}$$

$$\mathrm{SS} = \frac{\frac{F_{REF}}{M}}{F_{MOD} * 2}$$

$$SSRATE = SS - 1$$

$$SLOPE = \frac{N.X * \delta}{SS} * 2^{24}$$

Equation 6.5-4 Advanced PLL Spread Spectrum Mode Equation

SSRATE = PLLxCTL0 [30:20]
SLOPE = PLLxCTL2 [23:0]

For proper operation in Spread Spectrum mode, the following constraints must be satisfied:

1 MHz ≤ F_{REF} ≤ 200 MHz

10 MHz ≤ F_{REF} /M ≤ 40 MHz

600 MHz ≤ F_{CLK} ≤ 2400 MHz

85.7 MHz ≤ F_{CLKO} ≤ 2400 MHz

-3% ≤ δ (spread range) ≤ 0%

15 kHz ≤ F_{MOD} (modulation frequency) ≤ 50 kHz

1. Spread Spectrum Mode Example 1

   266 MHz, FCLKO= 266 MHz, (P*266)/(N.X)=(24/M), δ (spread range) = 1.94%, modulation frequency = 50 kHz

   For 24/M > 10, M <= 2, choose M = 2.

   For 16 ≤ N ≤ 2047

   266/12 = 133/6 = 22.167= (N.X)/P, N= 133, X = 0, FRAC = 0, P= 6.

   FCLK = 2400 MHz ≤ 1596 MHz ≤ 600 MHz

   SSRATE = ((24 MHz/2)/(50 kHz*2)) – 1 = 119 (SPEC max is 50 kHz)

   SLOPE = ((133.0*1.94%) / 119) *2^{24}=363769.5187 ~= 363770

   FBDIV [10:0] = N = 133 = 0x85

   INDIV [4:0] = M = 2

   OUTDIV [2:0] = P = 6, OUTDIV = 6

   FRAC [23:0] = FRAC= 0 = 0x0

   SSRATE [10:0] = SSRATE = 119 = 0x77

   SLOPE [23:0] = SLOPE = 363770 = 0x58CFA

2. Spread Spectrum Mode Example 2

   533 MHz, FCLKO= 533 MHz, (P*533)/(N.X)=(24/M), δ (spread range) = 3%, modulation frequency = 40 kHz

   For 24/M > 10, M = 1, 1066/24 =44.4167= P*(N.X), P=2, N= 44, X = 0.4167, FRAC = 6990506.667~= 6990507

   SSRATE = ((24 MHz/1)/(40 kHz*2)) -1 = 300-1 =299

SLOPE = ((44.4167)*3%/300)*2^{24}=74518.85699 ~= 74519

FBDIV [10:0] = N = 44 =0x2C

INDIV [4:0] = M = 1

OUTDIV [2:0] = P = 2

FRAC [23:0] = FRAC= 6990507 = 0x6AAAAB

SSRATE [10:0] = SSRATE = 299 = 0x12B

SLOPE [23:0] = SLOPE = 74519 = 0x12317

Mode	F_{Out}	F_{REF}	Spread Range (Δ)	Modulation Frequency (F_{MOD})	PLLxCTL0[31:0]	PLLxCTL1[31:0]	PLLxCTL2[31:0]
Fractional mode	147.456 MHz	24 MHz	N/A	N/A	0x0004_101E	0x B851_EC50	0x0000_0000
Spread Spectrum Mode	266 MHz	24 MHz	1.94%	50 kHz	0x0778_2085	0x0000_0060	0x0005_8CFA
Spread Spectrum Mode	533 MHz	24 MHz	3%	40 kHz	0x12B8_102C	0x6AAA_AB20	0x0001_2317

Table 6.5-1 Advanced PLL Settings Example

6.5.6 Power-down Mode Clock

When entering Power-down mode, some PLL and clock sources can be configured to turn off automatically by setting control bit in CLK_PWRCTL for the better power consumption.

Besides these PLLs and clock sources, other clock sources can be enabled or disabled by different applications.

The clock sources and PLLs, which can be configured to turn off automatically, are listed below:

- When CA35 is power down:
  - Turn off 12 MHz internal high speed RC oscillator (HIRC) automatically by setting HIRCAPD (CLK_PWRCTL [23]).
  - Turn off 24 MHz external high speed crystal oscillator (HXT) automatically by setting HXTAPD (CLK_PWRCTL [22]).
- When CA35 is in power gating mode:
  - Turn off General Interrupt Controller (GIC) clock automatically by setting GICAOFF (CLK_PWRCTL [21]).
  - Turn off 12 MHz internal high speed RC oscillator (HIRC) automatically by setting HIRCAOFF (CLK_PWRCTL [15]).
  - Turn off 24 MHz external high speed crystal oscillator (HXT) automatically by setting HXTAOFF (CLK_PWRCTL [14]).
  - Turn off DDR-PLL automatically by setting DDRPLLAPD (CLK_PWRCTL [13]).
  - Turn off CA-PLL automatically by setting CAPLLAPD (CLK_PWRCTL [12]).
- When CA35 and RTP-M4 are power down:

- Turn off SYS-PLL automatically by setting SYSPLLAPD (CLK_PWRCTL [11]).

6.5.7 Clock Output

This device is equipped with a power-of-2 frequency divider that is composed by 16 chained divide-by-2 shift registers. One of the 16 shift register outputs selected by a sixteen to one multiplexer is reflected to CLKO function pin. Therefore, there are 16 options of power-of-2 divided clocks with the frequency from $F_{in}/2^1$ to $F_{in}/2^{16}$ where F_{in} is input clock frequency to the clock divider.

The output formula is $\mathbf{F_{out} = F_{in}/2^{(N+1)}}$, where F_{in} is the input clock frequency, F_{out} is the clock divider output frequency and N is the 4-bit value in FREQSEL (CLK_CLKOCTL[3:0]). When writing 1 to CLKOEN (CLK_CLKOCTL[4]), the chained counter starts to count. When writing 0 to CLKOEN (CLK_CLKOCTL[4]), the chained counter continuously runs till divided clock reaches low state and stays in low state.

If DIV1EN(CLK_CLKOCTL[5]) set to 1, the clock output clock (CLKO_CLK) will bypass power-of-2 frequency divider. The output divider clock will be output to CLKO pin directly.

When entering Power-down mode, clock output does not output clock even if the CLKO clock source is LXT.

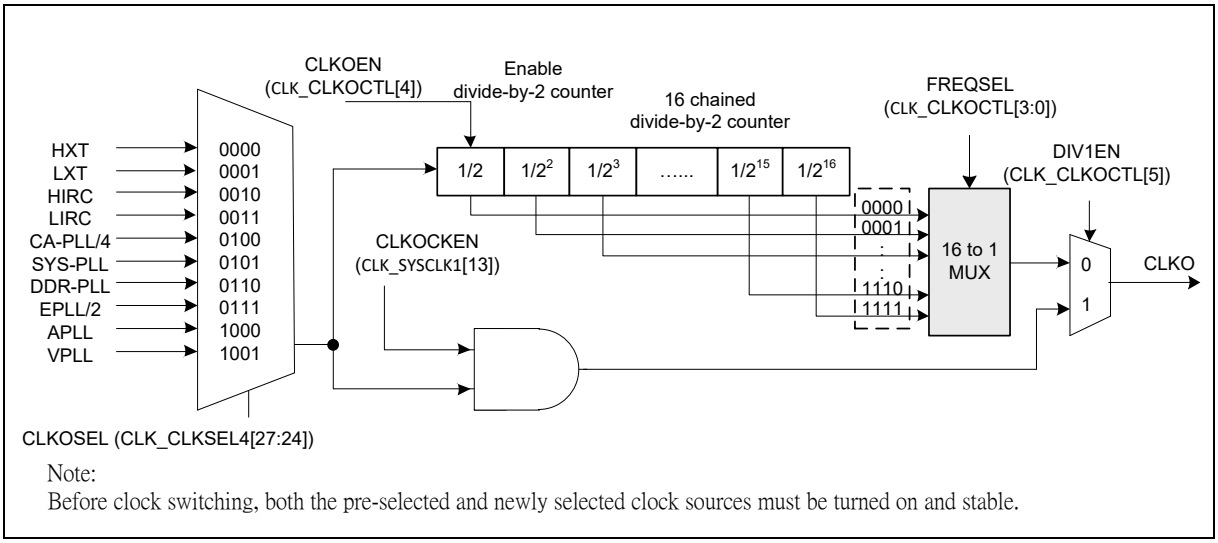


Figure 6.5-11 Clock Output Block Diagram

6.5.8 Control Registers Access Attribute

The clock controller shares part of register information to non-secure world with enable bits in SYSSINFAEN (SCU_SINFAEN[1]) register. Shared registers are enabled by default.

The clock control registers access attributes are shown in Table 6.5-2.

R: read only, **W:** write only, **R/W:** both read and write, **RAZ/WI**: read as zero, write ignore, **W_{ip}**: write authority is set by SSPCC.

Register	SYSSIAEN (SSPCC_SINFAEN[1]) =0			SYSSIAEN (SSPCC_SINFAEN[1]) =1		
	TZS	TZNS	SubM	TZS	TZNS	SubM
CLK_PWRCTL	R/W	R		R/W	R	
CLK_SYSCLK0	R/ Wip			R/ Wip		
CLK_SYSCLK1	R/ Wip			R/ Wip		
CLK_APBCLK0	R/ Wip			R/ Wip		

CLK_APBCLK1	R/ Wip		R/ Wip	
CLK_APBCLK2	R/ Wip		R/ Wip	
CLK_CLKSEL0	R/ Wip		R/ Wip	
CLK_CLKSEL1	R/ Wip		R/ Wip	
CLK_CLKSEL2	R/ Wip		R/ Wip	
CLK_CLKSEL3	R/ Wip		R/ Wip	
CLK_CLKSEL4	R/ Wip		R/ Wip	
CLK_CLKDIV0	R/ Wip		R/ Wip	
CLK_CLKDIV1	R/ Wip		R/ Wip	
CLK_CLKDIV2	R/ Wip		R/ Wip	
CLK_CLKDIV3	R/ Wip		R/ Wip	
CLK_CLKDIV4	R/ Wip		R/ Wip	
CLK_CLKOCTL	R/W		R/ W	
CLK_STATUS	R		R	
CLK_PLL0CTL0	R/W	R	R/W	R
CLK_PLL2CTL0	R/W	R	R/W	R
CLK_PLL2CTL1	R/W	R	R/W	R
CLK_PLL2CTL2	R/W	R	R/W	R
CLK_PLL3CTL0	R/W	R	R/W	R
CLK_PLL3CTL1	R/W	R	R/W	R
CLK_PLL3CTL2	R/W	R	R/W	R
CLK_PLL4CTL0	R/W	R	R/W	R
CLK_PLL4CTL1	R/W	R	R/W	R
CLK_PLL4CTL2	R/W	R	R/W	R
CLK_PLL5CTL0	R/W	R	R/W	R
CLK_PLL5CTL1	R/W	R	R/W	R
CLK_PLL5CTL2	R/W	R	R/W	R
CLK_CLKDCTL	R/W	R	R/W	R
CLK_CLKDSTS	R/W	R	R/W	
CLK_CDUPB	R/W	R	R/W	R
CLK_CDLOWB	R/W	R	R/W	R
CLK_HXTFSEL	R/W	R	R/W	R

Table 6.5-2 Clock Controller Registers Access Attribute

6.5.9 Register Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
CLK Base Address: **CLK_BA = 0x4046_0200**				
CLK_PWRCTL	CLK_BA+0x00	R/W	System Power-down Control Register	0x0000_040X
CLK_SYSCLK0	CLK_BA+0x04	R/W	AXI and AHB Device Clock Enable Control Register 0	0x0000_0035
CLK_SYSCLK1	CLK_BA+0x08	R/W	AXI and AHB Device Clock Enable Control Register 1	0x0000_5F20
CLK_APBCLK0	CLK_BA+0x0C	R/W	APB Devices Clock Enable Control Register 0	0x4000_0000
CLK_APBCLK1	CLK_BA+0x10	R/W	APB Devices Clock Enable Control Register 1	0x0007_0000
CLK_APBCLK2	CLK_BA+0x14	R/W	APB Devices Clock Enable Control Register 2	0x0000_0000
CLK_CLKSEL0	CLK_BA+0x18	R/W	Clock Source Select Control Register 0	0x0000_0000
CLK_CLKSEL1	CLK_BA+0x1C	R/W	Clock Source Select Control Register 1	0x7777_7777
CLK_CLKSEL2	CLK_BA+0x20	R/W	Clock Source Select Control Register 2	0x0000_7777
CLK_CLKSEL3	CLK_BA+0x24	R/W	Clock Source Select Control Register 3	0xFFF0_0000
CLK_CLKSEL4	CLK_BA+0x28	R/W	Clock Source Select Control Register 4	0x0200_0000
CLK_CLKDIV0	CLK_BA+0x2C	R/W	Clock Divider Number Register 0	0x0000_0000
CLK_CLKDIV1	CLK_BA+0x30	R/W	Clock Divider Number Register 1	0x0000_0000
CLK_CLKDIV2	CLK_BA+0x34	R/W	Clock Divider Number Register 2	0x0000_0000
CLK_CLKDIV3	CLK_BA+0x38	R/W	Clock Divider Number Register 3	0x0000_0000
CLK_CLKDIV4	CLK_BA+0x3C	R/W	Clock Divider Number Register 4	0x0000_0000
CLK_CLKOCTL	CLK_BA+0x40	R/W	Clock Output Control Register (Write Protect)	0x0000_0000
CLK_STATUS	CLK_BA+0x50	R	Clock Status Monitor Register	0x0000_00XX
CLK_PLL0CTL0	CLK_BA+0x60	R/W	CA-PLL Control Register 0(Write Protect)	0x0001_138A
CLK_PLL2CTL0	CLK_BA+0x80	R/W	DDR-PLL Control Register 0(Write Protect)	0x0F04_102C
CLK_PLL2CTL1	CLK_BA+0x84	R/W	DDR-PLL Control Register 1(Write Protect)	0x6B85_1E41
CLK_PLL2CTL2	CLK_BA+0x88	R/W	DDR-PLL Control Register 2(Write Protect)	0x0000_48A3
CLK_PLL3CTL0	CLK_BA+0x90	R/W	APLL Control Register 0(Write Protect)	0x0000_0000
CLK_PLL3CTL1	CLK_BA+0x94	R/W	APLL Control Register 1(Write Protect)	0x0000_0001
CLK_PLL3CTL2	CLK_BA+0x98	R/W	APLL Control Register 2(Write Protect)	0x0000_0000
CLK_PLL4CTL0	CLK_BA+0xA0	R/W	EPLL Control Register 0(Write Protect)	0x0000_60FA
CLK_PLL4CTL1	CLK_BA+0xA4	R/W	EPLL Control Register 1(Write Protect)	0x0000_0021

CLK_PLL4CTL2	CLK_BA+0xA8	R/W	EPLL Control Register 2(Write Protect)	0x0000_0000
CLK_PLL5CTL0	CLK_BA+0xB0	R/W	VPLL Control Register 0(Write Protect)	0x0000_0000
CLK_PLL5CTL1	CLK_BA+0xB4	R/W	VPLL Control Register 1(Write Protect)	0x0000_0001
CLK_PLL5CTL2	CLK_BA+0xB8	R/W	VPLL Control Register 2(Write Protect)	0x0000_0000
CLK_CLKDCTL	CLK_BA+0xC0	R/W	Clock Fail Detector Control Register(Write Protect)	0x0000_0000
CLK_CLKDSTS	CLK_BA+0xC4	R/W	Clock Fail Detector Status Register(Write Protect)	0x0000_0000
CLK_CDUPB	CLK_BA+0xC8	R/W	Clock Frequency Detector Upper Boundary Register(Write Protect)	0x0000_0000
CLK_CDLOWB	CLK_BA+0xCC	R/W	Clock Frequency Detector Lower Boundary Register(Write Protect)	0x0000_0000
CLK_CKFLTRCTL	CLK_BA+0xD0	R/W	Clock Filter Control Register (Write Protect)	0x0000_0000

6.5.10 Register Description

System Power Down Control Register (CLK_PWRCTL)

Register	Offset	R/W	Description	Reset Value
CLK_PWRCTL	CLK_BA+0x00	R/W	System Power-down Control Register	0x0000_040X

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
HIRCAPD	HXTAPD	GICAOFF	Reserved			LXTSTBS	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
HIRCAOFF	HXTAOFF	DDRPLLAPD	CAPLLAPD	SYSPLLAPD	HXTDS	Reserved	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				LIRCEN	HIRCEN	LXTEN	HXTEN

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23]	**HIRCAPD**	**HIRC Auto Off Option when CA35 is Power Down (Write Protect)** 0 = When CA35 cores are in WFI, the HIRC enable bit is the same as CLK_PWRCTL[2]. 1 = When CA35 cores are in WFI, switch the HIRC enable bit to low automatically. **Note 1:** This bit is write protected. Refer to the SYS_RLKTZS register. **Note 2:** This bit only works when the CA35 cores are in the WFI. If CA35 cores are not in WFI, the HIRC enable bit is controlled by CLK_PWRCTL[2].
[22]	**HXTAPD**	**HXT Auto Off Option when CA35 is Power Down (Write Protect)** 0 = When CA35 cores are in WFI, the HXT enable bit is the same as CLK_PWRCTL[0]. 1 = When CA35 cores are in WFI, switch the HXT enable bit to low automatically. **Note 1:** This bit is write protected. Refer to the SYS_RLKTZS register. **Note 2:** This bit only works when the CA35 cores are in the WFI. If CA35 cores are not in WFI, the HXT enable bit is controlled by CLK_PWRCTL[0].
[21]	**GICAOFF**	**GIC CLK Auto Off Option when CA35 is Power Gating (Write Protect)** 0 = When CA35 core power is not ready, the GIC CLK will not be gated. 1 = When CA35 core power is not ready, the GIC CLK will be gated. **Note 1:** This bit is write protected. Refer to the SYS_RLKTZS register. **Note 2:** This bit only works when the CA35 core power is not ready. If CA35 core power is ready, the HIRC enable bit is controlled by CLK_PWRCTL[2].
[20:18]	**Reserved**	Reserved.
[17:16]	**LXTSTBS**	**LXT Stable Count Select (Write Protect)** 00 = LXT stable count = 16384 clocks. 01 = LXT stable count = 65536 clocks. 10 = LXT stable count = 131072 clocks. 11 = LXT stable count = 327680 clocks. **Note:** These bits are write protected. Refer to the SYS_RLKTZS register.

[15]	**HIRCAOFF**	**HIRC Auto Off Option when CA35 is Power Gating (Write Protect)** 0 = When CA35 core power is not ready, the HIRC enable bit is the same as CLK_PWRCTL[2]. 1 = When CA35 core power is not ready, switch the HIRC enable bit to low automatically. **Note 1:** This bit is write protected. Refer to the SYS_RLKTZS register. **Note 2:** This bit only works when the CA35 core power is not ready. If CA35 core power is ready, the HIRC enable bit is controlled by CLK_PWRCTL[2].
[14]	**HXTAOFF**	**HXT Auto Off Option when CA35 is Power Gating (Write Protect)** 0 = When CA35 core power is not ready, the HXT enable bit is the same as CLK_PWRCTL[0]. 1 = When CA35 core power is not ready, switch the HXT enable bit to low automatically. **Note 1:** This bit is write protected. Refer to the SYS_RLKTZS register. **Note 2:** This bit only works when the CA35 core power is not ready. If CA35 core power is ready, the HXT enable bit is controlled by CLK_PWRCTL[0].
[13]	**DDRPLLAPD**	**DDRPLL Auto Power Down Option when CA35 is Power Gating (Write Protect)** 0 = When CA35 core power is not ready, the PD pin of DDR-PLL is the same as CLK_PLL2CTL1[0]. 1 = When CA35 core power is not ready, set the PD pin of DDR-PLL to high automatically. **Note 1:** This bit is write protected. Refer to the SYS_RLKTZS register. **Note 2:** This bit only works when the CA35 core power is not ready. If CA35 core power is ready, the PLL PD pin status is controlled by CLK_PLL2CTL1[0].
[12]	**CAPLLAPD**	**CAPLL Auto Power Down Option when CA35 is Power Gating (Write Protect)** 0 = When CA35 core power is not ready, the PD pin of CA-PLL is the same as CLK_PLL0CTL0[16]. 1 = When CA35 core power is not ready, set the PD pin of CA-PLL to high automatically. **Note 1:** This bit is write protected. Refer to the SYS_RLKTZS register. **Note 2:** This bit only works when the CA35 core power is not ready. If CA35 core power is ready, the PLL PD pin status is controlled by CLK_PLL0CTL0[16].
[11]	**SYSPLLAPD**	**SYSPLL Auto Power Down Option when CA35 and RTP-M4 are Power Down (Write Protect)** 0 = When CA35 and RTP-M4 and TSI are all in WFI, the PD pin of SYS-PLL is the same as CLK_PLLCTL[16]. 1 = When CA35 and RTP-M4 and TSI are all in WFI, set the PD pin of SYS-PLL to high automatically. **Note 1:** This bit is write protected. Refer to the SYS_RLKTZS register. **Note 2:** This bit only works when the CA35 and RTP-M4 and TSI are not all in Power-down mode. If one of RTP-M4 , CA35, or TSI is not in Power-down mode, the PLL PD pin status is controlled by CLK_PLLCTL [16] in Crypto Sub System.
[10]	**HXTDS**	**HXT Drive Current Strength (Write Protect)** 0 = Lower power consumption requirement for 2.5V~3.3V I/O power application. 1 = higher noise immunity requirement for 2.5V~3.3V I/O power application.(default) **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[9:4]	**Reserved**	Reserved.
[3]	**LIRCEN**	**LIRC Enable Bit (Write Protect)** 0 = 32 kHz internal low speed RC oscillator (LIRC) Disabled. 1 = 32 kHz internal low speed RC oscillator (LIRC) Enabled. **Note 1:** This bit is write protected. Refer to the SYS_RLKTZS register.
[2]	**HIRCEN**	**HIRC Enable Bit (Write Protect)** The HCLK default clock source is from HIRC and this bit default value is 1. 0 = 12 MHz internal high speed RC oscillator (HIRC) Disabled. 1 = 12 MHz internal high speed RC oscillator (HIRC) Enabled. **Note 1:** This bit is write protected. Refer to the SYS_RLKTZS register. **Note 2:** HIRC cannot be disabled and HIRCEN will always read as 1 if HXTFQIEN or is set.

[1]	**LXTEN**	**LXT Enable Bit (Write Protect)** 0 = 32.768 kHz external low speed crystal (external LXT) Disabled. 1 = 32.768 kHz external low speed crystal (external LXT) Enabled. **Note 1:** This bit is write protected. Refer to the SYS_RLKTZS register. **Note 2:** LXT cannot be disabled and LXTEN will always read as 1 if HCLK clock source is selected from LXT when the LXT clock source is selected as external LXT by setting C32KS(RTC_LXTCTL[6]) to 1.
[0]	**HXTEN**	**HXT Enable Bit (Write Protect)** 0 = 24 MHz external high speed crystal (HXT) Disabled. 1 = 24 MHz external high speed crystal (HXT) Enabled. **Note 1:** This bit is write protected. Refer to the SYS_RLKTZS register. **Note 2:** HXT cannot be disabled and HXTEN will always read as 1 if HCLK clock source is selected from HXT or PLL (clock source from HXT).

System Clock Enable Control Register 0 (CLK_SYSCLK0)

The bits in this register are used to enable/disable clock for system clock, AXI bus devices clock.

Register	Offset	R/W	Description	Reset Value
CLK_SYSCLK0	CLK_BA+0x04	R/W	AXI and AHB Device Clock Enable Control Register 0	0x0000_0035

31	30	29	28	27	26	25	24
Reserved	CCAP1EN	CCAP0EN	GMAC1EN	GMAC0EN	DCUEN	VDECEN	GFXEN
23	22	21	20	19	18	17	16
Reserved	HUSBH1EN	HUSBH0EN	USBHEN	USBDEN	NANDEN	SDH1EN	SDH0EN
15	14	13	12	11	10	9	8
Reserved				CAN3CKEN	CAN2CKEN	CAN1CKEN	CAN0CKEN
7	6	5	4	3	2	1	0
Reserved		DDR6CKEN	DDR0CKEN	LVRDBEN	TAHBCKEN	RTPEN	Reserved

Bits	Description	
[31]	**Reserved**	Reserved.
[30]	**CCAP1EN**	**CCAP1 Clock Enable Bit (TZNS)** 0 = CCAP1 clock Disabled. 1 = CCAP1 clock Enabled.
[29]	**CCAP0EN**	**CCAP0 Clock Enable Bit (TZNS)** 0 = CCAP0 clock Disabled. 1 = CCAP0 clock Enabled.
[28]	**GMAC1EN**	**Gigabit Ethernet MAC 1 AXI Clock Enable Bit (TZNS)** 0 = Gigabit Ethernet MAC 1 AXI clock Disabled. 1 = Gigabit Ethernet MAC 1 AXI clock Enabled.
[27]	**GMAC0EN**	**Gigabit Ethernet MAC 0 AXI Clock Enable Bit (TZNS)** 0 = Gigabit Ethernet MAC 0 AXI clock Disabled. 1 = Gigabit Ethernet MAC 0 AXI clock Enabled.
[26]	**DCUEN**	**DCUltra Clock Enable Bit (TZNS)** 0 = DCUltra clock Disabled. 1 = DCUltra clock Enabled.
[25]	**VDECEN**	**VC8000 Clock Enable Bit (TZNS)** 0 = VC8000 clock Disabled. 1 = VC8000 clock Enabled.
[24]	**GFXEN**	**GFX Clock Enable Bit (TZNS)** 0 = GFX clock Disabled. 1 = GFX clock Enabled.
[23]	**Reserved**	Reserved.
[22]	**HUSBH1EN**	**High Speed USBH Clock Enable Bit (TZNS)**

		0 = HUSBH1 clock Disabled. 1 = HUSBH1 clock Enabled.
[21]	**HUSBH0EN**	**High Speed USBH Clock Enable Bit (TZNS)** 0 = HUSBH0 clock Disabled. 1 = HUSBH0 clock Enabled.
[20]	**USBHEN**	**USBH Clock Enable Bit (TZNS)** 0 = USBH clock Disabled. 1 = USBH clock Enabled.
[19]	**USBDEN**	**USBD Clock Enable Bit** 0 = USBD clock Disabled. 1 = USBD clock Enabled.
[18]	**NANDEN**	**NAND Controller Clock Enable Bit** 0 = NAND controller clock Disabled. 1 = NAND controller clock Enabled.
[17]	**SDH1EN**	**SD1 Host Controller AHB clock Enable Bit** 0 = SDH1 controller AHB clock Disabled. 1 = SDH1 controller AHB clock Enabled.
[16]	**SDH0EN**	**SD0 Host Controller AHB clock Enable Bit** 0 = SDH0 controller AHB clock Disabled. 1 = SDH0 controller AHB clock Enabled.
[15:12]	**Reserved**	Reserved.
[11]	**CAN3CKEN**	**CAN FD 3 Clock Enable Bit** 0 = CAN FD 3 clock Disabled. 1 = CAN FD 3 clock Enabled.
[10]	**CAN2CKEN**	**CAN FD 2 Clock Enable Bit** 0 = CAN FD 2 clock Disabled. 1 = CAN FD 2 clock Enabled.
[9]	**CAN1CKEN**	**CAN FD 1 Clock Enable Bit** 0 = CAN FD 1 clock Disabled. 1 = CAN FD 1 clock Enabled.
[8]	**CAN0CKEN**	**CAN FD 0 Clock Enable Bit** 0 = CAN FD 0 clock Disabled. 1 = CAN FD 0 clock Enabled.
[7:6]	**Reserved**	Reserved.
[5]	**DDR6CKEN**	**DDR Port6 Clock Enable Bit (Write Protect)** 0 = DDR Port6 peripheral clock Disabled. 1 = DDR Port6 peripheral clock Enabled. **Note 1:** This bit is write protected. Refer to the SYS_RLKTZS register. **Note 2:** The peripheral clock of DDR Port 1, DDR Port 2, DDR Port 3, DDR Port 4, DDR Port5, DDR Port7 are enabled automatically by the peripheral clock enable bit of the IP on the corresponding bus.
[4]	**DDR0CKEN**	**DDR Port0 Clock Enable Bit (Write Protect)** 0 = DDR Port0 peripheral clock Disabled. 1 = DDR Port0 peripheral clock Enabled. **Note 1:** This bit is write protected. Refer to the SYS_RLKTZS register.

		Note 2: The peripheral clock of DDR Port 1, DDR Port 2, DDR Port 3, DDR Port 4, DDR Port5, DDR Port7 are enabled automatically by the peripheral clock enable bit of the IP on the corresponding bus.
[3]	**LVRDBEN**	**LVR Debounce Clock Enable Bit (Write Protect)** 0 = LVR Debounce clock Disabled. 1 = LVR Debounce clock Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[2]	**TAHBCKEN**	**TSI AHB HCLK Clock Enable Bit (Write Protect)** 0 = TSI AHB HCLK Disabled. 1 = TSI AHB HCLK Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[1]	**RTPEN**	**CPU RTP CortexM4 AHB Clock Enable Bit (Write Protect)** 0 = Cortex M4 AHB clock Disabled. 1 = Cortex M4 AHB clock Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[0]	**Reserved**	Reserved.

System Clock Enable Control Register 1 (CLK_SYSCLK1)

The bits in this register are used to enable/disable clock for system clock, AHB bus devices clock.

Register	Offset	R/W	Description	Reset Value
CLK_SYSCLK1	CLK_BA+0x08	R/W	AXI and AHB Device Clock Enable Control Register 1	0x0000_5F20

31	30	29	28	27	26	25	24
Reserved		GPNCKEN	GPMCKEN	GPLCKEN	GPKCKEN	GPJCKEN	GPICKEN
23	22	21	20	19	18	17	16
GPHCKEN	GPGCKEN	GPFCKEN	GPECKEN	GPDCKEN	GPCCKEN	GPBCKEN	GPACKEN
15	14	13	12	11	10	9	8
Reserved	GTMRCKEN	CLKOCKEN	DBGCKEN	TRACKEN	ROMCKEN	SRAM1CKEN	SRAM0CKEN
7	6	5	4	3	2	1	0
EBICKEN	HWSCKEN	WH1CKEN	WH0CKEN	PDMA3EN	PDMA2EN	PDMA1EN	PDMA0EN

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29]	**GPNCKEN**	**GPION AHB Clock Enable Bit** 0 = GPION port clock Disabled. 1 = GPION port clock Enabled.
[28]	**GPMCKEN**	**GPIOM AHB Clock Enable Bit** 0 = GPIOM port clock Disabled. 1 = GPIOM port clock Enabled.
[27]	**GPLCKEN**	**GPIOL AHB Clock Enable Bit** 0 = GPIOL port clock Disabled. 1 = GPIOL port clock Enabled.
[26]	**GPKCKEN**	**GPIOK AHB Clock Enable Bit** 0 = GPIOK port clock Disabled. 1 = GPIOK port clock Enabled.
[25]	**GPJCKEN**	**GPIOJ AHB Clock Enable Bit** 0 = GPIOJ port clock Disabled. 1 = GPIOJ port clock Enabled.
[24]	**GPICKEN**	**GPIOI AHB Clock Enable Bit** 0 = GPIOI port clock Disabled. 1 = GPIOI port clock Enabled.
[23]	**GPHCKEN**	**GPIOH AHB Clock Enable Bit** 0 = GPIOH port clock Disabled. 1 = GPIOH port clock Enabled.
[22]	**GPGCKEN**	**GPIOG AHB Clock Enable Bit** 0 = GPIOG port clock Disabled.

		1 = GPIOG port clock Enabled.
[21]	**GPFCKEN**	**GPIOF AHB Clock Enable Bit** 0 = GPIOF port clock Disabled. 1 = GPIOF port clock Enabled.
[20]	**GPECKEN**	**GPIOE AHB Clock Enable Bit** 0 = GPIOE port clock Disabled. 1 = GPIOE port clock Enabled.
[19]	**GPDCKEN**	**GPIOD AHB Clock Enable Bit** 0 = GPIOD port clock Disabled. 1 = GPIOD port clock Enabled.
[18]	**GPCCKEN**	**GPIOC AHB Clock Enable Bit** 0 = GPIOC port clock Disabled. 1 = GPIOC port clock Enabled.
[17]	**GPBCKEN**	**GPIOB AHB Clock Enable Bit** 0 = GPIOB port clock Disabled. 1 = GPIOB port clock Enabled.
[16]	**GPACKEN**	**GPIOA AHB Clock Enable Bit** 0 = GPIOA port clock Disabled. 1 = GPIOA port clock Enabled.
[15]	**Reserved**	Reserved.
[14]	**GTMRCKEN**	**Cortex A35 Generic Timer Clock Enable Bit (Write Protect)** 0 = Cortex A35 Generic timer clock Disabled. 1 = Cortex A35 Generic timer clock Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[13]	**CLKOCKEN**	**CLKO Clock Enable Bit (Write Protect)** 0 = CLKO clock Disabled. 1 = CLKO clock Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[12]	**DBGCKEN**	**Coresight Debug Clock Enable Bit (Write Protect)** 0 = Coresight debug clock Disabled. 1 = Coresight debug clock Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[11]	**TRACKEN**	**Coresight Trace Clock Enable Bit (Write Protect)** 0 = Coresight trace clock Disabled. 1 = Coresight trace clock Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[10]	**ROMCKEN**	**ROM AHB Clock Enable Bit (Write Protect)** 0 = ROM AHB clock Disabled. 1 = ROM AHB clock Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[9]	**SRAM1CKEN**	**SRAM Bank1 Controller Clock Enable Bit (Write Protect)** 0 = SRAM bank1 clock Disabled. 1 = SRAM bank1 clock Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.

[8]	**SRAM0CKEN**	**SRAM Bank0 Controller Clock Enable Bit (Write Protect)** 0 = SRAM bank0 clock Disabled. 1 = SRAM bank0 clock Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[7]	**EBICKEN**	**EBI Controller Clock Enable Bit** 0 = EBI peripheral clock Disabled. 1 = EBI peripheral clock Enabled.
[6]	**HWSCKEN**	**Hardware Semaphore Clock Enable Bit (Write Protect, TZNS)** 0 = Hardware Semaphore peripheral clock Disabled. 1 = Hardware Semaphore clock Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZNS register.
[5]	**WH1CKEN**	**Wormhole 1 Peripheral Clock Enable Bit (Write Protect)** 0 = Wormhole 1 clock Disabled. 1 = Wormhole 1 clock Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[4]	**WH0CKEN**	**Wormhole 0 Peripheral Clock Enable Bit (Write Protect, TZNS)** 0 = Wormhole 0 clock Disabled. 1 = Wormhole 0 clock Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZNS register.
[3]	**PDMA3EN**	**PDMA3 Controller Clock Enable Bit** 0 = PDMA3 peripheral clock Disabled. 1 = PDMA3 peripheral clock Enabled.
[2]	**PDMA2EN**	**PDMA2 Controller Clock Enable Bit** 0 = PDMA2 peripheral clock Disabled. 1 = PDMA2 peripheral clock Enabled.
[1]	**PDMA1EN**	**PDMA1 Controller Clock Enable Bit** 0 = PDMA1 peripheral clock Disabled. 1 = PDMA1 peripheral clock Enabled.
[0]	**PDMA0EN**	**PDMA0 Controller Clock Enable Bit** 0 = PDMA0 peripheral clock Disabled. 1 = PDMA0 peripheral clock Enabled.

APB Devices Clock Enable Control Register 0 (CLK_APBCLK0)

The bits in this register are used to enable/disable clock for peripheral controller clocks.

Register	Offset	R/W	Description	Reset Value
CLK_APBCLK0	CLK_BA+0x0C	R/W	APB Devices Clock Enable Control Register 0	0x4000_0000

31	30	29	28	27	26	25	24
KPICKEN	DDRPCKEN	RTCCKEN	UART16CKEN	UART15CKEN	UART14CKEN	UART13CKEN	UART12CKEN
23	22	21	20	19	18	17	16
UART11CKEN	UART10CKEN	UART9CKEN	UART8CKEN	UART7CKEN	UART6CKEN	UART5CKEN	UART4CKEN
15	14	13	12	11	10	9	8
UART3CKEN	UART2CKEN	UART1CKEN	UART0CKEN	TMR11CKEN	TMR10CKEN	TMR9CKEN	TMR8CKEN
7	6	5	4	3	2	1	0
TMR7CKEN	TMR6CKEN	TMR5CKEN	TMR4CKEN	TMR3CKEN	TMR2CKEN	TMR1CKEN	TMR0CKEN

Bits	Description	
[31]	**KPICKEN**	**Keypad Interface Clock Enable Bit** 0 = Keypad interface clock Disabled. 1 = Keypad interface clock Enabled.
[30]	**DDRPCKEN**	**DDR PHY Utility Block Clock Enable Bit** 0 = DDR PHY utility clock Disabled. 1 = DDR PHY utility clock Enabled.
[29]	**RTCCKEN**	**RTC Clock Enable Bit (Shared)** 0 = RTC Clock Disabled. 1 = RTC Clock Enabled. **Note:** This bit is shared attribute when **SYSSIAEN** is set to 1.
[28]	**UART16CKEN**	**UART16 Clock Enable Bit** 0 = UART16 clock Disabled. 1 = UART16 clock Enabled.
[27]	**UART15CKEN**	**UART15 Clock Enable Bit** 0 = UART15 clock Disabled. 1 = UART15 clock Enabled.
[26]	**UART14CKEN**	**UART14 Clock Enable Bit** 0 = UART14 clock Disabled. 1 = UART14 clock Enabled.
[25]	**UART13CKEN**	**UART13 Clock Enable Bit** 0 = UART13 clock Disabled. 1 = UART13 clock Enabled.
[24]	**UART12CKEN**	**UART12 Clock Enable Bit** 0 = UART12 clock Disabled. 1 = UART12 clock Enabled.

[23]	**UART11CKEN**	**UART11 Clock Enable Bit** 0 = UART11 clock Disabled. 1 = UART11 clock Enabled.
[22]	**UART10CKEN**	**UART10 Clock Enable Bit** 0 = UART10 clock Disabled. 1 = UART10 clock Enabled.
[21]	**UART9CKEN**	**UART9 Clock Enable Bit** 0 = UART9 clock Disabled. 1 = UART9 clock Enabled.
[20]	**UART8CKEN**	**UART8 Clock Enable Bit** 0 = UART8 clock Disabled. 1 = UART8 clock Enabled.
[19]	**UART7CKEN**	**UART7 Clock Enable Bit** 0 = UART7 clock Disabled. 1 = UART7 clock Enabled.
[18]	**UART6CKEN**	**UART6 Clock Enable Bit** 0 = UART6 clock Disabled. 1 = UART6 clock Enabled.
[17]	**UART5CKEN**	**UART5 Clock Enable Bit** 0 = UART5 clock Disabled. 1 = UART5 clock Enabled.
[16]	**UART4CKEN**	**UART4 Clock Enable Bit** 0 = UART4 clock Disabled. 1 = UART4 clock Enabled.
[15]	**UART3CKEN**	**UART3 Clock Enable Bit** 0 = UART3 clock Disabled. 1 = UART3 clock Enabled.
[14]	**UART2CKEN**	**UART2 Clock Enable Bit** 0 = UART2 clock Disabled. 1 = UART2 clock Enabled.
[13]	**UART1CKEN**	**UART1 Clock Enable Bit** 0 = UART1 clock Disabled. 1 = UART1 clock Enabled.
[12]	**UART0CKEN**	**UART0 Clock Enable Bit** 0 = UART0 clock Disabled. 1 = UART0 clock Enabled.
[11]	**TMR11CKEN**	**Timer11 Clock Enable Bit** 0 = Timer11 clock Disabled. 1 = Timer11 clock Enabled.
[10]	**TMR10CKEN**	**Timer10 Clock Enable Bit** 0 = Timer10 clock Disabled. 1 = Timer10 clock Enabled.
[9]	**TMR9CKEN**	**Timer9 Clock Enable Bit**

		0 = Timer9 clock Disabled. 1 = Timer9 clock Enabled.
[8]	**TMR8CKEN**	**Timer8 Clock Enable Bit** 0 = Timer8 clock Disabled. 1 = Timer8 clock Enabled.
[7]	**TMR7CKEN**	**Timer7 Clock Enable Bit** 0 = Timer7 clock Disabled. 1 = Timer7 clock Enabled.
[6]	**TMR6CKEN**	**Timer6 Clock Enable Bit** 0 = Timer6 clock Disabled. 1 = Timer6 clock Enabled.
[5]	**TMR5CKEN**	**Timer5 Clock Enable Bit** 0 = Timer5 clock Disabled. 1 = Timer5 clock Enabled.
[4]	**TMR4CKEN**	**Timer4 Clock Enable Bit** 0 = Timer4 clock Disabled. 1 = Timer4 clock Enabled.
[3]	**TMR3CKEN**	**Timer3 Clock Enable Bit** 0 = Timer3 clock Disabled. 1 = Timer3 clock Enabled.
[2]	**TMR2CKEN**	**Timer2 Clock Enable Bit** 0 = Timer2 clock Disabled. 1 = Timer2 clock Enabled.
[1]	**TMR1CKEN**	**Timer1 Clock Enable Bit** 0 = Timer1 clock Disabled. 1 = Timer1 clock Enabled.
[0]	**TMR0CKEN**	**Timer0 Clock Enable Bit** 0 = Timer0 clock Disabled. 1 = Timer0 clock Enabled.

APB Devices Clock Enable Control Register 1 (CLK_APBCLK1)

The bits in this register are used to enable/disable clock for peripheral controller clocks.

Register	Offset	R/W	Description	Reset Value
CLK_APBCLK1	CLK_BA+0x10	R/W	APB Devices Clock Enable Control Register 1	0x0007_0000

31	30	29	28	27	26	25	24
Reserved					EPWM2CKEN	EPWM1CKEN	EPWM0CKEN
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved					WDT2CKEN	WDT1CKEN	WDT0CKEN
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		SC1CKEN	SC0CKEN	Reserved			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
QSPI1CKEN	QSPI0CKEN	I2C5CKEN	I2C4CKEN	I2C3CKEN	I2C2CKEN	I2C1CKEN	I2C0CKEN

Bits	Description	
[31:27]	**Reserved**	Reserved.
[26]	**EPWM2CKEN**	**EPWM2 Clock Enable Bit** 0 = EPWM2 clock Disabled. 1 = EPWM2 clock Enabled.
[25]	**EPWM1CKEN**	**EPWM1 Clock Enable Bit** 0 = EPWM1 clock Disabled. 1 = EPWM1 clock Enabled.
[24]	**EPWM0CKEN**	**EPWM0 Clock Enable Bit** 0 = EPWM0 clock Disabled. 1 = EPWM0 clock Enabled.
[23:19]	**Reserved**	Reserved.
[18]	**WDT2CKEN**	**Watchdog Timer 2 Clock Enable Bit (Write Protect, SUBM)** 0 = Watchdog timer 2 clock Disabled. 1 = Watchdog timer 2 clock Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKSUBM register.
[17]	**WDT1CKEN**	**Watchdog Timer 1 Clock Enable Bit (Write Protect)** 0 = Watchdog timer 1 clock Disabled. 1 = Watchdog timer 1 clock Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register.
[16]	**WDT0CKEN**	**Watchdog Timer 0 Clock Enable Bit (Write Protect)** 0 = Watchdog timer 0 clock Disabled. 1 = Watchdog timer 0 clock Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[15:14]	**Reserved**	Reserved.

[13]	**SC1CKEN**	**SC1 Clock Enable Bit** 0 = SC1 clock Disabled. 1 = SC1 clock Enabled.
[12]	**SC0CKEN**	**SC0 Clock Enable Bit** 0 = SC0 clock Disabled. 1 = SC0 clock Enabled.
[11:8]	**Reserved**	Reserved.
[7]	**QSPI1CKEN**	**QSPI1 Clock Enable Bit** 0 = QSPI1 clock Disabled. 1 = QSPI1 clock Enabled.
[6]	**QSPI0CKEN**	**QSPI0 Clock Enable Bit** 0 = QSPI0 clock Disabled. 1 = QSPI0 clock Enabled.
[5]	**I2C5CKEN**	**I2C5 Clock Enable Bit** 0 = I2C5 clock Disabled. 1 = I2C5 clock Enabled.
[4]	**I2C4CKEN**	**I2C4 Clock Enable Bit** 0 = I2C4 clock Disabled. 1 = I2C4 clock Enabled.
[3]	**I2C3CKEN**	**I2C3 Clock Enable Bit** 0 = I2C3 clock Disabled. 1 = I2C3 clock Enabled.
[2]	**I2C2CKEN**	**I2C2 Clock Enable Bit** 0 = I2C2 clock Disabled. 1 = I2C2 clock Enabled.
[1]	**I2C1CKEN**	**I2C1 Clock Enable Bit** 0 = I2C1 clock Disabled. 1 = I2C1 clock Enabled.
[0]	**I2C0CKEN**	**I2C0 Clock Enable Bit** 0 = I2C0 clock Disabled. 1 = I2C0 clock Enabled.

APB Devices Clock Enable Control Register 2 (CLK_APBCLK2)

The bits in this register are used to enable/disable clock for peripheral controller clocks.

Register	Offset	R/W	Description	Reset Value
CLK_APBCLK2	CLK_BA+0x14	R/W	APB Devices Clock Enable Control Register 2	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						EADCCKEN	ADCCKEN
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	QEI2CKEN	QEI1CKEN	QEI0CKEN	Reserved	ECAP2CKEN	ECAP1CKEN	ECAP0CKEN
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SPI3CKEN	SPI2CKEN	SPI1CKEN	SPI0CKEN	SSPCCEN	SSMCCEN	I2S1CKEN	I2S0CKEN

Bits	Description	
[31:26]	**Reserved**	Reserved.
[25]	**EADCCKEN**	**EADC Clock Enable Bit** 0 = EADC clock Disabled. 1 = EADC clock Enabled.
[24]	**ADCCKEN**	**ADC Clock Enable Bit** 0 = ADC clock Disabled. 1 = ADC clock Enabled.
[23:15]	**Reserved**	Reserved.
[14]	**QEI2CKEN**	**QEI2 Clock Enable Bit** 0 = QEI2 clock Disabled. 1 = QEI2 clock Enabled.
[13]	**QEI1CKEN**	**QEI1 Clock Enable Bit** 0 = QEI1 clock Disabled. 1 = QEI1 clock Enabled.
[12]	**QEI0CKEN**	**QEI0 Clock Enable Bit** 0 = QEI0 clock Disabled. 1 = QEI0 clock Enabled.
[11]	**Reserved**	Reserved.
[10]	**ECAP2CKEN**	**ECAP2 Clock Enable Bit** 0 = ECAP2 clock Disabled. 1 = ECAP2 clock Enabled.
[9]	**ECAP1CKEN**	**ECAP1 Clock Enable Bit** 0 = ECAP1 clock Disabled. 1 = ECAP1 clock Enabled.

[8]	**ECAP0CKEN**	**ECAP0 Clock Enable Bit** 0 = ECAP0 clock Disabled. 1 = ECAP0 clock Enabled.
[7]	**SPI3CKEN**	**SPI3 Clock Enable Bit** 0 = SPI3 clock Disabled. 1 = SPI3 clock Enabled.
[6]	**SPI2CKEN**	**SPI2 Clock Enable Bit** 0 = SPI2 clock Disabled. 1 = SPI2 clock Enabled.
[5]	**SPI1CKEN**	**SPI1 Clock Enable Bit** 0 = SPI1 clock Disabled. 1 = SPI1 clock Enabled.
[4]	**SPI0CKEN**	**SPI0 Clock Enable Bit** 0 = SPI0 clock Disabled. 1 = SPI0 clock Enabled.
[3]	**SSPCCEN**	**SSPCC Clock Enable Bit (Write Protect)** 0 = SSPCC clock Disabled. 1 = SSPCC clock Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[2]	**SSMCCEN**	**SSMCC Clock Enable Bit (Write Protect)** 0 = SSMCC clock Disabled. 1 = SSMCC clock Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[1]	**I2S1CKEN**	**I2S1 Clock Enable Bit** 0 = I2S1 Clock Disabled. 1 = I2S1 Clock Enabled.
[0]	**I2S0CKEN**	**I2S0 Clock Enable Bit** 0 = I2S0 Clock Disabled. 1 = I2S0 Clock Enabled.

Clock Source Select Control Register 0 (CLK_CLKSEL0)

Register	Offset	R/W	Description	Reset Value
CLK_CLKSEL0	CLK_BA+0x18	R/W	Clock Source Select Control Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				DBGSEL	GFXSEL	Reserved	DCUSEL
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				SD1SEL		SD0SEL	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CCAP1SEL		CCAP0SEL		Reserved	RTPSTSEL		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			SYSCK1SEL	LVRDBSEL	SYSCK0SEL	CA35CKSEL	

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27]	**DBGSEL**	**Coresight DBG Clock Source Selection (Write Protect)** Before clock switching, the related clock sources (both pre-select and new-select) must be turned on. 0 = Clock source from HIRC.1. 1 = Clock source from SYS-PLL. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[26]	**GFXSEL**	**GFX Core Clock Source Selection (TZNS)** Before clock switching, the related clock sources (both pre-select and new-select) must be turned on. 0 = Clock source from EPLL. 1 = Clock source from SYS-PLL.
[25]	**Reserved**	Reserved.
[24]	**DCUSEL**	**Display Controller Ultra Core Clock Source Selection (TZNS)** Before clock switching, the related clock sources (both pre-select and new-select) must be turned on. 0 = Clock source from EPLL/2. 1 = Clock source from SYS-PLL.
[23:20]	**Reserved**	Reserved.
[19:18]	**SD1SEL**	**SD HOST1 Controller Core Logic Clock Source Selection** Before clock switching, the related clock sources (both pre-select and new-select) must be turned on. 00 = Clock source from SYS-PLL. 01 = Clock source from APLL. Others = Reserved.
[17:16]	**SD0SEL**	**SD HOST0 Controller Core Logic Clock Source Selection** Before clock switching, the related clock sources (both pre-select and new-select) must be turned on. 00 = Clock source from SYS-PLL. 01 = Clock source from APLL. Others = Reserved.

[15:14]	**CCAP1SEL**	**CCAP1 Sensor Clock Source Selection (TZNS)** Before clock switching, the related clock sources (both pre-select and new-select) must be turned on. 00 = Clock source from HXT. 01 = Clock source from VPLL. 10 = Clock source from APLL. 11 = Clock source from SYS-PLL.
[13:12]	**CCAP0SEL**	**CCAP0 Sensor Clock Source Selection (TZNS)** Before clock switching, the related clock sources (both pre-select and new-select) must be turned on. 00 = Clock source from HXT. 01 = Clock source from VPLL. 10 = Clock source from APLL. 11 = Clock source from SYS-PLL.
[11]	**Reserved**	Reserved.
[10:8]	**RTPSTSEL**	**RTP CortexM4 SysTick Clock Source Selection (Write Protect, SUBM)** If SYST_CTRL[2]=0, SysTick uses listed clock source below. 000 = Clock source from HXT. 001 = Clock source from LXT. 010 = Clock source from HXT/2. 011 = Clock source from SYSCLK1/2. Others = Clock source from HIRC. **Note 1:** if SysTick clock source is not from HCLK (i.e. SYST_CTRL[2] = 0), SysTick need clock frequency must less than or equal to HCLK/2. **Note 2:** These bits are write protected. Refer to the SYS_RLKSUBM register.
[7:5]	**Reserved**	Reserved.
[4]	**SYSCK1SEL**	**System Clock Source Selection (Write Protect)** Before clock switching, the related clock sources (both pre-select and new-select) must be turned on. 0 = Clock source from HXT. 1 = Clock source from SYS-PLL. **Note:** These bits are write protected. Refer to the SYS_RLKTZS register.
[3]	**LVRDBSEL**	**LVR Debounce Clock Selection (Write Protect)** Before clock switching, the related clock sources (both pre-select and new-select) must be turned on. 0 = Clock source from LIRC. 1 = Clock source from HIRC. **Note:** These bits are write protected. Refer to the SYS_RLKTZS register.
[2]	**SYSCK0SEL**	**System Clock Source Selection (Write Protect)** Before clock switching, the related clock sources (both pre-select and new-select) must be turned on. 0 = Clock source from EPLL/2. 1 = Clock source from SYS-PLL. **Note:** These bits are write protected. Refer to the SYS_RLKTZS register.
[1:0]	**CA35CKSEL**	**Cortex A35 CPU Clock Source Selection (Write Protect)** Before clock switching, the related clock sources (both pre-select and new-select) must be turned on. 00 = Clock source from HXT. 01 = Clock source from CA-PLL. 10 = Clock source from DDR-PLL. Others = Reserved. **Note 2:** These bits are write protected. Refer to the SYS_RLKTZS register.

Clock Source Select Control Register 1 (CLK_CLKSEL1)

Before clock switching, the related clock sources (pre-selected and newly-selected) must be turned on.

Register	Offset	R/W	Description	Reset Value
CLK_CLKSEL1	CLK_BA+0x1C	R/W	Clock Source Select Control Register 1	0x7777_7777

31	30	29	28	27	26	25	24
Reserved	TMR7SEL			Reserved	TMR6SEL		
23	22	21	20	19	18	17	16
Reserved	TMR5SEL			Reserved	TMR4SEL		
15	14	13	12	11	10	9	8
Reserved	TMR3SEL			Reserved	TMR2SEL		
7	6	5	4	3	2	1	0
Reserved	TMR1SEL			Reserved	TMR0SEL		

Bits	Description	
[31]	**Reserved**	Reserved.
[30:28]	**TMR7SEL**	**TIMER7 Clock Source Selection** 000 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 001 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT). 010 = Clock source from PCLK0. 011 = Clock source from external clock TM7 pin. 101 = Clock source from 32 kHz internal low speed RC oscillator (LIRC). 111 = Clock source from 12 MHz internal high speed RC oscillator (HIRC). Others = Reserved.
[27]	**Reserved**	Reserved.
[26:24]	**TMR6SEL**	**TIMER6 Clock Source Selection** 000 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 001 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT). 010 = Clock source from PCLK0. 011 = Clock source from external clock TM6 pin. 101 = Clock source from 32 kHz internal low speed RC oscillator (LIRC). 111 = Clock source from 12 MHz internal high speed RC oscillator (HIRC). Others = Reserved.
[23]	**Reserved**	Reserved.
[22:20]	**TMR5SEL**	**TIMER5 Clock Source Selection** 000 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 001 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT). 010 = Clock source from PCLK2. 011 = Clock source from external clock TM5 pin. 101 = Clock source from 32 kHz internal low speed RC oscillator (LIRC). 111 = Clock source from 12 MHz internal high speed RC oscillator (HIRC).

		Others = Reserved.
[19]	**Reserved**	Reserved.
[18:16]	**TMR4SEL**	**TIMER4 Clock Source Selection** 000 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 001 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT). 010 = Clock source from PCLK2. 011 = Clock source from external clock TM4 pin. 101 = Clock source from 32 kHz internal low speed RC oscillator (LIRC). 111 = Clock source from 12 MHz internal high speed RC oscillator (HIRC). Others = Reserved.
[15]	**Reserved**	Reserved.
[14:12]	**TMR3SEL**	**TIMER3 Clock Source Selection** 000 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 001 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT). 010 = Clock source from PCLK1. 011 = Clock source from external clock TM3 pin. 101 = Clock source from 32 kHz internal low speed RC oscillator (LIRC). 111 = Clock source from 12 MHz internal high speed RC oscillator (HIRC). Others = Reserved.
[11]	**Reserved**	Reserved.
[10:8]	**TMR2SEL**	**TIMER2 Clock Source Selection** 000 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 001 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT). 010 = Clock source from PCLK1. 011 = Clock source from external clock TM2 pin. 101 = Clock source from 32 kHz internal low speed RC oscillator (LIRC). 111 = Clock source from 12 MHz internal high speed RC oscillator (HIRC). Others = Reserved.
[7]	**Reserved**	Reserved.
[6:4]	**TMR1SEL**	**TIMER1 Clock Source Selection** 000 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 001 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT). 010 = Clock source from PCLK0. 011 = Clock source from external clock TM1 pin. 101 = Clock source from 32 kHz internal low speed RC oscillator (LIRC). 111 = Clock source from 12 MHz internal high speed RC oscillator (HIRC). Others = Reserved.
[3]	**Reserved**	Reserved.
[2:0]	**TMR0SEL**	**TIMER0 Clock Source Selection** 000 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 001 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT). 010 = Clock source from PCLK0. 011 = Clock source from external clock TM0 pin. 101 = Clock source from 32 kHz internal low speed RC oscillator (LIRC). 111 = Clock source from 12 MHz internal high speed RC oscillator (HIRC).

		Others = Reserved.

Clock Source Select Control Register 2 (CLK_CLKSEL2)

Before clock switching, the related clock sources (pre-selected and newly-selected) must be turned on.

Register	Offset	R/W	Description	Reset Value
CLK_CLKSEL2	CLK_BA+0x20	R/W	Clock Source Select Control Register 2	0x0000_7777

31	30	29	28	27	26	25	24
UART7SEL		UART6SEL		UART5SEL		UART4SEL	
23	22	21	20	19	18	17	16
UART3SEL		UART2SEL		UART1SEL		UART0SEL	
15	14	13	12	11	10	9	8
Reserved	TMR11SEL			Reserved	TMR10SEL		
7	6	5	4	3	2	1	0
Reserved	TMR9SEL			Reserved	TMR8SEL		

Bits	Description	
[31:30]	**UART7SEL**	**UART7 Clock Source Selection** 00 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 01 = Clock source from SYSCLK1/2. Others = Reserved.
[29:28]	**UART6SEL**	**UART6 Clock Source Selection** 00 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 01 = Clock source from SYSCLK1/2. Others = Reserved.
[27:26]	**UART5SEL**	**UART5 Clock Source Selection** 00 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 01 = Clock source from SYSCLK1/2. Others = Reserved.
[25:24]	**UART4SEL**	**UART4 Clock Source Selection** 00 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 01 = Clock source from SYSCLK1/2. Others = Reserved.
[23:22]	**UART3SEL**	**UART3 Clock Source Selection** 00 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 01 = Clock source from SYSCLK1/2. Others = Reserved.
[21:20]	**UART2SEL**	**UART2 Clock Source Selection** 00 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 01 = Clock source from SYSCLK1/2. Others = Reserved.
[19:18]	**UART1SEL**	**UART1 Clock Source Selection**

		00 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 01 = Clock source from SYSCLK1/2. Others = Reserved.
[17:16]	**UART0SEL**	**UART0 Clock Source Selection** 00 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 01 = Clock source from SYSCLK1/2. Others = Reserved.
[15]	**Reserved**	Reserved.
[14:12]	**TMR11SEL**	**TIMER11 Clock Source Selection** 000 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 001 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT). 010 = Clock source from PCLK2. 011 = Clock source from external clock TM11 pin. 101 = Clock source from 32 kHz internal low speed RC oscillator (LIRC). 111 = Clock source from 12 MHz internal high speed RC oscillator (HIRC). Others = Reserved.
[11]	**Reserved**	Reserved.
[10:8]	**TMR10SEL**	**TIMER10 Clock Source Selection** 000 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 001 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT). 010 = Clock source from PCLK2. 011 = Clock source from external clock TM10 pin. 101 = Clock source from 32 kHz internal low speed RC oscillator (LIRC). 111 = Clock source from 12 MHz internal high speed RC oscillator (HIRC). Others = Reserved.
[7]	**Reserved**	Reserved.
[6:4]	**TMR9SEL**	**TIMER9 Clock Source Selection** 000 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 001 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT). 010 = Clock source from PCLK1. 011 = Clock source from external clock TM9 pin. 101 = Clock source from 32 kHz internal low speed RC oscillator (LIRC). 111 = Clock source from 12 MHz internal high speed RC oscillator (HIRC). Others = Reserved.
[3]	**Reserved**	Reserved.
[2:0]	**TMR8SEL**	**TIMER8 Clock Source Selection** 000 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 001 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT). 010 = Clock source from PCLK1. 011 = Clock source from external clock TM8 pin. 101 = Clock source from 32 kHz internal low speed RC oscillator (LIRC). 111 = Clock source from 12 MHz internal high speed RC oscillator (HIRC). Others = Reserved.

Clock Source Select Control Register 3 (CLK_CLKSEL3)

Before clock switching, the related clock sources (pre-select and new-select) must be turned on.

Register	Offset	R/W	Description	Reset Value
CLK_CLKSEL3	CLK_BA+0x24	R/W	Clock Source Select Control Register 3	0xFFF0_0000

31	30	29	28	27	26	25	24
WWDT2SEL		WDT2SEL		WWDT1SEL		WDT1SEL	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
WWDT0SEL		WDT0SEL		Reserved		UART16SEL	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
UART15SEL		UART14SEL		UART13SEL		UART12SEL	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UART11SEL		UART10SEL		UART9SEL		UART8SEL	

Bits	Description	
[31:30]	**WWDT2SEL**	**Window Watchdog Timer Clock Source Selection (Write Protect, SUBM)** 10 = Clock source from PCLK4/4096. 11 = Clock source from 32 kHz internal low speed RC oscillator (LIRC). Others = Reserved. **Note:** These bits are write protected. Refer to the SYS_RLKTZNS register.
[29:28]	**WDT2SEL**	**Watchdog Timer Clock Source Selection (Write Protect, SUBM)** 01 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT). 10 = Clock source from PCLK4/4096. 11 = Clock source from 32 kHz internal low speed RC oscillator (LIRC). Others = Reserved. **Note:** These bits are write protected. Refer to the SYS_RLKTZNS register.
[27:26]	**WWDT1SEL**	**Window Watchdog Timer Clock Source Selection (Write Protect)** 10 = Clock source from PCLK3/4096. 11 = Clock source from 32 kHz internal low speed RC oscillator (LIRC). Others = Reserved. **Note:** These bits are write protected. Refer to the SYS_RLKTZNS register or SYS_RLKTZS register.
[25:24]	**WDT1SEL**	**Watchdog Timer Clock Source Selection (Write Protect)** 01 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT). 10 = Clock source from PCLK3/4096. 11 = Clock source from 32 kHz internal low speed RC oscillator (LIRC). Others = Reserved. **Note:** These bits are write protected. Refer to the SYS_RLKTZNS register or SYS_RLKTZS register.
[23:22]	**WWDT0SEL**	**Window Watchdog Timer Clock Source Selection (Write Protect)** 10 = Clock source from PCLK3/4096. 11 = Clock source from 32 kHz internal low speed RC oscillator (LIRC). Others = Reserved.

		Note: These bits are write protected. Refer to the SYS_RLKTZS register.
[21:20]	**WDT0SEL**	**Watchdog Timer Clock Source Selection (Write Protect)** 01 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT). 10 = Clock source from PCLK3/4096. 11 = Clock source from 32 kHz internal low speed RC oscillator (LIRC). Others = Reserved. **Note:** These bits are write protected. Refer to the SYS_RLKTZS register.
[19:18]	**Reserved**	Reserved.
[17:16]	**UART16SEL**	**UART16 Clock Source Selection** 00 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 01 = Clock source from SYSCLK1/2. Others = Reserved.
[15:14]	**UART15SEL**	**UART15 Clock Source Selection** 00 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 01 = Clock source from SYSCLK1/2. Others = Reserved.
[13:12]	**UART14SEL**	**UART14 Clock Source Selection** 00 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 01 = Clock source from SYSCLK1/2. Others = Reserved.
[11:10]	**UART13SEL**	**UART13 Clock Source Selection** 00 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 01 = Clock source from SYSCLK1/2. Others = Reserved.
[9:8]	**UART12SEL**	**UART12 Clock Source Selection** 00 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 01 = Clock source from SYSCLK1/2. Others = Reserved.
[7:6]	**UART11SEL**	**UART11 Clock Source Selection** 00 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 01 = Clock source from SYSCLK1/2. Others = Reserved.
[5:4]	**UART10SEL**	**UART10 Clock Source Selection** 00 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 01 = Clock source from SYSCLK1/2. Others = Reserved.
[3:2]	**UART9SEL**	**UART6 Clock Source Selection** 00 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 01 = Clock source from SYSCLK1/2. Others = Reserved.
[1:0]	**UART8SEL**	**UART8 Clock Source Selection** 00 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 01 = Clock source from SYSCLK1/2. Others = Reserved.

Clock Source Select Control Register 4 (CLK_CLKSEL4)

Before clock switching, the related clock sources (pre-select and new-select) must be turned on.

Register	Offset	R/W	Description	Reset Value
CLK_CLKSEL4	CLK_BA+0x28	R/W	Clock Source Select Control Register 4	0x0200_0000

31	30	29	28	27	26	25	24
Reserved	KPISEL	SC1SEL	SC0SEL	CKOSEL			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				CANFD3SEL	CANFD2SEL	CANFD1SEL	CANFD0SEL
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
I2S1SEL		I2S0SEL		QSPI1SEL		QSPI0SEL	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SPI3SEL		SPI2SEL		SPI1SEL		SPI0SEL	

Bits	Description	
[31]	**Reserved**	Reserved.
[30]	**KPISEL**	**Key Pad Interface Clock Source Selection** 0 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 1 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT).
[29]	**SC1SEL**	**Smart Card 1 Clock Source Selection** 0 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 1 = Clock source from PCLK4.
[28]	**SC0SEL**	**Smart Card 0 Clock Source Selection** 0 = Clock source from 24 MHz external high speed crystal oscillator (HXT). 1 = Clock source from PCLK4.
[27:24]	**CKOSEL**	**Reference Clock Our Source Selection** This field selects which clock is used to be the source of reference clock output 0000 = Clock source from HXT. 0001 = Clock source from LXT. 0010 = Clock source from HIRC. 0011 = Clock source from LIRC. 0100 = Clock source from CA-PLL/4. 0101 = Clock source from SYS-PLL. 0110 = Clock source from DDR -PLL. 0111 = Clock source from EPLL/2. 1000 = Clock source from APLL. 1001 = Clock source from VPLL. Others = Reserved.
[23:20]	**Reserved**	Reserved.
[19]	**CANFD3SEL**	**CAN FD 3 Clock Source Selection**

		0 = Clock source from APLL clock. 1 = Clock source from VPLL clock.
[18]	**CANFD2SEL**	**CAN FD 2 Clock Source Selection** 0 = Clock source from APLL clock. 1 = Clock source from VPLL clock.
[17]	**CANFD1SEL**	**CAN FD 1 Clock Source Selection** 0 = Clock source from APLL clock. 1 = Clock source from VPLL clock.
[16]	**CANFD0SEL**	**CAN FD 0 Clock Source Selection** 0 = Clock source from APLL clock. 1 = Clock source from VPLL clock.
[15:14]	**I2S1SEL**	**I2S1 Clock Source Selection** 00 = Clock source from APLL clock. 01 = Clock source from SYSCLK1/2 clock. Others = Reserved.
[13:12]	**I2S0SEL**	**I2S0 Clock Source Selection** 00 = Clock source from APLL clock. 01 = Clock source from SYSCLK1/2 clock. Others = Reserved.
[11:10]	**QSPI1SEL**	**QSPI1 Clock Source Selection** 00 = Clock source from PCLK0. 01 = Clock source from APLL. Others = Reserved.
[9:8]	**QSPI0SEL**	**QSPI0 Clock Source Selection** 00 = Clock source from PCLK0. 01 = Clock source from APLL. Others = Reserved.
[7:6]	**SPI3SEL**	**SPI3 Clock Source Selection** 00 = Clock source from PCLK2. 01 = Clock source from APLL. Others = Reserved.
[5:4]	**SPI2SEL**	**SPI2 Clock Source Selection** 00 = Clock source from PCLK1. 01 = Clock source from APLL. Others = Reserved.
[3:2]	**SPI1SEL**	**SPI1 Clock Source Selection** 00 = Clock source from PCLK2. 01 = Clock source from APLL. Others = Reserved.
[1:0]	**SPI0SEL**	**SPI0 Clock Source Selection** 00 = Clock source from PCLK1. 01 = Clock source from APLL. Others = Reserved.

Clock Divider Number Register 0 (CLK_CLKDIV0)

Register	Offset	R/W	Description	Reset Value
CLK_CLKDIV0	CLK_BA+0x2C	R/W	Clock Divider Number Register 0	0x0000_0000

31	30	29	28	27	26	25	24
GMAC1DIV		GMAC0DIV		Reserved	ACLK0DIV	Reserved	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved					DCUPDIV		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	CANFD3DIV			Reserved	CANFD2DIV		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	CANFD1DIV			Reserved	CANFD0DIV		

Bits	Description	
[31:30]	**GMAC1DIV**	**GMAC1 TX Clock Divide Selection From EPLL Clock Source (Read Only)** 00 = Clock source from EPLL /2.(RGMII). 01 = Clock source from EPLL /2.(RGMII). 10 = Clock source from EPLL /100 (RGMII). 11 = Clock source from EPLL /10 (RGMII). X0 = Clock source from RMII reference clock /20 (RMII). X1 = Clock source from RMII reference clock /2 (RMII). Others = Reserved. **Note:** This field definition depends on the Ethernet mac is RMII or not.
[29:28]	**GMAC0DIV**	**GMAC0 TX Clock Divide Selection From EPLL Clock Source (Read Only)** 00 = Clock source from EPLL /2.(RGMII). 01 = Clock source from EPLL /2.(RGMII). 10 = Clock source from EPLL /100 (RGMII). 11 = Clock source from EPLL /10 (RGMII). X0 = Clock source from RMII reference clock /20 (RMII). X1 = Clock source from RMII reference clock /2 (RMII). Others = Reserved. **Note:** This field definition depends on the Ethernet mac is RMII or not.
[27]	**Reserved**	Reserved.
[26]	**ACLK0DIV**	**AXI0 Bus Clock Divide Number From CA-PLL Clock Source (Write Protect)** 0 = Clock source from CA-PLL /2. 1 = Clock source from CA-PLL /4. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[25:19]	**Reserved**	Reserved.
[18:16]	**DCUPDIV**	**Display Controller Ultra Pixel Divided Clock Source (Write Protect)** DCUP clock frequency = (DCUP clock source frequency) / (2*(DCUPDIV+1)).

[15]	**Reserved**	Reserved.
[14:12]	**CANFD3DIV**	**CAN FD 3 Clock Divide Number From CAN FD 3 Clock Source** CAN FD 3 clock frequency = (CANFD3 clock source frequency) / (2*(CANFD3DIV+1)).
[11]	**Reserved**	Reserved.
[10:8]	**CANFD2DIV**	**CAN FD 2 Clock Divide Number From CAN FD 2 Clock Source** CAN FD 2 clock frequency = (CANFD2 clock source frequency) / (2*(CANFD2DIV+1)).
[7]	**Reserved**	Reserved.
[6:4]	**CANFD1DIV**	**CAN FD 1 Clock Divide Number From CAN FD 1 Clock Source** CAN FD 1 clock frequency = (CANFD1 clock source frequency) / (2*(CANFD1DIV+1)).
[3]	**Reserved**	Reserved.
[2:0]	**CANFD0DIV**	**CAN FD 0 Clock Divide Number From CAN FD 0 Clock Source** CAN FD 0 clock frequency = (CANFD0 clock source frequency) / (2*(CANFD0DIV+1)).

Clock Divider Number Register 1 (CLK_CLKDIV1)

Register	Offset	R/W	Description	Reset Value
CLK_CLKDIV1	CLK_BA+0x30	R/W	Clock Divider Number Register 1	0x0000_0000

31	30	29	28	27	26	25	24
UART3DIV				UART2DIV			
23	22	21	20	19	18	17	16
UART1DIV				UART0DIV			
15	14	13	12	11	10	9	8
CCAP1DIV				CCAP0DIV			
7	6	5	4	3	2	1	0
SC1DIV				SC0DIV			

Bits	Description	
[31:28]	**UART3DIV**	**UART3 Clock Divide Number From UART3 Clock Source** UART3 clock frequency = (UART3 clock source frequency) / (UART3DIV + 1).
[27:24]	**UART2DIV**	**UART2 Clock Divide Number From UART2 Clock Source** UART2 clock frequency = (UART2 clock source frequency) / (UART2DIV + 1).
[23:20]	**UART1DIV**	**UART1 Clock Divide Number From UART1 Clock Source** UART1 clock frequency = (UART1 clock source frequency) / (UART1DIV + 1).
[19:16]	**UART0DIV**	**UART0 Clock Divide Number From UART0 Clock Source** UART0 clock frequency = (UART0 clock source frequency) / (UART0DIV + 1).
[15:12]	**CCAP1DIV**	**CCAP1 Clock Divide Number From CCAP1 Clock Source (TZNS)** CCAP1 clock frequency = (CCAP1 clock source frequency) / (CCAP1DIV + 1).
[11:8]	**CCAP0DIV**	**CCAP0 Clock Divide Number From CCAP0 Clock Source (TZNS)** CCAP0 clock frequency = (CCAP0 clock source frequency) / (CCAP0DIV + 1).
[7:4]	**SC1DIV**	**Smart Card 1 Clock Divide Number From SC1 Clock Source** SC1 clock frequency = (SC1 clock source frequency) / (SC1DIV + 1).
[3:0]	**SC0DIV**	**Smart Card 0 Clock Divide Number From SC0 Clock Source** SC0 clock frequency = (SC0 clock source frequency) / (SC0DIV + 1).

Clock Divider Number Register 2 (CLK_CLKDIV2)

Register	Offset	R/W	Description	Reset Value
CLK_CLKDIV2	CLK_BA+0x34	R/W	Clock Divider Number Register 2	0x0000_0000

31	30	29	28	27	26	25	24
UART11DIV				UART10DIV			
23	22	21	20	19	18	17	16
UART9DIV				UART8DIV			
15	14	13	12	11	10	9	8
UART7DIV				UART6DIV			
7	6	5	4	3	2	1	0
UART5DIV				UART4DIV			

Bits	Description	
[31:28]	**UART11DIV**	**UART11 Clock Divide Number From UART11 Clock Source** UART11 clock frequency = (UART11 clock source frequency) / (UART11DIV + 1).
[27:24]	**UART10DIV**	**UART10 Clock Divide Number From UART10 Clock Source** UART10 clock frequency = (UART10 clock source frequency) / (UART10DIV + 1).
[23:20]	**UART9DIV**	**UART9 Clock Divide Number From UART9 Clock Source** UART9 clock frequency = (UART9 clock source frequency) / (UART9DIV + 1).
[19:16]	**UART8DIV**	**UART8 Clock Divide Number From UART8 Clock Source** UART8 clock frequency = (UART8 clock source frequency) / (UART8DIV + 1).
[15:12]	**UART7DIV**	**UART7 Clock Divide Number From UART7 Clock Source** UART7 clock frequency = (UART7 clock source frequency) / (UART7DIV + 1).
[11:8]	**UART6DIV**	**UART6 Clock Divide Number From UART6 Clock Source** UART6 clock frequency = (UART6 clock source frequency) / (UART6DIV + 1).
[7:4]	**UART5DIV**	**UART5 Clock Divide Number From UART5 Clock Source** UART5 clock frequency = (UART5 clock source frequency) / (UART5DIV + 1).
[3:0]	**UART4DIV**	**UART4 Clock Divide Number From UART4 Clock Source** UART4 clock frequency = (UART4 clock source frequency) / (UART4DIV + 1).

Clock Divider Number Register 3 (CLK_CLKDIV3)

Register	Offset	R/W	Description	Reset Value
CLK_CLKDIV3	CLK_BA+0x38	R/W	Clock Divider Number Register 3	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				UART16DIV			
15	14	13	12	11	10	9	8
UART15DIV				UART14DIV			
7	6	5	4	3	2	1	0
UART13DIV				UART12DIV			

Bits	Description	
[31:20]	**Reserved**	Reserved.
[19:16]	**UART16DIV**	**UART16 Clock Divide Number From UART16 Clock Source** UART16 clock frequency = (UART16 clock source frequency) / (UART16DIV + 1).
[15:12]	**UART15DIV**	**UART15 Clock Divide Number From UART15 Clock Source** UART15 clock frequency = (UART15 clock source frequency) / (UART15DIV + 1).
[11:8]	**UART14DIV**	**UART14 Clock Divide Number From UART14 Clock Source** UART14 clock frequency = (UART14 clock source frequency) / (UART14DIV + 1).
[7:4]	**UART13DIV**	**UART13 Clock Divide Number From UART13 Clock Source** UART13 clock frequency = (UART13 clock source frequency) / (UART13DIV + 1).
[3:0]	**UART12DIV**	**UART12 Clock Divide Number From UART12 Clock Source** UART12 clock frequency = (UART12 clock source frequency) / (UART12DIV + 1).

Clock Divider Number Register 4 (CLK_CLKDIV4)

Register	Offset	R/W	Description	Reset Value
CLK_CLKDIV4	CLK_BA+0x3C	R/W	Clock Divider Number Register 4	0x0000_0000

31	30	29	28	27	26	25	24
KPIDIV							
23	22	21	20	19	18	17	16
Reserved			ADCDIV				
15	14	13	12	11	10	9	8
ADCDIV							
7	6	5	4	3	2	1	0
ADCDIV				EADCDIV			

Bits	Description	
[31:24]	**KPIDIV**	**Keypad Interface Clock Divide Number from KPI Clock Source** KPI clock frequency = (KPI clock source frequency) / (KPIDIV + 1).
[23:21]	**Reserved**	Reserved.
[20:4]	**ADCDIV**	**ADC Clock Divide Number from ADC Clock Source** ADC clock frequency = (ADC clock source frequency) / (2*(ADCDIV+1)).
[3:0]	**EADCDIV**	**EADC Clock Divide Number from EADC Clock Source** EADC clock frequency = (EADC clock source frequency) / (2*(EADCDIV+1)).

Clock Output Control Register (CLK_CLKOCTL)

Register	Offset	R/W	Description	Reset Value
CLK_CLKOCTL	CLK_BA+0x40	R/W	Clock Output Control Register (Write Protect)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		DIV1EN	CLKOEN	FREQSEL			

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**DIV1EN**	**Clock Output Divide One Enable Bit** 0 = Clock Output will output clock with source frequency divided by FREQSEL. 1 = Clock Output will output clock with source frequency.
[4]	**CLKOEN**	**Clock Output Enable Bit** 0 = Clock Output function Disabled. 1 = Clock Output function Enabled.
[3:0]	**FREQSEL**	**Clock Output Frequency Selection** The formula of output frequency is $\mathbf{F_{out} = F_{in}/2^{(N+1)}}$. F_{in} is the input clock frequency. F_{out} is the frequency of divider output clock. N is the 4-bit value of FREQSEL [3:0].

Clock Status Monitor Register (CLK_STATUS)

The bits in this register are used to monitor if the chip clock source is stable or not, and whether the clock switch is failed.

Register	Offset	R/W	Description	Reset Value
CLK_STATUS	CLK_BA+0x50	R	Clock Status Monitor Register	0x0000_00XX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				VPLLSTB	APLLSTB	EPLLSTB	DDRPLLSTB
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	CAPLLSTB	Reserved	HIRCSTB	LIRCSTB	SYSPLLSTB	LXTSTB	HXTSTB

Bits	Description	
[31:12]	**Reserved**	Reserved.
[11]	**VPLLSTB**	**VPLL Clock Source Stable Flag (Read Only)** 0 = VPLL clock is not stable or disabled. 1 = VPLL clock is stable and enabled.
[10]	**APLLSTB**	**APLL Clock Source Stable Flag (Read Only)** 0 = APLL clock is not stable or disabled. 1 = APLL clock is stable and enabled.
[9]	**EPLLSTB**	**EPLL Clock Source Stable Flag (Read Only)** 0 = EPLL clock is not stable or disabled. 1 = EPLL clock is stable and enabled.
[8]	**DDRPLLSTB**	**DDR-PLL Clock Source Stable Flag (Read Only)** 0 = DDR-PLL clock is not stable or disabled. 1 = DDR-PLL clock is stable and enabled.
[7]	**Reserved**	Reserved.
[6]	**CAPLLSTB**	**Cortex35 PLL Clock Source Stable Flag (Read Only)** 0 = CA-PLL clock is not stable or disabled. 1 = CA-PLL clock is stable and enabled.
[5]	**Reserved**	Reserved.
[4]	**HIRCSTB**	**HIRC Clock Source Stable Flag (Read Only)** 0 = 12 MHz internal high speed RC oscillator (HIRC) clock is not stable or disabled. 1 = 12 MHz internal high speed RC oscillator (HIRC) clock is stable and enabled.
[3]	**LIRCSTB**	**LIRC Clock Source Stable Flag (Read Only)** 0 = 32 kHz internal low speed RC oscillator (LIRC) clock is not stable or disabled.

		1 = 32 kHz internal low speed RC oscillator (LIRC) clock is stable and enabled.
[2]	**SYSPLLSTB**	**SYS-PLL Clock Source Stable Flag (Read Only)** 0 = SYS-PLL clock is not stable or disabled. 1 = SYS-PLL clock is stable and enabled.
[1]	**LXTSTB**	**LXT Clock Source Stable Flag (Read Only)** LXT clock source can be selected as external LXT or LIRC32 by setting C32KS(RTC_LXTCTL[6]). If C32KS is set to 0 the LXT stable flag is set when external LXT clock source is stable. If C32KS is set to 1 the LXT stable flag is set when LIRC32 clock source is stable. 0 = 32.768 kHz external low speed crystal oscillator (LXT) clock is not stable or disabled. 1 = 32.768 kHz external low speed crystal oscillator (LXT) clock is stabled and enabled.
[0]	**HXTSTB**	**HXT Clock Source Stable Flag (Read Only)** 0 = 24 MHz external high speed crystal oscillator (HXT) clock is not stable or disabled. 1 = 24 MHz external high speed crystal oscillator (HXT) clock is stable and enabled.

PLL0 Control Register 0 (CLK_PLL0CTL0)

The PLL reference clock input is from the 24 MHz external high speed crystal oscillator (HXT) clock input. This register is used to control the PLL output frequency and PLL operation mode.

Programming these bits needs to write “59h”, “16h” and “88h” to address 0x4046_01A0 for security registers to disable register protection. Refer to the register SYS_RLKTZS at address SYS_BA+0x1A0.

Programming these bits needs to write “59h”, “16h” and “88h” to address 0x4046_01A4 for non-security registers to disable register protection. Refer to the register SYS_RLKTZNS at address SYS_BA+0x1A4.

CLK_PLL0CTL0 can only be modified while PD(CLK_PLL0CTL0[16]) is set.

Register	Offset	R/W	Description	Reset Value
CLK_PLL0CTL0	CLK_BA+0x60	R/W	CA-PLL Control Register 0(Write Protect)	0x0001_138A

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						BP	PD
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		OUTDIV		INDIV			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FBDIV							

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17]	**BP**	**PLL Bypass Control (Write Protect)** 0 = PLL is in normal mode (default). 1 = PLL clock output is same as PLL input clock FIN. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[16]	**PD**	**Power-down Mode (Write Protect)** If set the PDEN bit to 1 in CLK_PWRCTL register, the PLL will enter Power-down mode, too. 0 = PLL is in normal mode. 1 = PLL is in Power-down mode (default). **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[15:14]	**Reserved**	Reserved.
[13:12]	**OUTDIV**	**PLL Output Divider Control (Write Protect)** Refer to the formulas at PLL Control Mode. This register is referring to the parameter P. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[11:8]	**INDIV**	**PLL Input Divider Control (Write Protect)** Refer to the formulas at PLL Control Mode. This register is referring to the parameter M.

		Note: This bit is write protected. Refer to the SYS_RLKTZS register.
[7:0]	**FBDIV**	**PLL Feedback Divider Control (Write Protect)** Refer to the formulas at PLL Control Mode. This register is referring to the parameter N. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.

PLLx Control Register 0 (CLK_PLLxCTL0)

The PLL reference clock input is from the 24 MHz external high speed crystal oscillator (HXT) clock input. This register is used to control the PLL output frequency and PLL operation mode.

Programming these bits needs to write “59h”, “16h” and “88h” to address 0x4046_01A0 for security registers to disable register protection. Refer to the register SYS_RLKTZS at address SYS_BA+0x1A0.

Programming these bits needs to write “59h”, “16h” and “88h” to address 0x4046_01A4 for non-security registers to disable register protection. Refer to the register SYS_RLKTZNS at address SYS_BA+0x1A4.

CLK_PLLxCTL0 can only be modified while PD(CLK_PLLxCTL1[0]) is set.

Register	Offset	R/W	Description	Reset Value
CLK_PLL2CTL0	CLK_BA+0x80	R/W	DDR-PLL Control Register 0(Write Protect)	0x0F04_102C
CLK_PLL3CTL0	CLK_BA+0x90	R/W	APLL Control Register 0(Write Protect)	0x0000_0000
CLK_PLL4CTL0	CLK_BA+0xA0	R/W	EPLL Control Register 0(Write Protect)	0x0000_60FA
CLK_PLL5CTL0	CLK_BA+0xB0	R/W	VPLL Control Register 0(Write Protect)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	SSRATE						
23	22	21	20	19	18	17	16
SSRATE				MODE		INDIV	
15	14	13	12	11	10	9	8
INDIV				Reserved	FBDIV		
7	6	5	4	3	2	1	0
FBDIV							

Bits	Description	
[31]	**Reserved**	Reserved.
[30:20]	**SSRATE**	**Spreading Frequency Control (Write Protect)** Set the spread step factor SSRATE from 0 to 2047, SSRATE = SSRATE [10:0]. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register. For PLL3, PLL4, and PLL5, user can also refer to the SYS_RLKTZNS register if SYSSIAEN is 1.
[19:18]	**MODE**	**Operation Mode Selection (Write Protect)** 00 = Integer mode. In this mode, the rising edges of the two clocks at the input of PFD are phase aligned. And the output clock frequency is at multiples of the input clock frequency contingent on the configuration of OUTDIV, INDIV and FBDIV. Refer to the formulas at Advanced PLL Integer Mode. 01 = Fractional mode. This mode is suitable for applications which need small output frequency steps, like 20 kHz. The jitter performance in this mode may be worse than in Integer Mode. In this mode, the output clock frequency is at the fractional multiples of the input clock frequency. By setting the control pins FRAC [23:0], a small output frequency step is achieved. Refer to the formulas at Advanced

		PLL Fractional Mode. 10 = Spread Spectrum Mode. This mode is suitable for in output frequency of PLL will be modulated by triangle wave. It is for EMI consideration. By setting SSRATE [10:0] and SLOPE [23:0], the modulation index and the modulation frequency can be programmed. Refer to the formulas at Advanced PLL Spread Spectrum Mode. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register. For PLL3, PLL4, and PLL5, user can also refer to the SYS_RLKTZNS register if SYSSIAEN is 1.
[17:12]	**INDIV**	**PLL Reference Input Divider Control (Write Protect)** Set the reference divider factor (M) from 1 to 63. The M = INDIV[5:0]. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register. For PLL3, PLL4, and PLL5, user can also refer to the SYS_RLKTZNS register if SYSSIAEN is 1.
[11]	**Reserved**	Reserved.
[10:0]	**FBDIV**	**PLL Feedback Divider Control (Write Protect)** Set the feedback divider factor (N) from 16 to 2047. The N = FBDIV[10:0]. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register. For PLL3, PLL4, and PLL5, user can also refer to the SYS_RLKTZNS register if SYSSIAEN is 1.

PLLx Control Register 1 (CLK_PLLxCTL1)

CLK_PLLxCTL1 can only be modified while PD(CLK_PLLxCTL1[0]) is set.

Register	Offset	R/W	Description	Reset Value
CLK_PLL2CTL1	CLK_BA+0x84	R/W	DDR-PLL Control Register 1(Write Protect)	0x6B85_1E41
CLK_PLL3CTL1	CLK_BA+0x94	R/W	APLL Control Register 1(Write Protect)	0x0000_0001
CLK_PLL4CTL1	CLK_BA+0xA4	R/W	EPLL Control Register 1(Write Protect)	0x0000_0021
CLK_PLL5CTL1	CLK_BA+0xB4	R/W	VPLL Control Register 1(Write Protect)	0x0000_0001

31	30	29	28	27	26	25	24
FRAC							
23	22	21	20	19	18	17	16
FRAC							
15	14	13	12	11	10	9	8
FRAC							
7	6	5	4	3	2	1	0
Reserved	OUTDIV			Reserved		BP	PD

Bits	Description	
[31:8]	**FRAC**	**PLL Fractional Portion of DN Value (Write Protect)** Set the fraction part (X) of Fractional Portion of DN Value factor. The X = FRAC[23:0] / 2^{24}. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register. For PLL3, PLL4, and PLL5, user can also refer to the SYS_RLKTZNS register if SYSSIAEN is 1.
[7]	**Reserved**	Reserved.
[6:4]	**OUTDIV**	**PLL Output Divider Control (Write Protect)** Set the output divider factor (P) from 1 to 7. P = OUTDIV[2:0]. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register. For PLL3, PLL4, and PLL5, user can also refer to the SYS_RLKTZNS register if SYSSIAEN is 1.
[3:2]	**Reserved**	Reserved.
[1]	**BP**	**PLL Bypass Control (Write Protect)** 0 = PLL is in normal mode (default). 1 = PLL clock output is same as PLL input clock Fref. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register. For PLL3, PLL4, and PLL5, user can also refer to the SYS_RLKTZNS register if SYSSIAEN is 1.
[0]	**PD**	**Power-down Mode (Write Protect)** 0 = PLL is enable (in normal mode). 1 = PLL is disable (in Power-down mode) (default). **Note:** This bit is write protected. Refer to the SYS_RLKTZS register. For PLL3, PLL4, and PLL5, user can also refer to the SYS_RLKTZNS register if SYSSIAEN is 1.

PLLx Control Register 2 (CLK_PLLxCTL2)

CLK_PLLxCTL2 can only be modified while PD(CLK_PLLxCTL1[0]) is set.

Register	Offset	R/W	Description	Reset Value
CLK_PLL2CTL2	CLK_BA+0x88	R/W	DDR-PLL Control Register 2(Write Protect)	0x0000_48A3
CLK_PLL3CTL2	CLK_BA+0x98	R/W	APLL Control Register 2(Write Protect)	0x0000_0000
CLK_PLL4CTL2	CLK_BA+0xA8	R/W	EPLL Control Register 2(Write Protect)	0x0000_0000
CLK_PLL5CTL2	CLK_BA+0xB8	R/W	VPLL Control Register 2(Write Protect)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SLOPE							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SLOPE							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SLOPE							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**SLOPE**	**PLL Stable Counter Selection (Write Protect)** Set the spread step factor SLOPE from 0 to 16777215, SLOPE = SLOPE[23:0]. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register. For PLL3, PLL4, and PLL5, user can also refer to the SYS_RLKTZNS register if SYSSIAEN is 1.

Clock Fail Detector Control Register (CLK_CLKDCTL)

Register	Offset	R/W	Description	Reset Value
CLK_CLKDCTL	CLK_BA+0xC0	R/W	Clock Fail Detector Control Register(Write Protect)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved						HXTFQIEN	HXTFQDEN
15	14	13	12	11	10	9	8
Reserved		LXTFIEN	LXTFDEN	Reserved			
7	6	5	4	3	2	1	0
Reserved		HXTFIEN	HXTFDEN	Reserved			

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17]	**HXTFQIEN**	**HXT Clock Frequency Monitor Interrupt Enable Bit** 0 = 24 MHz external high speed crystal oscillator (HXT) clock frequency monitor fail interrupt Disabled. 1 = 24 MHz external high speed crystal oscillator (HXT) clock frequency monitor fail interrupt Enabled.
[16]	**HXTFQDEN**	**HXT Clock Frequency Monitor Enable Bit** 0 = 24 MHz external high speed crystal oscillator (HXT) clock frequency monitor Disabled. 1 = 24 MHz external high speed crystal oscillator (HXT) clock frequency monitor Enabled.
[15:14]	**Reserved**	Reserved.
[13]	**LXTFIEN**	**LXT Clock Fail Interrupt Enable Bit** 0 = 32.768 kHz external low speed crystal oscillator (LXT) clock fail interrupt Disabled. 1 = 32.768 kHz external low speed crystal oscillator (LXT) clock fail interrupt Enabled.
[12]	**LXTFDEN**	**LXT Clock Fail Detector Enable Bit** 0 = 32.768 kHz external low speed crystal oscillator (LXT) clock fail detector Disabled. 1 = 32.768 kHz external low speed crystal oscillator (LXT) clock fail detector Enabled.
[11:6]	**Reserved**	Reserved.
[5]	**HXTFIEN**	**HXT Clock Fail Interrupt Enable Bit** 0 = 24 MHz external high speed crystal oscillator (HXT) clock fail interrupt Disabled. 1 = 24 MHz external high speed crystal oscillator (HXT) clock fail interrupt Enabled.
[4]	**HXTFDEN**	**HXT Clock Fail Detector Enable Bit** 0 = 24 MHz external high speed crystal oscillator (HXT) clock fail detector Disabled. 1 = 24 MHz external high speed crystal oscillator (HXT) clock fail detector Enabled.
[3:0]	**Reserved**	Reserved.

Clock Fail Detector Status Register (CLK_CLKDSTS)

Register	Offset	R/W	Description	Reset Value
CLK_CLKDSTS	CLK_BA+0xC4	R/W	Clock Fail Detector Status Register(Write Protect)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							HXTFQIF
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						LXTFIF	HXTFIF

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8]	**HXTFQIF**	**HXT Clock Frequency Monitor Interrupt Flag (Write Protect, Write 1 to Clear)** 0 = 24 MHz external high speed crystal oscillator (HXT) clock is normal. 1 = 24 MHz external high speed crystal oscillator (HXT) clock frequency is abnormal. **Note 1:** Write 1 to clear the bit to 0. **Note 2:** This bit is write protected. Refer to the SYS_RLKTZS register.
[7:2]	**Reserved**	Reserved.
[1]	**LXTFIF**	**LXT Clock Fail Interrupt Flag (Write Protect, Write 1 to Clear)** 0 = 32.768 kHz external low speed crystal oscillator (LXT) clock is normal. 1 = 32.768 kHz external low speed crystal oscillator (LXT) stops. **Note 1:** Write 1 to clear the bit to 0. **Note 2:** This bit is write protected. Refer to the SYS_RLKTZS register.
[0]	**HXTFIF**	**HXT Clock Fail Interrupt Flag (Write Protect, Write 1 to Clear)** 0 = 24 MHz external high speed crystal oscillator (HXT) clock is normal. 1 = 24 MHz external high speed crystal oscillator (HXT) clock stops. **Note 1:** Write 1 to clear the bit to 0. **Note 2:** This bit is write protected. Refer to the SYS_RLKTZS register.

Clock Frequency Detector Upper Boundary Register (CLK_CDUPB)

Register	Offset	R/W	Description	Reset Value
CLK_CDUPB	CLK_BA+0xC8	R/W	Clock Frequency Detector Upper Boundary Register(Write Protect)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						UPERBD	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UPERBD							

Bits	Description	
[31:10]	**Reserved**	Reserved.
[9:0]	**UPERBD**	**HXT Clock Frequency Detector Upper Boundary** The bits define the high value of frequency monitor window. When HXT frequency monitor value higher than this register, the HXT frequency detect fail interrupt flag will set to 1.

Clock Frequency Detector Lower Boundary Register (CLK_CDLOWB)

Register	Offset	R/W	Description	Reset Value
CLK_CDLOWB	CLK_BA+0xCC	R/W	Clock Frequency Detector Lower Boundary Register(Write Protect)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						LOWERBD	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
LOWERBD							

Bits	Description	
[31:10]	**Reserved**	Reserved.
[9:0]	**LOWERBD**	**HXT Clock Frequency Detector Lower Boundary** The bits define the low value of frequency monitor window. When HXT frequency monitor value lower than this register, the HXT frequency detect fail interrupt flag will set to 1.

Clock Filter Control Register (CLK_CKFLTRCTL)

Register	Offset	R/W	Description	Reset Value
CLK_CKFLTRCTL	CLK_BA+0xD0	R/W	Clock Filter Control Register (Write Protect)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved			HIRCGTEN	Reserved		HIRCFLTRSEL	HIRCFLTREN
7	6	5	4	3	2	1	0
Reserved		HXTBYPSEN	HXTGTEN	Reserved		HXTFLTRSEL	HXTFLTREN

Bits	Description	
[31:13]	**Reserved**	Reserved.
[12]	**HIRCGTEN**	**HIRC Clock Gating Enable Control Bit (Write Protect)** 0 = HIRC clock filter function Disabled. 1 = HIRC clock filter function Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[11:10]	**Reserved**	Reserved.
[9]	**HIRCFLTRSEL**	**HIRC Clock Filter Frequency Select** 0 = HIRC frequency is > 12 MHz. 1 = HIRC frequency is <= 12 MHz. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[8]	**HIRCFLTREN**	**HIRC Clock Filter Enable Control Bit (Write Protect)** 0 = HIRC clock filter function Disabled. 1 = HIRC clock filter function Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[7:6]	**Reserved**	Reserved.
[5]	**HXTBYPSEN**	**HXT Clock Bypass Enable Control Bit (Write Protect)** 0 = HXT clock filter function Disabled. 1 = HXT clock filter function Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[4]	**HXTGTEN**	**HXT Clock Gating Enable Control Bit (Write Protect)** 0 = HXT clock filter function Disabled. 1 = HXT clock filter function Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[3:2]	**Reserved**	Reserved.

[1]	**HXTFLTRSEL**	**HXT Clock Filter Frequency Select** 0 = HXT frequency is greater than 24 MHz. 1 = HXT frequency is less than or equal to 24 MHz. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.
[0]	**HXTFLTREN**	**HXT Clock Filter Enable Control Bit (Write Protect)** 0 = HXT clock filter function Disabled. 1 = HXT clock filter function Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register.

6.6 SSPCC

6.6.1 Overview

System security peripheral configuration controller, SSPCC, is used to configure the security attribution of SRAM, GPIO and all other peripherals for Cortex-A35 and Cortex-M4. SSPCC also collects peripherals' security violation response and generates interrupt when violation event occurs.

In addition, SSPCC contains DPM, a module controlling debug access of this chip, and PLM, which uses pre-defined life-cycle stages to provide hardware-based controls controlling the ability of some functions.

6.6.2 Features

- Configures SRAM's security attribution by boundary
- Configures GPIO security attribution pin by pin
- Configures peripherals' security attribution
- Write protect mechanism to security attribution configuration
- Generates secure violation interrupt
- Debug protection mechanism (DPM)
- Product life-cycle management (PLM)

6.6.3 Block Diagram

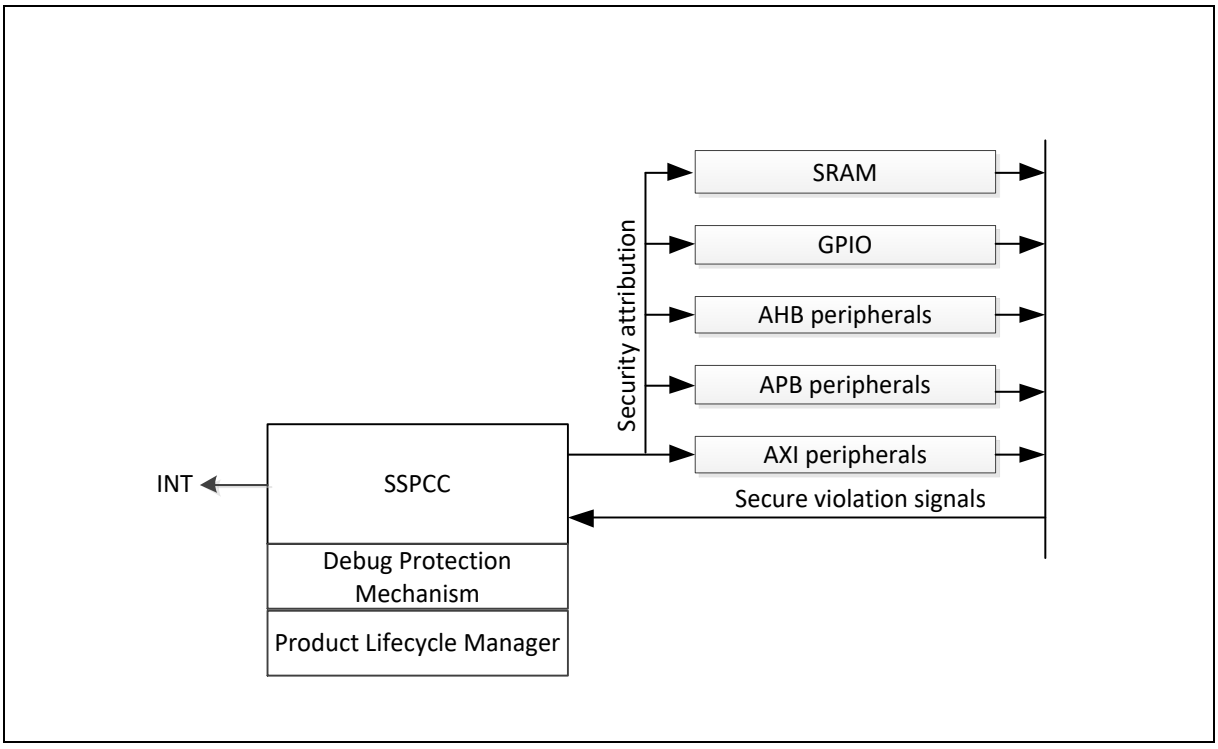


Figure 6.6-1 SSPCC Block Diagram

6.6.4 Functional Description

This chip is a heterogeneous multi-core system, including Cortex-A and Cortex-M core, that has its own required peripherals. In addition, this chip implements Arm TrustZone technique, which separates the system into two worlds, secure world and non-secure world. Combining these two reasons, the whole

system is partitioned to three domains: TrustZone Secure (TZS), TrustZone Non-secure (TZNS) and Cortex-M Subsystem (SubM), as shown in Figure 6.6-2.

Each domain has its own members based on the configuration. Masters in the system can be classified to TZS masters, TZNS masters and SubM masters; while slaves in the system can be classified to TZS slaves, TZNS slaves and SubM slaves. For example, when the Cortex-A35 core is executing in secure state, it is a TZS master. The PDMA that is configured to TZS is a TZS master. The SPI peripheral that is configured to SubM is a SubM slave.

Some slaves do not belong to only one domain but multiple domains are called “hybrid slave” or “hybrid peripheral.” In addition, a type of memory, called ShareMemory, can be accessed by all kinds of masters (TZS, TZNS, SubM). SRAM0 is the case.

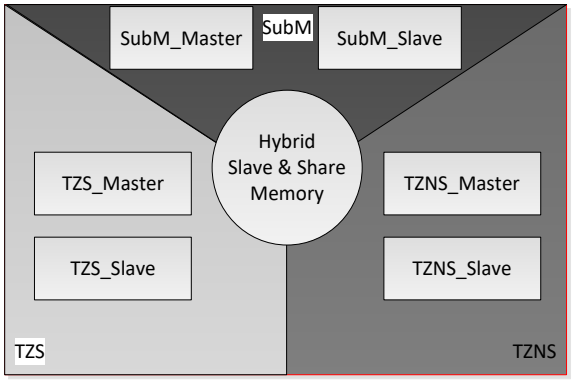


Figure 6.6-2 System Partition Graphical View

6.6.4.1 Security Configuration

The SSPCC is used to configure the security attribution of all peripherals, including both masters and slaves, SRAM and GPIO. The security attribution of all peripherals can be configured by setting SSPCC registers, SSPCC_PSSET0 to SSPCC_PSSET11. In general, each peripheral uses 2 bits to assign the attribution. These 2 bits are TZSn and SubM, as described in Table 6.6-1.

TZSn	SubM	Security Attribution
0	0	The peripheral belongs to TZS domain
1	0	The peripheral belongs to TZNS domain
x	1	The peripheral belongs to SubM domain

Table 6.6-1 Bits for Security Attribution Configuration

Secure code as well as non-secure code can get the security attribution of all peripherals, SRAM regions, GPIOs by reading the corresponding registers: SSPCC_PSSETn, SSPCC_SRAMSB, and SSPCC_IOnSSET.

After powered on, all peripherals and I/Os are in the TZS domain. The peripherals can be set to TZNS or SubM domain by secure code only. After finishing the setting of security configuration, secure code can turn on the write protect mechanism to prevent the security configuration setting to be changed accidently. Refer to Write Protect of Security Configuration section for more details.

Security Attribution of Peripherals

Table 6.6-2 lists the security attribution of peripherals.

Some peripherals can be configured to TZS or TZNS, such as High-speed USB Device (HSUSBD), SD Host (SDH), PDMA0, and PDMA1.

Some peripherals can be configured to TZNS or SubM, such as PDMA2, PDMA3.

Some peripherals are fixed to TZS, such as SSMCC, WDT/WWDT0 , TSI.

Some peripherals are fixed to TZNS domain, such as CCAP0, CCAP1, DCUltra, GFX, VC8000, GMAC0, GMAC1, USBH.

Only WDT/WWDT2 is fixed to SubM domain.

Peripherals	**Security Attribution**	
	Static	Configurable
EBI		TZNS, SubM
HSUSBH0/USBH0	TZNS	
HSUSBH1/USBH1	TZNS	
HSUSBD		TZS, TZNS
NANDC		TZS, TZNS
SDH0		TZS, TZNS
SDH1		TZS, TZNS
CCAP0	TZNS	
CCAP1	TZNS	
GFX	TZNS	
DCUltra	TZNS	
GMAC0	TZNS	
GMAC1	TZNS	
VC8000	TZNS	
PDMA0		TZS, TZNS
PDMA1		TZS, TZNS
PDMA2		TZNS, SubM
PDMA3		TZNS, SubM
Crypto		TZS, TZNS
TRNG		TZS, TZNS
Timer0/1		TZS, TZNS
Timer6/7		TZNS, SubM
UART0		TZS, TZNS
QSPI0		TZS, TZNS
QSPI1		TZNS, SubM
I2S0		TZNS, SubM
I2C0		TZS, TZNS
I2C3		TZNS, SubM
ADC0		TZNS, SubM
Timer2/3		TZNS, SubM
Timer8/9		TZNS, SubM
UART1		TZNS, SubM
UART3		TZNS, SubM
UART5		TZNS, SubM
UART7		TZNS, SubM

UART9		TZNS, SubM
UART11		TZNS, SubM
UART13		TZNS, SubM
UART15		TZNS, SubM
SPI0		TZNS, SubM
SPI2		TZNS, SubM
EPWM0		TZNS, SubM
EPWM2		TZNS, SubM
QEI0		TZNS, SubM
QEI2		TZNS, SubM
ECAP0		TZNS, SubM
ECAP2		TZNS, SubM
I2C1		TZNS, SubM
I2C4		TZNS, SubM
Timer4/5		TZNS, SubM
Timer10/11		TZNS, SubM
UART2		TZNS, SubM
UART4		TZNS, SubM
UART6		TZNS, SubM
UART8		TZNS, SubM
UART10		TZNS, SubM
UART12		TZNS, SubM
UART14		TZNS, SubM
UART16		TZNS, SubM
I2C2		TZNS, SubM
I2C5		TZNS, SubM
SPI1		TZNS, SubM
SPI3		TZNS, SubM
EPWM1		TZNS, SubM
QEI1		TZNS, SubM
I2S1		TZNS, SubM
EADC0		TZNS, SubM
ECAP1		TZNS, SubM
WDT,WWDT0	TZS	
WDT,WWDT1		TZS,TZNS
KPI		TZNS, SubM
DDR PUB (PHY)		TZS, TZNS
MCTL2		TZS, TZNS
SSMCC	TZS	
WDT,WWDT2	SubM	
CAN FD 0		TZNS, SubM
CAN FD 1		TZNS, SubM
CAN FD 2		TZNS, SubM

CAN FD 3		TZNS, SubM
SC0		TZNS, SubM
SC1		TZNS, SubM

Table 6.6-2 List of Security Attribution of Peripherals

In addition to peripheral listed above, some peripherals are hybrid peripherals, which containing registers across plural domains, such as: System, RTC, Wormhole, HWSEM, SSPCC. Refer to "Security Attribution of Hybrid Peripheral and Special Register" for more details.

Security Attribution of GPIO

The security attribution of GPIO can be configured pin by pin through SSPCC_IOnSSET registers, where n = A, B, C… to N.

Note that when the I/O is used as a multi-function pin and routed to other peripheral, the secure software should take the responsibility to make the security attribution consistent between the GPIO pin, by setting SSPCC_IOnSSET, and target peripheral, by setting SSPCC_PSSETn.

Some registers in GPIO is controlled by port basis, such as GPx_DBCTL, Clock enable, Reset registers. The security attribution of these kinds of registers follows the rule defined in Table 6.6-3. For example, if GPA[0] is set to TZS, no matter what security attribution of GPA[1:15] is, the GPA_DBCTL register is TZS and accepts TZS master's access only.

Attribution Of GPIO Bits	**Attribution Of GPIO Port**
At least 1 TZS pin(s)	TZS
All SubM pins	SubM
TZNS + SubM pins	TZNS

Table 6.6-3 Security Attribution of GPIO Register by Port Basis

Security Attribution of SRAM

SRAM security attribution is partitioned by a 16-KB aligned boundary. By default, all SRAM1 region is TZS and can be set to TZNS, SRAM0 is ShareMemory and can be set to SubM. Secure software can set the boundary to SRAM0, SRAM1 through the register, SSPCC_SRAMSB.

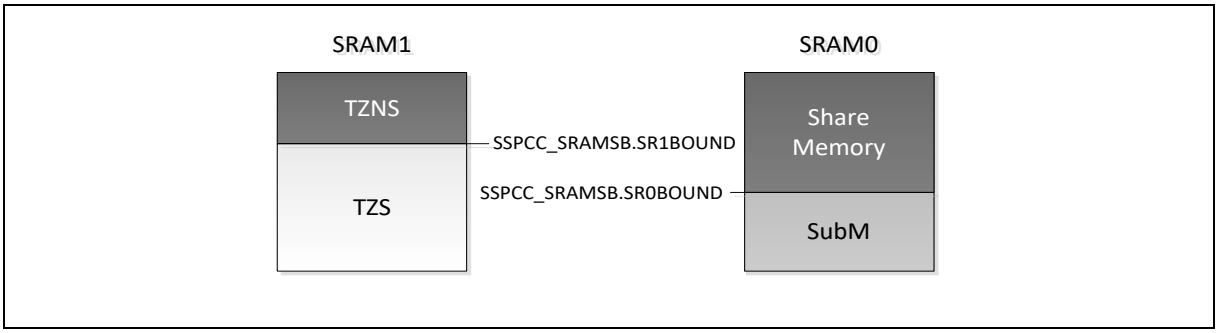


Figure 6.6-3 SRAM Security Attribution Configuration

Security Attribution of Hybrid Peripheral and Special Register

In hybrid peripherals, there are registers with additional access properties. Moreover, controlled by SINFAEN registers, some registers' access properties will be changed. For example, when SINFAEN.SSPCC is 1, the SSPCC_PSSETn registers accept TZNS master's read access, while when SINFAEN.SSPCC is 0, the SSPCC_PSSETn registers do not accept any of TZNS master's access.

Refer to the following sections for accessing these hybrid peripherals.

Access rule of SSPCC registers

Most of SSPCC registers are TZS registers, which accept TZS master's access only.

Apart from TZS registers, there are some registers accepting TZNS master's access, as listed in Table 6.6-4. Note that the accessibility is controlled by SINFAEN.SSPCCSIAEN bit.

SSPCC			
SINFAEN.SSPCC = 1	TZS Access	TZNS Access	SubM Access
Read Only	PLM_STS	PLM_STS, SSPCC_PSSET0~11, SSPCC_SRAMSB, SSPCC_EBISSET, SSPCC_IOASSET~SSPCC_IONSSET	
Read Write	SSPCC_PSSET0~SSPCC_PSSET11, SSPCC_SRAMSB, SSPCC_EBISSET, SSPCC_IOASSET~SSPCC_IONSSET		
No Access			PLM_STS, SSPCC_PSSET0~11, SSPCC_SRAMSB, SSPCC_EBISSET, SSPCC_IOASSET~SSPCC_IONSSET
SINFAEN.SSPCC = 0	TZS Access	TZNS Access	SubM Access
Read Only	PLM_STS	PLM_STS	
Read Write	SSPCC_PSSET0~11, SSPCC_SRAMSB, SSPCC_EBISSET, SSPCC_IOASSET~SSPCC_IONSSET		
No Access		SSPCC_PSSET0~11, SSPCC_SRAMSB, SSPCC_EBISSET, SSPCC_IOASSET~SSPCC_IONSSET	PLM_STS, SSPCC_PSSET0~11, SSPCC_SRAMSB, SSPCC_EBISSET, SSPCC_IOASSET~SSPCC_IONSSET

Table 6.6-4 SSPCC Special Registers

Access rule of RTC registers

Refer to the "RTC Control Registers Access Attribute" sub-section in the RTC section.

Note that RTC Clock Enable register has similar security attribution.

Access rule of System Manager and Clock Controller registers

System Manager and Clock Controller have many registers for TZS, TZNS and SubM master to access. Refer to the "Control Registers Access Attribute" sub-section in the System Manager section and Clock Controller section.

Security Attribution of Clock Enable and Reset Bit

To facilitate setting clock and reset of peripherals, the security attribution of clock enable bits, and reset bits in registers of System manager and Clock controller is the same as the corresponding peripherals.

For example, when PDMA0 is set to TZNS, the security attribution of PDMA0EN bit in CLK_SYSCLK1 register becomes TZNS and the memory access policy follows the rule of the peripheral type equal to "Hybrid – TZNS register" in Table 6.6-5 and Table 6.6-6.

Clock enable bit and reset bit of SSPCC are always TZS.

Clock enable bit of RTC, RTC_CKEN, follows the access policy defined in RTC. Clock enable bits and reset bits of GPIO are controlled by port basis, so the security attribution of these

bits follows the rule defined in Table 6.6-3.

6.6.4.2 Memory Access Policy

The memory access policy, MAP, defines the access rule of master of each domain to the target slave. Also, MAP defines what and how to report information when a violation access occurs.

In general, TZS master has the highest authority to access both TZS, TZNS slaves and some Hybrid slave, but SubM system slaves are not able to be accessed, which is in an isolated sub-region.

More details about memory access policy are shown in Table 6.6-5 and Table 6.6-6.

Master	Peripheral Type	Write	Read	Bus Error And Interrupt
TZS	TZS	✓	✓	No
	TZNS	✓	✓	No
	Hybrid – TZS and TZNS register	✓	✓	No
	Hybrid – SubM register	WI	RAZ	No
	Hybrid – Shared register	✓	✓	No
	ShareMemory	✓	✓	No
	SubM	WI	RAZ	No
TZNS	TZS	WI	RAZ	Yes
	TZNS	✓	✓	No
	Hybrid – TZS register	WI	RAZ ✓ (Refer to share list)	No
	Hybrid – TZNS register	✓	✓	No
	Hybrid – SubM register	WI	RAZ	No
	Hybrid – Shared register	✓	✓	No
	ShareMemory	✓	✓	No
	SubM	WI	RAZ	No

Table 6.6-5 Security Memory Access Policy (Master is in TrustZone Domain)

Master	Peripheral Type	Write	Read	Bus Error And Interrupt
SubM	TZS	WI	RAZ	No
	TZNS	WI	RAZ	No

	Hybrid – TZS register	WI	RAZ	No
	Hybrid – TZNS register	WI	RAZ	No
	Hybrid – SubM register	✓	✓	No
	Hybrid – Shared register	✓	✓	No
	ShareMemory	✓	✓	No
	SubM	✓	✓	No

Table 6.6-6 Security Memory Access Policy (Master is in SubM Domain)

The access policy of Hybrid peripheral is similar. TZNS master can access TZNS registers in Hybrid peripheral and TZS registers in Hybrid peripheral with limited authority. The TZNS master can only access TZNS registers. SubM registers in Hybrid peripherals accepts access from SubM masters. The hybrid peripherals and registers are listed in the "Security Attribution of Hybrid Peripheral and Special Register" section.

6.6.4.3 Security Violation Interrupt

SSPCC is also the module to provide information about security violation event occurred in the system. Described in Table 6.6-5 and Table 6.6-6, when an event of violating memory access policy occurs, due to improper setting or malicious attack probably, the slave that receives the access which violates memory access policy will issue a violation signal to SSPCC and report the master number of violating master (shown in Table 6.6-7) and violated address information, which will be recorded in SSPCC. Furthermore, the slave will respond error to the master.

If the enable bit of security violation interrupt in SSPCC_SVIOIEN is set, when SSPCC receives the security violation event from the peripheral, it will trigger SSPCC interrupt to the core processor, Cortex-A35. When SSPCC interrupt asserts, Cortex-A35 can access SSPCC_SVINSTS register to determine which slave generates the violation event. Then, it can get violation source from the corresponding violation source registers such as SSPCC_APB0VSRC, SSPCC_APB1VSRC. Also, it can get violation address from the corresponding violation address registers such as SSPCC_APB0VA, SSPCC_APB1VA.

Note that a violation event can be recorded to _VSRC, _VA registers only when the corresponding flag in SSPCC_SVINSTS register is cleared. In other words, if there are a sequence of violation event to the same slave, only the first event will be recorded.

Masters in this chip and the corresponding Master ID are listed in Table 6.6-7

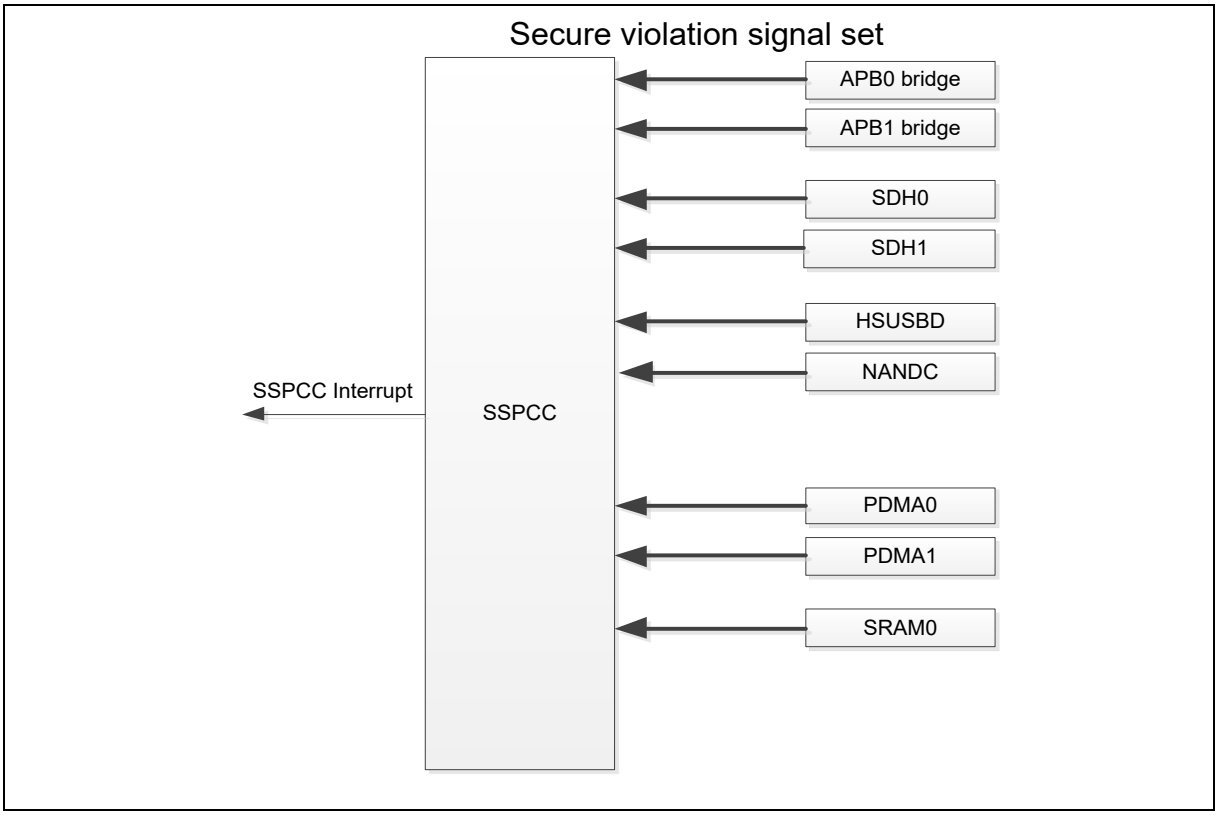


Figure 6.6-4 Security Violation Signal Block Diagram

Master ID	Description
0x00	CA35_0
0x10	CA35_1
0x01	CoreSight AP
0x03	TSI
0x02	CM4
0x04	PDMA0
0x05	PDMA1
0x06	PDMA2
0x07	PDMA3
0x08	SDH0
0x09	SDH1
0x0B	USBH0/HSUSBH0
0x0C	USBH1/HSUSBH1
0x0D	HSUSBD
0x0E	NAND

Table 6.6-7 Master ID Number List

6.6.4.4 Write Protection of Security Configuration

In order to prevent security configuration from accident change causing information leakage and or severe fault, user can turn on the write protection mechanism after setting the security configuration.

The write protection mechanism default to disabled state after reset, and can be turned on by setting SSPCC_SCWP register. Enable bit in SSPCC_SCWP register default to 0, which means the write protection is disabled and security configurations are able to be set. When the security configurations has been initialized, user can set 1 to the enable bit, SSPCC_SCWP[0], with a valid write verify code, to enable write protection. When write protection is enabled, all security configuration registers are not able to be set before unlocking the write protection.

In addition to Enable bit, there is a Lock bit to enable the write protection. When the security configurations has been initialized and user intends to prevent further change of security configuration, the Lock bit of SSPCC_SCWP[1] can be used. This bit, like SSPCC_SCWP[0], makes the ability of setting security configuration disabled. The difference is that this bit can be set to 1 but cannot be cleared to 0. It can only be cleared by a cold reset.

Note that without the correct write-verify code, any write access to SSPCC_SCWP will be ignored.

6.6.4.5 Debug Protection Mechanism

Debug protection mechanism, DPM, controls the whole chip's debug functions, which is basically based on Arm architecture, including invasive and non-invasive debug of Cortex-A35 and Cortex-M4. Debug signals are used to control these debug functions, which are based on the concept of debug state. The debug signal is enabled at some debug states while disabled at other debug states, refer to "Debug State" sub-section for more details.

With software-based authentication algorithm, DPM can provide authority management to each debug function. As shown in Figure 6.6-5, the architecture of DPM includes software and hardware. There are two main parts of software: DPM Firmware and DPM Authentication Algorithm. DPM Firmware provides functions (API) for user's application to manage the hardware, including DPM peripheral and OTP controller. DPM Authentication Algorithm is the user-defined method to do debug authentication. "Example of Authentication Algorithm" sub-section provides example usage of DPM and example of authentication algorithm.

Typically, both DPM Firmware and DPM Authentication Algorithm are secure software so that DPM Firmware have the authority to access DPM IP as well as OTP controller, and the executing of DPM Authentication Algorithm is protected from threat in non-secure world.

Note that software of DPM is not stored in on-chip non-volatile memory but stored in external storage. The software of DPM should be loaded into the chip in the process of boot flow. The integrity and authenticity of DPM software is based on the secure boot mechanism.

The DPM peripheral provides control registers to control all debug functions, such as DPM_A35SDS register indicating the debug state of the secure A35 invasive debug function, DPM_M4DS register indicating the debug state of the M4 invasive debug function. User can read the debug state through these control registers, and also set the REGDEN bit of each debug function through DPM peripheral. On the contrary, the OTP bits of each debug function is read-only in control register in DPM peripheral, and should be programmed through OTP controller that is only achievable by secure user.

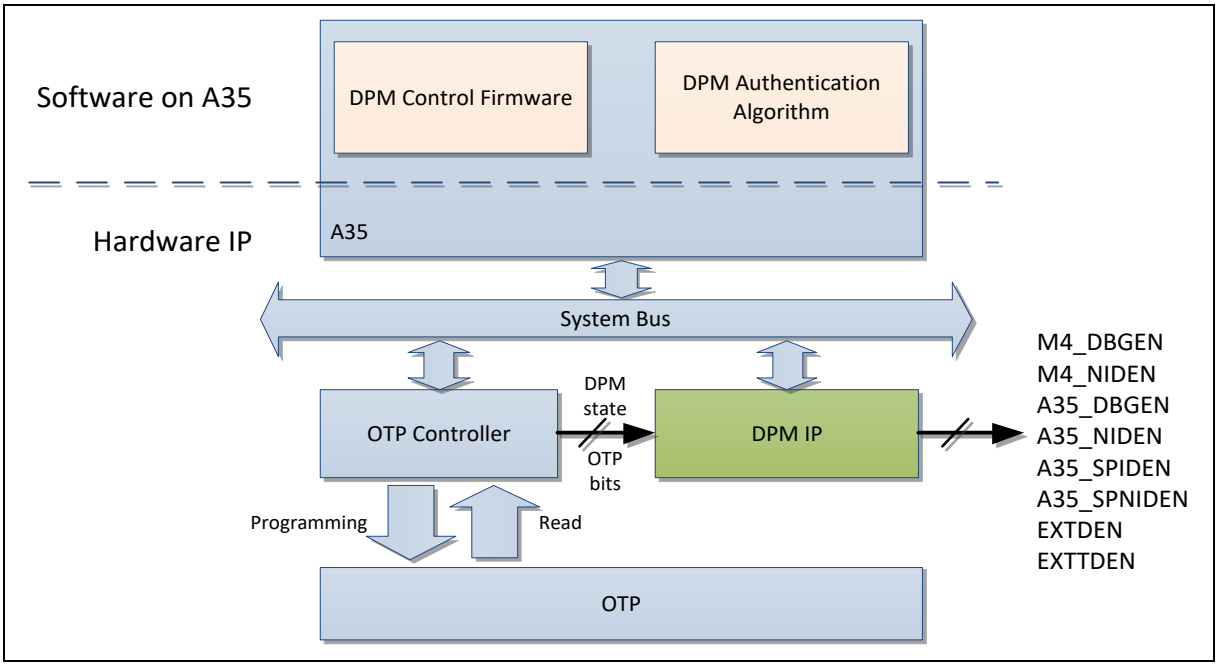


Figure 6.6-5 DPM Architecture

Debug State

Based on suggestions from Arm TBSA document, DPM defines debug state as: Reset, Locked, Default, Open and Closed state. The state is encoded by 3 bits consisting of 2 non-volatile OTP bits: OLOCK and ODIS, and a volatile software writable register, REGDEN. The encoding of these three bit is shown in Table 6.6-8. Description of each state is shown as below:

- Reset state: The state when the system is reset, and value to encode the debug state is not ready. At this state, debug function is disabled.
- Default state: The state when both OLOCK and ODIS are low, which is the default value when DPM has not been set. At this state, debug function is enabled.
- Closed state: The state when OLOCK is low while ODIS has been set to high and REGDEN bit is low, indicating the chip is under development, and the debug function should not be used without authentication first. At this state, the debug function is disabled and will be enabled after secure user sets the REGDEN bit to change the state to Open state.
- Open state: The state when OLOCK is low, ODIS is high and REGDEN bit has been set to high, indicating the debug user has been authenticated and should be able to use this debug function. At this state, the debug function is enabled.
- Locked state: The state when OLOCK has been set to high, indicating this debug function is disabled permanently and will not be used anymore.

When the chip runs after reset, the REGDEN bit in DPM is cleared to zero, and, if the OTP bits are not ready yet, the debug state is at Reset state, where the debug function is disabled. After the OTP bits are ready, the debug state enters one of Default, Locked or Closed state, depending on the OTP value. The debug function is enabled when the debug state is Default state while disabled when the debug state is at Locked or Closed state.

When the debug state is at Closed state, secure software can set the REGDEN bit in DPM registers to make the debug state enter Open state to enable the debug function. When the debug state is at Open state, secure user can also change it back to Closed state by set REGDEN to zero.

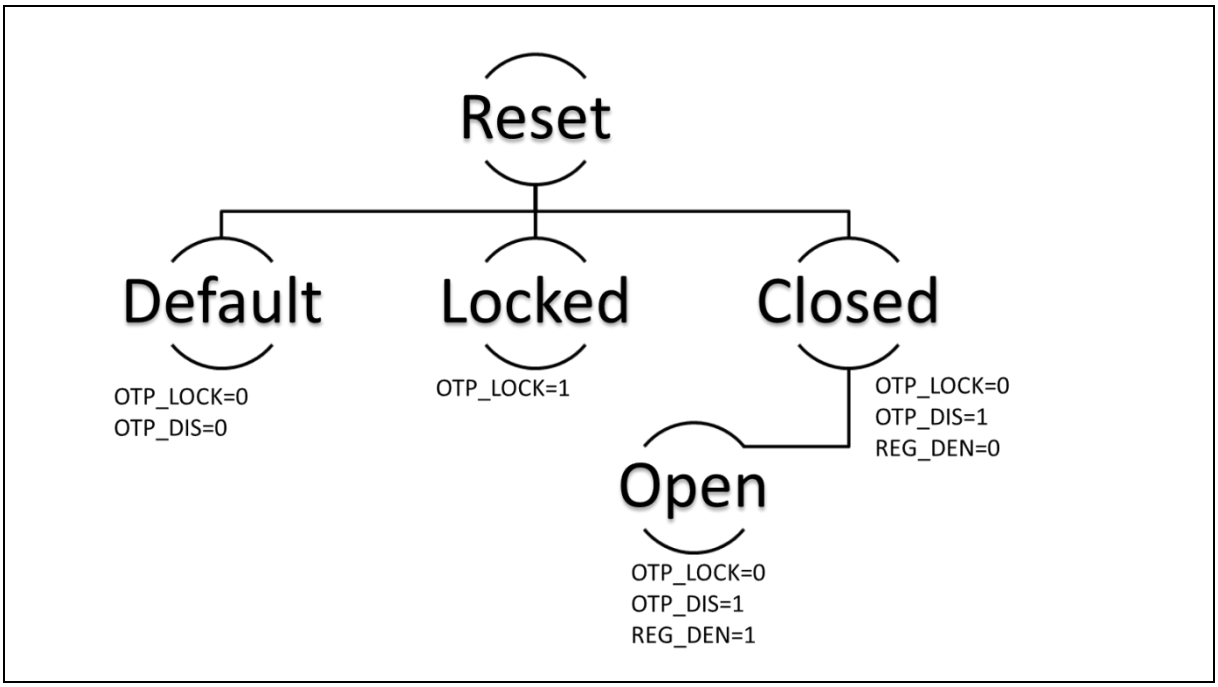


Figure 6.6-6 Flow of Debug State

OLOCK	ODIS	REGDEN	Debug State	Debug Function
-	-	-	Reset state	Disable
1	X	x	Locked state	Disable
0	0	x	Default state	Enable
0	1	0	Closed state	Disable
0	1	1	Open state	Enable

Table 6.6-8 Encoding of Debug State

Debug Control Signals

Debug signals of this chip include debug signals to Cortex-A35: DBGEN_A35, NIDEN_A35, SPIDEN_A35, SPNIDEN_A35, debug signals to Cortex-M4: DBGEN_M4, NIDEN_M4, and debug signals to debug components: EXTDEN, EXTTDEN.

All debug signals and corresponding register are listed in Table 6.6-9.

To achieve a kind of debug function scheme, it may require single signal or multiple signals. For example: To do invasive debug to Cortex-M4 for development, both DBGEN_M4, EXTDEN signals should be taken to enable. Refer to the “Classic Debug Functions” sub-section for more details.

Register	*Debug Function*	*Related Components*	*Description*
DPM_M4DS	DBGEN_M4	CM4	Controls the ability of Cortex-M4 invasive debug
DPM_M4NDS	NIDEN_M4	CM4	Controls the ability of Cortex-M4 non-invasive debug
DPM_A35NSDS	DBGEN_A35	CA35, AXI-AP	Controls the ability of Cortex-A35 non-secure invasive debug

DPM_A35NSNDS	NIDEN_A35	CA35	Controls the ability of Cortex-A35 non-secure non-invasive debug
DPM_A35SDS	SPIDEN_A35	CA35, AXI-AP	Controls the ability of Cortex-A35 secure invasive debug
DPM_A35SNDS	SPNIDEN_A35	CA35	Controls the ability of Cortex-A35 secure non-invasive debug
DPM_EXTDS	EXTDEN	AXI-AP, M4-AP, APB-AP	Controls all APs' accessibility to system
DPM_EXTTDS	EXTTDEN	TPIU	Controls the ability of tracing data output to external

Table 6.6-9 List of Debug Functions and Corresponding Register

Note: DBGEN_M4 cannot restrict external debugger accessing SubM domain system, but can only prevent M4 from halted. If user intends to restrict external debugger accessing SubM domain, EXTDEN should be used.

Classic Debug Functions

In this chip, there are a TrustZone-based cortex®-A core and a Cortex-M core. The debug scenario may be various. According to Arm architecture reference manual, Cortex-A core can have external debug, as well as self-hosted debug. Furthermore, the debug of TrustZone-based core can be divided to secure debug and non-secure debug. Cortex-M can only have external debug.

For self-hosted debug, the control is based on Armv8 architecture, which is not controlled by DPM registers. Refer to Armv8 architecture reference manual for more details.

Table 6.6-10 lists all kinds of debug functions and related control signals. If user needs a kind of debug function, all related control signals should be at "Enabled" state, i.e. Default state or Open state. For example: if user intend to do invasive debug of TrustZone secure by external debugger, registers of DPM_A35SDS, DPM_A35NSDS, DPM_EXTDS should all be at Open state or Default state. Otherwise, this debug function cannot be achieved.

Debug Scheme	**Debug Function**	**Invasive?**	**Registers Of Required Control Signal**
TrustZone Secure and TrustZone Non-secure	External debug	Invasive	DPM_A35SDS, DPM_A35NSDS, DPM_EXTDS
		Non-invasive	DPM_A35SNDS, DPM_A35NSNDS, DPM_EXTDS, DPM_EXTTDS
	Self-hosted debug	Invasive	Controlled by Cortex-A35 Registers[1]
		Non-invasive	Controlled by Cortex-A35 Registers
TrustZone Non-secure only	External debug	Invasive	DPM_A35NSDS, DPM_EXTDS
		Non-invasive	DPM_A35NSNDS, DPM_EXTDS, DPM_EXTTDS

[1] Refer to Arm "ARM Architecture Reference Manual, Armv8"

	Self-hosted debug	Invasive	Controlled by Cortex-A35 Registers
		Non-invasive	Controlled by Cortex-A35 Registers
Cortex-M4 Region only	External debug	Invasive	DPM_M4DS, DPM_EXTDS
		Non-invasive	DPM_M4NDS, DPM_EXTDS, DPM_EXTTDS

Table 6.6-10 List of Classic Debug Functions

Example of Authentication Algorithm

Some simple methods to do debug authentication are described below.

Password-based debug authentication

Password is a simple way to do authentication. The user with specific password can be authenticated to gain the specific debug function.

In this scheme, the password or the hash of the password is stored in a secure storage in advance. User can insert the password to do authentication. The DPM Authentication Algorithm will compare the inserted password and the password hash stored in advance. If the result is match, the DPM Firmware will access the OTP controller or DPM IP to change corresponding debug state.

An example of debug authentication scheme is shown in Figure 6.6-7. The example describes the condition that the debug state is initially at Close state, that is ODIS is 1 and REGDEN is 0. The secure software provides an interface for user to do debug authentication. User can insert password for compare through this interface. Then, the secure software will load the existed hash of the password which is stored in a secure area to do the comparison. If the result of the comparison is correct, then the secure software will write 1 to the REGDEN to make the debug state change from Close state to Open state.

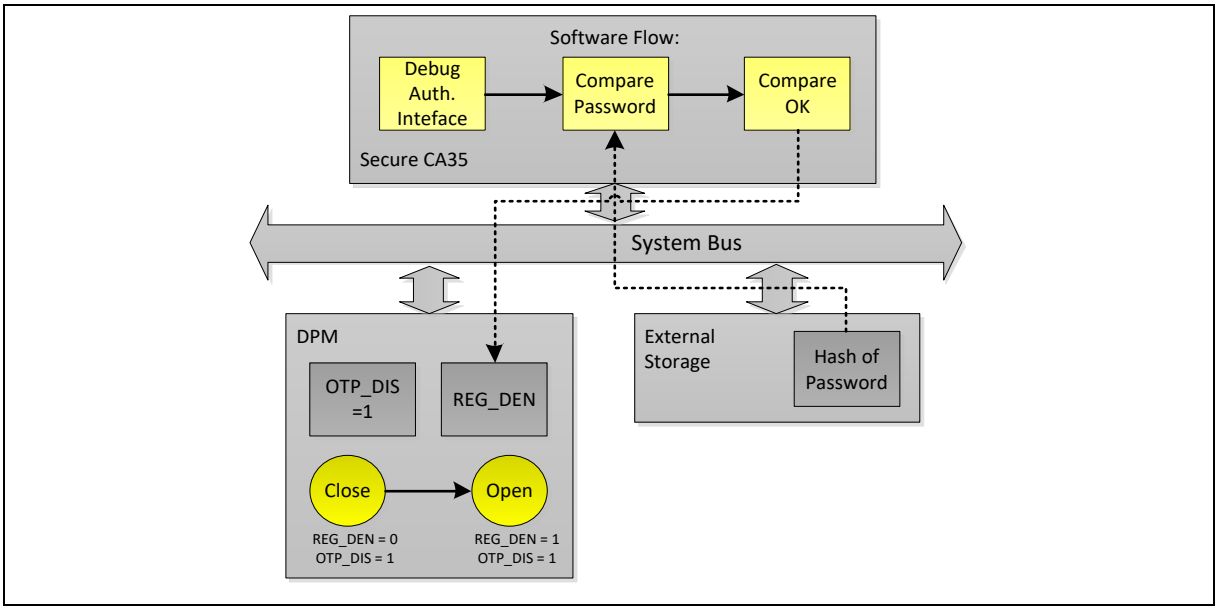


Figure 6.6-7 Password-based Debug Authentication Scheme

Debug State OTP bits

DPM OTP bits is stored in OTP, and should be updated through OTP controller. The arrangement of OTP bits is listed in Table 6.6-11. To update OTP value, refer to the “OTP Controller” section for more details.

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
EXTT_LOCK	EXTT_DIS	EXT_LOCK	EXT_DIS	M4NDS_ LOCK	M4NDS_DIS	M4DS_LOCK	M4DS_DIS
7	6	5	4	3	2	1	0
A35NSNDS_ LOCK	A35NSNDS_ DIS	A35NSDS_ LOCK	A35NSDS_ DIS	A35SNDS_ LOCK	A35SNDS_ DIS	A35SDS_ LOCK	A35SDS_DIS

Bits	Description	
[15]	EXTT_LOCK	DPM External Tracing Debug State LOCK OTP Bit
[14]	EXTT_DIS	DPM External Tracing Debug State DISABLE OTP Bit
[13]	EXT_LOCK	DPM External Debug State LOCK OTP Bit
[12]	EXT_DIS	DPM External Debug State DISABLE OTP Bit
[11]	M4NDS_LOCK	DPM M4 Non-invasive Debug State LOCK OTP Bit
[10]	M4NDS_DIS	DPM M4 Non-invasive Debug State DISABLE OTP Bit
[9]	M4DS_LOCK	DPM M4 Debug State LOCK OTP Bit
[8]	M4DS_DIS	DPM M4 Debug State DISABLE OTP Bit
[7]	A35NSNDS _LOCK	DPM A35 Non-secure Non-invasive Debug State LOCK OTP Bit
[6]	A35NSNDS_DIS	DPM A35 Non-secure Non-invasive Debug State DISABLE OTP Bit
[5]	A35NSDS_LOCK	DPM A35 Non-secure Debug State LOCK OTP Bit
[4]	A35NSDS_DIS	DPM A35 Non-secure Debug State DISABLE OTP Bit
[3]	A35SNDS_LOCK	DPM A35 Secure Non-invasive Debug State LOCK OTP Bit
[2]	A35SNDS_DIS	DPM A35 Secure Non-invasive Debug State DISABLE OTP Bit
[1]	A35SDS_LOCK	DPM A35 Secure Debug State LOCK OTP Bit
[0]	A35SDS_DIS	DPM A35 Secure Debug State DISABLE OTP Bit

Table 6.6-11 DPM Bit Description in OTP

6.6.4.6 Product Life-cycle Management

Product life-cycle management, PLM, uses pre-defined life-cycle stages to provide hardware-based controls controlling the ability of some functions, such as Boot source selection, external debug access ability and so forth. The life-cycle stage of this chip defines 3 main stages:, OEM, Deployed, RMA stage as shown in Table 6.6-12.

Concept and Functions of Lifecycle Management

The lifecycle flow of a chip should be one-way. When the chip is shipped for development, it is at OEM stage, where external debug ability is opened to user (developer) for convenience.

After development of the chip, the chip will be shipped to be used by end user which means the chip is progressed to Deployed stage, where valuable assets have already been injected to the chip and, thus, the external accessibility to the chip should be restricted for the sake of asset protection.

When the chip, used by end users, is required to be thoroughly debugged or tested due to some reasons, the chip should be progressed to RMA stage, where the users' assets should be protected from leakage. It is achieved by the Key Store, which will revoke user's key when the chip enters to RMA stage.

This chip uses the on-chip One-Time-Program memory to record the lifecycle stage of the chip to make sure the flow of lifecycle to be unidirectional. The lifecycle record will be loaded to shadow registers after reset de-asserted. Secure user can read the current lifecycle stage value through PLM_STAGE register.

Update Lifecycle Stage

User with enough authority can update lifecycle stage through OTP controller. Updating life-cycle stage can only be triggered by secure user.. Table 6.6-13 shows lifecycle bit field stored in OTP. After user updates the lifecycle stage setting in OTP, the setting will not take place without a cold reset (pinReset, POR, CHIP_RST) to trigger the shadow register reload process.

Lifecycle Stage	PLM_STS.STAGE	Description Of Functions
OEM	4'b0001	CA35 External Debug – Controlled by DPM CM4 Debug – Controlled by DPM WDT1 Enable – Controlled by OTP bit Boot Source – Controlled by OTP bit or by default Secure boot – Controlled by OTP bit or by default
Deployed	4'b0011	CA35 External Debug – Disabled CM4 Debug – Disabled WDT1 Enable – Controlled by OTP bit Boot Source – Controlled by OTP bit or by default Secure boot – Controlled by OTP bit or by default
RMA	4'b0111, 4'b0101	CA35 External Debug – Controlled by DPM CM4 Debug – Controlled by DPM WDT1 Disabled Key Store – All OTP keys revoked. Boot Source – Default setting, regardless of OTP setting Secure boot – Disabled
Note: WDT1 = Watchdog timer 1		

Table 6.6-12 List of Functions of Each PLM Stage

PLM Setting (OTP Logic Address = 0x108)

31	30	29	28	27	26	25	24

Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					RMA	Deployed	OEM

Bits	Description	
[31:3]	Reserved	Reserved
[2]	RMA	RMA Bit
[1]	Deployed	Deployed Bit
[0]	OEM	OEM Bit

Table 6.6-13 Lifecycle OTP Bits

6.6.5 Register Map

R: read only, **W**: write only, **R/W**: both read and write, **C**: Only value 0 can be written

Register	Offset	R/W	Description	Reset Value
SSPCC Base Address: **SSPCC_BA = 0x404F_0000** **DPM_BA = 0x404F_0600** **PLM_BA = 0x404F_0700**				
SSPCC_PSSET0	SSPCC_BA+0x00	R/W	Peripheral Security Attribution Set Register 0 (0x4000_0000~0x400F_FFFF)	0x0050_0000
SSPCC_PSSET1	SSPCC_BA+0x04	R/W	Peripheral Security Attribution Set Register 1 (0x4010_0000~0x401F_FFFF)	0x0000_0001
SSPCC_PSSET2	SSPCC_BA+0x08	R/W	Peripheral Security Attribution Set Register 2 (0x4020_0000~0x402F_FFFF)	0x1100_0000
SSPCC_PSSET3	SSPCC_BA+0x0C	R/W	Peripheral Security Attribution Set Register 3 (0x4030_0000~0x403F_FFFF)	0x1100_0000
SSPCC_PSSET4	SSPCC_BA+0x10	R/W	Peripheral Security Attribution Set Register 4 (0x4040_0000~0x404F_FFFF)	0x0015_0050
SSPCC_PSSET5	SSPCC_BA+0x14	R/W	Peripheral Security Attribution Set Register 5 (0x4050_0000~0x405F_FFFF)	0x0015_0554
SSPCC_PSSET6	SSPCC_BA+0x18	R/W	Peripheral Security Attribution Set Register 6 (0x4060_0000~0x406F_FFFF)	0x0004_0055
SSPCC_PSSET7	SSPCC_BA+0x1C	R/W	Peripheral Security Attribution Set Register 7 (0x4070_0000~0x407F_FFFF)	0x5555_5554
SSPCC_PSSET8	SSPCC_BA+0x20	R/W	Peripheral Security Attribution Set Register 8 (0x4080_0000~0x408F_FFFF)	0x0001_0554
SSPCC_PSSET9	SSPCC_BA+0x24	R/W	Peripheral Security Attribution Set Register 9 (0x4090_0000~0x409F_FFFF)	0x0000_0005
SSPCC_PSSET10	SSPCC_BA+0x28	R/W	Peripheral Security Attribution Set Register 10 (0x40A0_0000~0x40AF_FFFF)	0x0000_0000
SSPCC_PSSET11	SSPCC_BA+0x2C	R/W	Peripheral Security Attribution Set Register 11 (0x40B0_0000~0x40BF_FFFF)	0x0000_1515
SSPCC_SRAMSB	SSPCC_BA+0x40	R/W	SRAM Security Boundary Set Register	0x0000_1000
SSPCC_EBISSET	SSPCC_BA+0x50	R/W	EBI Memory Security Set Register	0x0000_0015
SSPCC_IOASSET	SSPCC_BA+0x60	R/W	GPIO Port A Security Attribution Set Register	0x0000_0000
SSPCC_IOBSSET	SSPCC_BA+0x64	R/W	GPIO Port B Security Attribution Set Register	0x0000_0000
SSPCC_IOCSSET	SSPCC_BA+0x68	R/W	GPIO Port C Security Attribution Set Register	0x0000_0000
SSPCC_IODSSET	SSPCC_BA+0x6C	R/W	GPIO Port D Security Attribution Set Register	0x0000_0000
SSPCC_IOESSET	SSPCC_BA+0x70	R/W	GPIO Port E Security Attribution Set Register	0x0000_0000
SSPCC_IOFSSET	SSPCC_BA+0x74	R/W	GPIO Port F Security Attribution Set Register	0x0000_0000
SSPCC_IOGSSET	SSPCC_BA+0x78	R/W	GPIO Port G Security Attribution Set Register	0x0000_0000
SSPCC_IOHSSET	SSPCC_BA+0x7C	R/W	GPIO Port H Security Attribution Set Register	0x0000_0000

SSPCC_IOISSET	SSPCC_BA+0x80	R/W	GPIO Port I Security Attribution Set Register	0x0000_0000
SSPCC_IOJSSET	SSPCC_BA+0x84	R/W	GPIO Port J Security Attribution Set Register	0x0000_0000
SSPCC_IOKSSET	SSPCC_BA+0x88	R/W	GPIO Port K Security Attribution Set Register	0x0000_0000
SSPCC_IOLSSET	SSPCC_BA+0x8C	R/W	GPIO Port L Security Attribution Set Register	0x0000_0000
SSPCC_IOMSSET	SSPCC_BA+0x90	R/W	GPIO Port M Security Attribution Set Register	0x0000_0000
SSPCC_IONSSET	SSPCC_BA+0x94	R/W	GPIO Port N Security Attribution Set Register	0x0000_0000
SSPCC_SVIEN	SSPCC_BA+0xF0	R/W	Security Violation Interrupt Enable Register	0x0000_0000
SSPCC_SVINTSTS	SSPCC_BA+0xF4	R/W	Security Violation Interrupt Status Register	0x0000_0000
SSPCC_APB0VSRC	SSPCC_BA+0x100	R	APB0 Security Policy Violation Source	0x0000_0000
SSPCC_APB3VSRC	SSPCC_BA+0x104	R	APB3 Security Policy Violation Source	0x0000_0000
SSPCC_SRAM1VSRC	SSPCC_BA+0x108	R	SRAM1 Security Policy Violation Source	0x0000_0000
SSPCC_SDH0VSRC	SSPCC_BA+0x10C	R	SDH0 Security Policy Violation Source	0x0000_0000
SSPCC_SDH1VSRC	SSPCC_BA+0x110	R	SDH1 Security Policy Violation Source	0x0000_0000
SSPCC_HSUSBDVSRC	SSPCC_BA+0x114	R	HSUSBD Security Policy Violation Source	0x0000_0000
SSPCC_NANDCVSRC	SSPCC_BA+0x118	R	NANDC Security Policy Violation Source	0x0000_0000
SSPCC_PDMA0VSRC	SSPCC_BA+0x11C	R	PDMA0 Security Policy Violation Source	0x0000_0000
SSPCC_PDMA1VSRC	SSPCC_BA+0x120	R	PDMA1 Security Policy Violation Source	0x0000_0000
SSPCC_TRNGVSRC	SSPCC_BA+0x124	R	TRNG Security Policy Violation Source	0x0000_0000
SSPCC_CRYPTOVSRC	SSPCC_BA+0x128	R	CRYPTO Security Policy Violation Source	0x0000_0000
SSPCC_APB0VA	SSPCC_BA+0x180	R	APB0 Security Policy Violation Address	0x0000_0000
SSPCC_APB3VA	SSPCC_BA+0x184	R	APB3 Security Policy Violation Address	0x0000_0000
SSPCC_SRAM1VA	SSPCC_BA+0x188	R	SRAM1 Security Policy Violation Address	0x0000_0000
SSPCC_SDH0VA	SSPCC_BA+0x18C	R	SDH0 Security Policy Violation Address	0x0000_0000
SSPCC_SDH1VA	SSPCC_BA+0x190	R	SDH1 Security Policy Violation Address	0x0000_0000
SSPCC_HSUSBDVA	SSPCC_BA+0x194	R	HSUSBD Security Policy Violation Address	0x0000_0000
SSPCC_NANDCVA	SSPCC_BA+0x198	R	NANDC Security Policy Violation Address	0x0000_0000
SSPCC_PDMA0VA	SSPCC_BA+0x19C	R	PDMA0 Security Policy Violation Address	0x0000_0000
SSPCC_PDMA1VA	SSPCC_BA+0x1A0	R	PDMA1 Security Policy Violation Address	0x0000_0000
SSPCC_TRNGVA	SSPCC_BA+0x1A4	R	TRNG Security Policy Violation Address	0x0000_0000
SSPCC_CRYPTOVA	SSPCC_BA+0x1A8	R	CRYPTO Security Policy Violation Address	0x0000_0000
SSPCC_SINFAEN	SSPCC_BA+0x200	R/W	Shared Information Access Enable Register	0x0000_00CF

SSPCC_SCWP	SSPCC_BA+0x204	R/W	Security Configuration Write Protection Register	0x0000_0000
DPM_A35SDS	DPM_BA+0x10	R/W	DPM A35 Secure Debug State Register	0xA500_000X
DPM_A35SNDS	DPM_BA+0x14	R/W	DPM A35 Secure Non-invasive Debug State Register	0xA500_000X
DPM_A35NSDS	DPM_BA+0x18	R/W	DPM A35 Non-secure Debug State Register	0xA500_000X
DPM_A35NSNDS	DPM_BA+0x1C	R/W	DPM A35 Non-secure Non-invasive Debug State Register	0xA500_000X
DPM_M4DS	DPM_BA+0x20	R/W	DPM M4 Debug State Register	0xA500_000X
DPM_M4NDS	DPM_BA+0x24	R/W	DPM M4 Non-invasive Debug State Register	0xA500_000X
DPM_EXTDS	DPM_BA+0x30	R/W	DPM External Debug State Register	0xA500_000X
DPM_EXTTDS	DPM_BA+0x34	R/W	DPM External Tracing Debug State Register	0xA500_000X
PLM_STS	PLM_BA+0x04	R	Product Life-cycle Status Register (TZNS)	0xA500_000X

6.6.6 Register Description

Peripheral Security Attribution Set Register 0

Register	Offset	R/W	Description	Reset Value
SSPCC_PSSET0	SSPCC_BA+0x00	R/W	Peripheral Security Attribution Set Register 0 (0x4000_0000~0x400F_FFFF)	0x0050_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
PDMA3		PDMA2		PDMA1		PDMA0	
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:22]	**PDMA3**	**PDMA3 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[21:20]	**PDMA2**	**PDMA2 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[19:18]	**PDMA1**	**PDMA1 Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Reserved.
[17:16]	**PDMA0**	**PDMA0 Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Reserved.
[15:0]	**Reserved**	Reserved.

Peripheral Security Attribution Set Register 1

Register	Offset	R/W	Description	Reset Value
SSPCC_PSSET1	SSPCC_BA+0x04	R/W	Peripheral Security Attribution Set Register 1 (0x4010_0000~0x401F_FFFF)	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		NANDC		SDH1		SDH0	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						EBI	

Bits	Description	
[31:25]	**Reserved**	Reserved.
[21:20]	**NANDC**	**NAND Controller Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Reserved.
[19:18]	**SDH1**	**SD Host 1 Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Reserved.
[17:16]	**SDH0**	**SD Host 0 Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Reserved.
[15:2]	**Reserved**	Reserved.
[1:0]	**EBI**	**EBI Controller Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.

Peripheral Security Attribution Set Register 2

Register	Offset	R/W	Description	Reset Value
SSPCC_PSSET2	SSPCC_BA+0x08	R/W	Peripheral Security Attribution Set Register 2 (0x4020_0000~0x402F_FFFF)	0x1100_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						HSUSBD	

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1:0]	**HSUSBD**	**HS USB Device Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Reserved.

Peripheral Security Attribution Set Register 3

Register	Offset	R/W	Description	Reset Value
SSPCC_PSSET3	SSPCC_BA+0x0C	R/W	Peripheral Security Attribution Set Register 3 (0x4030_0000~0x403F_FFFF)	0x1100_0000

31	30	29	28	27	26	25	24
CANFD3		CANFD2		CANFD1		CANFD0	
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							

Bits	Description	
[31:30]	**CANFD3**	**CAN FD 3 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[29:28]	**CANFD2**	**CAN FD 2 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[27:26]	**CANFD1**	**CAN FD 1 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[25:24]	**CANFD0**	**CAN FD 0 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[23:0]	**Reserved**	Reserved.

Peripheral Security Attribution Set Register 4

Register	Offset	R/W	Description	Reset Value
SSPCC_PSSET4	SSPCC_BA+0x10	R/W	Peripheral Security Attribution Set Register 4 (0x4040_0000~0x404F_FFFF)	0x0015_0050

31	30	29	28	27	26	25	24
Reserved				MCTL		DDRPHYPUB	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		KPI		I2S1		I2S0	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						WDTWWDT1	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EADC0		ADC0		Reserved			

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:26]	**MCTL**	**DRAM Controller Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Reserved.
[25:24]	**DDRPHYPUB**	**DDR PHY Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Reserved.
[23:22]	**Reserved**	Reserved.
[21:20]	**KPI**	**KPI Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[19:18]	**I2S1**	**I^2S 1 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.

Bits	Description	
[17:16]	**I2S0**	**I²S 0 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[15:10]	**Reserved**	Reserved.
[9:8]	**WDTWWDT1**	**WDTWWDT1 Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Reserved.
[7:6]	**EADC0**	**EADC 0 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[5:4]	**ADC0**	**ADC 0 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[3:0]	**Reserved**	Reserved.

Peripheral Security Attribution Set Register 5

Register	Offset	R/W	Description	Reset Value
SSPCC_PSSET5	SSPCC_BA+0x14	R/W	Peripheral Security Attribution Set Register 5 (0x4050_0000~0x405F_FFFF)	0x0015_0554

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved		EPWM2		EPWM1		EPWM0	
15	14	13	12	11	10	9	8
Reserved				TMR1011		TMR89	
7	6	5	4	3	2	1	0
TMR67		TMR45		TMR23		TMR01	

Bits	Description	
[31:22]	**Reserved**	Reserved.
[21:20]	**EPWM2**	**EPWM 2 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[19:18]	**EPWM1**	**EPWM 1 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[17:16]	**EPWM0**	**EPWM 0 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[15:12]	**Reserved**	Reserved.
[11:10]	**TMR1011**	**TMR1011 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.

Bits	Description	
[9:8]	**TMR89**	**TMR89 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[7:6]	**TMR67**	**TMR67 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[5:4]	**TMR45**	**TMR45 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[3:2]	**TMR23**	**TMR23 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[1:0]	**TMR01**	**TMR01 Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Reserved.

Peripheral Security Attribution Set Register 6

Register	Offset	R/W	Description	Reset Value
SSPCC_PSSET6	SSPCC_BA+0x18	R/W	Peripheral Security Attribution Set Register 6 (0x4060_0000~0x406F_FFFF)	0x0004_0055

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				QSPI1		QSPI0	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SPI3		SPI2		SPI1		SPI0	

Bits	Description	
[31:20]	**Reserved**	Reserved.
[19:18]	**QSPI1**	**QSPI1 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[17:16]	**QSPI0**	**QSPI0 Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Reserved.
[15:8]	**Reserved**	Reserved.
[7:6]	**SPI3**	**SPI3 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[5:4]	**SPI2**	**SPI2 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.

Bits	Description	
[3:2]	**SPI1**	**SPI1 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[1:0]	**SPI0**	**SPI0 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.

Peripheral Security Attribution Set Register 7

Register	Offset	R/W	Description	Reset Value
SSPCC_PSSET7	SSPCC_BA+0x1C	R/W	Peripheral Security Attribution Set Register 7 (0x4070_0000~0x407F_FFFF)	0x5555_5554

31	30	29	28	27	26	25	24
UART15		UART14		UART13		UART12	
23	22	21	20	19	18	17	16
UART11		UART10		UART9		UART8	
15	14	13	12	11	10	9	8
UART7		UART6		UART5		UART4	
7	6	5	4	3	2	1	0
UART3		UART2		UART1		UART0	

Bits	Description	
[31:30]	**UART15**	**UART15 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[29:28]	**UART14**	**UART14 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[27:26]	**UART13**	**UART13 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[25:24]	**UART12**	**UART12 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[23:22]	**UART11**	**UART11 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.

Bits	Description	
[21:20]	**UART10**	**UART10 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[19:18]	**UART9**	**UART9 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[17:16]	**UART8**	**UART8 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[15:14]	**UART7**	**UART7 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[13:12]	**UART6**	**UART6 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[11:10]	**UART5**	**UART5 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[9:8]	**UART4**	**UART4 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[7:6]	**UART3**	**UART3 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.

Bits	Description	
[5:4]	**UART2**	**UART2 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[3:2]	**UART1**	**UART1 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[1:0]	**UART0**	**UART0 Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Reserved.

Peripheral Security Attribution Set Register 8

Register	Offset	R/W	Description	Reset Value
SSPCC_PSSET8	SSPCC_BA+0x20	R/W	Peripheral Security Attribution Set Register 8 (0x4080_0000~0x408F_FFFF)	0x0001_0554

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved						UART16	
15	14	13	12	11	10	9	8
Reserved				I2C5		I2C4	
7	6	5	4	3	2	1	0
I2C3		I2C2		I2C1		I2C0	

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17:16]	**UART16**	**UART16 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[15:12]	**Reserved**	Reserved.
[11:10]	**I2C5**	**I2C5 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[9:8]	**I2C4**	**I2C4 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[7:6]	**I2C3**	**I2C3 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.

Bits	Description	
[5:4]	**I2C2**	**I2C2 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[3:2]	**I2C1**	**I2C1 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[1:0]	**I2C0**	**I2C0 Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Reserved.

Peripheral Security Attribution Set Register 9

Register	Offset	R/W	Description	Reset Value
SSPCC_PSSET9	SSPCC_BA+0x24	R/W	Peripheral Security Attribution Set Register 9 (0x4090_0000~0x409F_FFFF)	0x0000_0005

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				SC1		SC0	

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3:2]	**SC1**	**SC1 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[1:0]	**SC0**	**SC0 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.

Peripheral Security Attribution Set Register 10

Register	Offset	R/W	Description	Reset Value
SSPCC_PSSET10	SSPCC_BA+0x28	R/W	Peripheral Security Attribution Set Register 10 (0x40A0_0000~0x40AF_FFFF)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:0]	**Reserved**	Reserved.

Peripheral Security Attribution Set Register 11

Register	Offset	R/W	Description	Reset Value
SSPCC_PSSET11	SSPCC_BA+0x2C	R/W	Peripheral Security Attribution Set Register 11 (0x40B0_0000~0x40BF_FFFF)	0x0000_1515

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				TRNG		Reserved	
15	14	13	12	11	10	9	8
Reserved		ECAP2		ECAP1		ECAP0	
7	6	5	4	3	2	1	0
Reserved		QEI2		QEI1		QEI0	

Bits	Description	
[31:20]	**Reserved**	Reserved.
[19:18]	**TRNG**	**TRNG Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Reserved.
[17:14]	**Reserved**	Reserved.
[13:12]	**ECAP2**	**ECAP2 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[11:10]	**ECAP1**	**ECAP1 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[9:8]	**ECAP0**	**ECAP0 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[7:6]	**Reserved**	Reserved.

Bits	Description	
[5:4]	**QEI2**	**QEI2 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[3:2]	**QEI1**	**QEI1 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[1:0]	**QEI0**	**QEI0 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.

SRAM Security Boundary Set Register (SSPCC_SRAMSB)

Register	Offset	R/W	Description	Reset Value
SSPCC_SRAMSB	SSPCC_BA+0x40	R/W	SRAM Security Boundary Set Register	0x0000_1000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			SR1BOUND				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			SR0BOUND				

Bits	Description	
[31:13]	**Reserved**	Reserved.
[12:8]	**SR1BOUND**	**Set SRAM1 Security Boundary** Set the 16 Kbytes-aligned boundary of data SRAM. The SRAM part below the boundary is TZS region and the part higher the boundary is TZNS region. When the boundary is set to higher than the size of SRAM, the whole SRAM is TZS region.
[7:5]	**Reserved**	Reserved.
[4:0]	**SR0BOUND**	**Set SRAM0 Security Boundary** Set the 16 Kbytes-aligned boundary of SRAM0. The SRAM part below the boundary is SubM region and the part higher the boundary is ShareMemory region. When the boundary is set to higher than the size of SRAM, the whole SRAM is SubM region.

EBI Memory Security Set Register (SSPCC_EBISSET)

Register	Offset	R/W	Description	Reset Value
SSPCC_EBISSET	SSPCC_BA+0x50	R/W	EBI Memory Security Set Register	0x0000_0015

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		BANK2		BANK1		BANK0	

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5:4]	**BANK2**	**EBI Memory Bank2 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[3:2]	**BANK1**	**EBI Memory Bank1 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[1:0]	**BANK0**	**EBI Memory Bank0 Security Attribution Bits** 00 = Reserved. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.

GPIO Port Security Attribution Set Register (SSPCC_IOnSSET)

Register	Offset	R/W	Description	Reset Value
SSPCC_IOASSET	SSPCC_BA+0x60	R/W	GPIO Port A Security Attribution Set Register	0x0000_0000
SSPCC_IOBSSET	SSPCC_BA+0x64	R/W	GPIO Port B Security Attribution Set Register	0x0000_0000
SSPCC_IOCSSET	SSPCC_BA+0x68	R/W	GPIO Port C Security Attribution Set Register	0x0000_0000
SSPCC_IODSSET	SSPCC_BA+0x6C	R/W	GPIO Port D Security Attribution Set Register	0x0000_0000
SSPCC_IOESSET	SSPCC_BA+0x70	R/W	GPIO Port E Security Attribution Set Register	0x0000_0000
SSPCC_IOFSSET	SSPCC_BA+0x74	R/W	GPIO Port F Security Attribution Set Register	0x0000_0000
SSPCC_IOGSSET	SSPCC_BA+0x78	R/W	GPIO Port G Security Attribution Set Register	0x0000_0000
SSPCC_IOHSSET	SSPCC_BA+0x7C	R/W	GPIO Port H Security Attribution Set Register	0x0000_0000
SSPCC_IOISSET	SSPCC_BA+0x80	R/W	GPIO Port I Security Attribution Set Register	0x0000_0000
SSPCC_IOJSSET	SSPCC_BA+0x84	R/W	GPIO Port J Security Attribution Set Register	0x0000_0000
SSPCC_IOKSSET	SSPCC_BA+0x88	R/W	GPIO Port K Security Attribution Set Register	0x0000_0000
SSPCC_IOLSSET	SSPCC_BA+0x8C	R/W	GPIO Port L Security Attribution Set Register	0x0000_0000
SSPCC_IOMSSET	SSPCC_BA+0x90	R/W	GPIO Port M Security Attribution Set Register	0x0000_0000
SSPCC_IONSSET	SSPCC_BA+0x94	R/W	GPIO Port N Security Attribution Set Register	0x0000_0000

31	30	29	28	27	26	25	24
PIN15		PIN14		PIN13		PIN12	
23	22	21	20	19	18	17	16
PIN11		PIN10		PIN9		PIN8	
15	14	13	12	11	10	9	8
PIN7		PIN6		PIN5		PIN4	
7	6	5	4	3	2	1	0
PIN3		PIN2		PIN1		PIN0	

Bits	Description	
[31:30]	**PIN15**	**GPIO Pin 15 Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.

Bits	Description	
[29:28]	**PIN14**	**GPIO Pin 14 Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[27:26]	**PIN13**	**GPIO Pin 13 Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[25:24]	**PIN12**	**GPIO Pin 12 Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[23:22]	**PIN11**	**GPIO Pin 11 Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[21:20]	**PIN10**	**GPIO Pin 10 Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[19:18]	**PIN9**	**GPIO Pin 9 Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[17:16]	**PIN8**	**GPIO Pin 8 Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[15:14]	**PIN7**	**GPIO Pin 7 Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.

Bits	Description	
[13:12]	**PIN6**	**GPIO Pin 6 Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[11:10]	**PIN5**	**GPIO Pin 5 Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[9:8]	**PIN4**	**GPIO Pin 4 Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[7:6]	**PIN3**	**GPIO Pin 3 Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[5:4]	**PIN2**	**GPIO Pin 2 Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[3:2]	**PIN1**	**GPIO Pin 1 Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.
[1:0]	**PIN0**	**GPIO Pin 0 Security Attribution Bits** 00 = Set to TZS. 01 = Set to TZNS. 10 = Reserved. 11 = Set to SubM.

Security Violation Interrupt Enable Register (SSPCC_SVIEN)

Register	Offset	R/W	Description	Reset Value
SSPCC_SVIEN	SSPCC_BA+0xF0	R/W	Security Violation Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved					CRYPTO	TRNG	PDMA1
7	6	5	4	3	2	1	0
PDMA0	NANDC	HSUSBD	SDH1	SDH0	SRAM1	APB3	APB0

Bits	Description	
[31:11]	**Reserved**	Reserved.
[10]	**CRYPTO**	**CRYPTO Security Violation Interrupt Enable Bit** 0 = Interrupt triggered from security violation of CRYPTO Disabled. 1 = Interrupt triggered from security violation of CRYPTO Enabled.
[9]	**TRNG**	**TRNG Security Violation Interrupt Enable Bit** 0 = Interrupt triggered from security violation of TRNG Disabled. 1 = Interrupt triggered from security violation of TRNG Enabled.
[8]	**PDMA1**	**PDMA1 Security Violation Interrupt Enable Bit** 0 = Interrupt triggered from security violation of PDMA1 Disabled. 1 = Interrupt triggered from security violation of PDMA1 Enabled.
[7]	**PDMA0**	**PDMA0 Security Violation Interrupt Enable Bit** 0 = Interrupt triggered from security violation of PDMA0 Disabled. 1 = Interrupt triggered from security violation of PDMA0 Enabled.
[6]	**NANDC**	**NAND Controller Security Violation Interrupt Enable Bit** 0 = Interrupt triggered from security violation of NAND Controller Disabled. 1 = Interrupt triggered from security violation of NAND Controller Enabled.
[5]	**HSUSBD**	**HighSpeed USB Device Security Violation Interrupt Enable Bit** 0 = Interrupt triggered from security violation of HighSpeed USB Device Disabled. 1 = Interrupt triggered from security violation of HighSpeed USB Device Enabled.
[4]	**SDH1**	**SD Host1 Security Violation Interrupt Enable Bit** 0 = Interrupt triggered from security violation of SD host 1 Disabled. 1 = Interrupt triggered from security violation of SD host 1 Enabled.
[3]	**SDH0**	**SD Host0 Security Violation Interrupt Enable Bit** 0 = Interrupt triggered from security violation of SD host 0 Disabled. 1 = Interrupt triggered from security violation of SD host 0 Enabled.

Bits	Description	
[2]	**SRAM1**	**SRAM1 Security Violation Interrupt Enable Bit** 0 = Interrupt triggered from security violation of SRAM1 Disabled. 1 = Interrupt triggered from security violation of SRAM1 Enabled.
[1]	**APB3**	**APB3 Security Violation Interrupt Enable Bit** 0 = Interrupt triggered from security violation of APB3 Disabled. 1 = Interrupt triggered from security violation of APB3 Enabled.
[0]	**APB0**	**APB0 Security Violation Interrupt Enable Bit** 0 = Interrupt triggered from security violation of APB0 Disabled. 1 = Interrupt triggered from security violation of APB0 Enabled.

Security Violation Interrupt Status Register (SSPCC_SVINTSTS)

Register	Offset	R/W	Description	Reset Value
SSPCC_SVINTSTS	SSPCC_BA+0xF4	R/W	Security Violation Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved					CRYPTO	TRNG	PDMA1
7	6	5	4	3	2	1	0
PDMA0	NANDC	HSUSBD	SDH1	SDH0	SRAM1	APB3	APB0

Bits	Description	
[31:11]	**Reserved**	Reserved.
[10]	**CRYPTO**	**CRYPTO Security Violation Interrupt Status** 0 = No CRYPTO violation interrupt event. 1 = There is CRYPTO violation interrupt event. **Note:** Write 1 to clear the interrupt flag.
[9]	**TRNG**	**TRNG Security Violation Interrupt Status** 0 = No TRNG violation interrupt event. 1 = There is TRNG violation interrupt event. **Note:** Write 1 to clear the interrupt flag.
[8]	**PDMA1**	**PDMA1 Security Violation Interrupt Status** 0 = No PDMA1 violation interrupt event. 1 = There is PDMA1 violation interrupt event. **Note:** Write 1 to clear the interrupt flag.
[7]	**PDMA0**	**PDMA0 Security Violation Interrupt Status** 0 = No PDMA0 violation interrupt event. 1 = There is PDMA0 violation interrupt event. **Note:** Write 1 to clear the interrupt flag.
[6]	**NANDC**	**NAND Controller Security Violation Interrupt Status** 0 = No NAND Controller violation interrupt event. 1 = There is NAND Controller violation interrupt event. **Note:** Write 1 to clear the interrupt flag.
[5]	**HSUSBD**	**High-speed USB Device Security Violation Interrupt Status** 0 = No High-speed USB Device violation interrupt event. 1 = There is High-speed USB Device violation interrupt event. **Note:** Write 1 to clear the interrupt flag.

Bits	Description	
[4]	**SDH1**	**SDH1 Security Violation Interrupt Status** 0 = No SDH1 violation interrupt event. 1 = There is SDH1 violation interrupt event. **Note:** Write 1 to clear the interrupt flag.
[3]	**SDH0**	**SDH0 Security Violation Interrupt Status** 0 = No SDH0 violation interrupt event. 1 = There is SDH0 violation interrupt event. **Note:** Write 1 to clear the interrupt flag.
[2]	**SRAM1**	**SRAM1 Security Violation Interrupt Status** 0 = No SRAM1 violation interrupt event. 1 = There is SRAM1 violation interrupt event. **Note:** Write 1 to clear the interrupt flag.
[1]	**APB3**	**APB3 Security Violation Interrupt Status** 0 = No APB3 violation interrupt event. 1 = There is APB3 violation interrupt event. **Note:** Write 1 to clear the interrupt flag.
[0]	**APB0**	**APB0 Security Violation Interrupt Status** 0 = No APB0 violation interrupt event. 1 = There is APB0 violation interrupt event. **Note:** Write 1 to clear the interrupt flag.

Peripheral Security Violation Master Register (SSPCC_PVSRC)

Register	Offset	R/W	Description	Reset Value
SSPCC_APB0VSRC	SSPCC_BA+0x100	R	APB0 Security Policy Violation Source	0x0000_0000
SSPCC_APB3VSRC	SSPCC_BA+0x104	R	APB3 Security Policy Violation Source	0x0000_0000
SSPCC_SRAM1VSRC	SSPCC_BA+0x108	R	SRAM1 Security Policy Violation Source	0x0000_0000
SSPCC_SDH0VSRC	SSPCC_BA+0x10C	R	SDH0 Security Policy Violation Source	0x0000_0000
SSPCC_SDH1VSRC	SSPCC_BA+0x110	R	SDH1 Security Policy Violation Source	0x0000_0000
SSPCC_HSUSBDVSRC	SSPCC_BA+0x114	R	HSUSBD Security Policy Violation Source	0x0000_0000
SSPCC_NANDCVSRC	SSPCC_BA+0x118	R	NANDC Security Policy Violation Source	0x0000_0000
SSPCC_PDMA0VSRC	SSPCC_BA+0x11C	R	PDMA0 Security Policy Violation Source	0x0000_0000
SSPCC_PDMA1VSRC	SSPCC_BA+0x120	R	PDMA1 Security Policy Violation Source	0x0000_0000
SSPCC_TRNGVSRC	SSPCC_BA+0x124	R	TRNG Security Policy Violation Source	0x0000_0000
SSPCC_CRYPTOVSRC	SSPCC_BA+0x128	R	CRYPTO Security Policy Violation Source	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				MASTER			

Bits	Description	
[31:4]	**Reserved**	Reserved.

Bits	Description	
[3:0]	**MASTER**	**Master Violating Security Policy** Indicate which master invokes the security violation. 0x0 = CA35. 0x1 = CoreSight AP. 0x3 = TSI. 0x2 = CM4. 0x4 = PDMA0. 0x5 = PDMA1. 0x6 = PDMA2. 0x7 = PDMA3. 0x8 = SDH0. 0x9 = SDH1. 0xB = USBH0/HSUSBH0. 0xC = USBH1/HSUSBH1. 0xD = HSUSBD. 0xE = NAND. Others is undefined.

Peripheral Security Violation Address Register (SSPCC_PVA)

Register	Offset	R/W	Description	Reset Value
SSPCC_APB0VA	SSPCC_BA+0x180	R	APB0 Security Policy Violation Address	0x0000_0000
SSPCC_APB3VA	SSPCC_BA+0x184	R	APB3 Security Policy Violation Address	0x0000_0000
SSPCC_SRAM1VA	SSPCC_BA+0x188	R	SRAM1 Security Policy Violation Address	0x0000_0000
SSPCC_SDH0VA	SSPCC_BA+0x18C	R	SDH0 Security Policy Violation Address	0x0000_0000
SSPCC_SDH1VA	SSPCC_BA+0x190	R	SDH1 Security Policy Violation Address	0x0000_0000
SSPCC_HSUSBDVA	SSPCC_BA+0x194	R	HSUSBD Security Policy Violation Address	0x0000_0000
SSPCC_NANDCVA	SSPCC_BA+0x198	R	NANDC Security Policy Violation Address	0x0000_0000
SSPCC_PDMA0VA	SSPCC_BA+0x19C	R	PDMA0 Security Policy Violation Address	0x0000_0000
SSPCC_PDMA1VA	SSPCC_BA+0x1A0	R	PDMA1 Security Policy Violation Address	0x0000_0000
SSPCC_TRNGVA	SSPCC_BA+0x1A4	R	TRNG Security Policy Violation Address	0x0000_0000
SSPCC_CRYPTOVA	SSPCC_BA+0x1A8	R	CRYPTO Security Policy Violation Address	0x0000_0000

31	30	29	28	27	26	25	24
VIOADDR							
23	22	21	20	19	18	17	16
VIOADDR							
15	14	13	12	11	10	9	8
VIOADDR							
7	6	5	4	3	2	1	0
VIOADDR							

Bits	Description	
[31:0]	**VIOADDR**	**Violation Address** Indicate the target address of the access, which invokes the security violation.

Shared Information Access Enable Register (SSPCC_SINFAEN)

Register	Offset	R/W	Description	Reset Value
SSPCC_SINFAEN	SSPCC_BA+0x200	R/W	Shared Information Access Enable Register	0x0000_00CF

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					RTCSIAEN	SYSSIAEN	SSPCCSIAEN

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**RTCSIAEN**	**RTC Shared Information Access Enable Bit** 0 = RTC shared information Disabled. 1 = RTC shared information Enabled.
[1]	**SYSSIAEN**	**SYS Shared Information Access Enable Bit** 0 = SYS shared information Disabled. 1 = SYS shared information Enabled. **Note:** Include clock information.
[0]	**SSPCCSIAEN**	**SSPCC Shared Information Access Enable Bit** 0 = SSPCC shared information Disabled. 1 = SSPCC shared information Enabled.

Security Configuration Write Protection Register (SSPCC_SCWP)

Register	Offset	R/W	Description	Reset Value
SSPCC_SCWP	SSPCC_BA+0x204	R/W	Security Configuration Write Protection Register	0x0000_0000

31	30	29	28	27	26	25	24
WVCODE							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
WVCODE							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**LOCK**	**ENABLE**

Bits	Description	
[31:16]	**WVCODE**	**Write Verify Code** Read operation: Reserved, all zeros. Write operation: 0x475A = The write verify code, 0x475A, is needed to do a valid write to SSPCC_SCWP. Others = Invalid write verify code.
[15:2]	**Reserved**	Reserved.
[1]	**LOCK**	**Enable Write Protection Lock Bit (Write Once)** 0 = Write protection lock Disabled. 1 = Write protection Enabled and locked. **Note:** This bit cannot be cleared to 0 after set to 1.
[0]	**ENABLE**	**Write Protection Enable Bit** 0 = Write Protection Disabled. 1 = Write Protection Enabled.

DPM A35 Secure Debug State Register (DPM_A35SDS)

Register	Offset	R/W	Description	Reset Value
DPM_A35SDS	DPM_BA+0x10	R/W	DPM A35 Secure Debug State Register	0xA500_000X

31	30	29	28	27	26	25	24
VCODE							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					OLOCK	ODIS	REGDEN

Bits	Description	
[31:24]	**VCODE**	**Write Verify Code and Read Verify Code** Read operation: 0xA5 = The read access for DPM_CTL is correct. Others = The read access for DPM_CTL is incorrect. Write operation: 0x5A = The write verify code, which is needed to do a valid write to DPM_CA35SDS. Others = Invalid write verify code.
[23:3]	**Reserved**	Reserved.
[2]	**OLOCK**	**Debug State OTP Lock Bit (Read Only)** Indicate the Lock bit stored in OTP.
[1]	**ODIS**	**Debug State OTP Disable Bit (Read Only)** Indicate the Disable bit stored in OTP.
[0]	**REGDEN**	**Debug State Register Enable Bit** 0 = Set the debug state to CLOSE state when current state is OPEN state. 1 = Set the debug state to OPEN state when current state is CLOSE state. **Note:** If SINFAEN.SSPCC is set to high, this bit becomes read-only by Non-Secure software.

DPM A35 Secure Non-invasive Debug State Register (DPM_A35SNDS)

Register	Offset	R/W	Description	Reset Value
DPM_A35SNDS	DPM_BA+0x14	R/W	DPM A35 Secure Non-invasive Debug State Register	0xA500_000X

31	30	29	28	27	26	25	24
VCODE							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					OLOCK	ODIS	REGDEN

Bits	Description	
[31:24]	**VCODE**	**Write Verify Code and Read Verify Code** Read operation: 0xA5 = The read access for DPM_CTL is correct. Others = The read access for DPM_CTL is incorrect. Write operation: 0x5A = The write verify code, which is needed to do a valid write to DPM_CA35SDS. Others = Invalid write verify code.
[23:3]	**Reserved**	Reserved.
[2]	**OLOCK**	**Debug State OTP Lock Bit (Read Only)** Indicate the Lock bit stored in OTP.
[1]	**ODIS**	**Debug State OTP Disable Bit (Read Only)** Indicate the Disable bit stored in OTP.
[0]	**REGDEN**	**Debug State Register Enable Bit** 0 = Set the debug state to CLOSE state when current state is OPEN state. 1 = Set the debug state to OPEN state when current state is CLOSE state. **Note:** If SINFAEN.SSPCC is set to high, this bit becomes read-only by Non-Secure software.

DPM A35 Non-secure Debug State Register (DPM_A35NSDS)

Register	Offset	R/W	Description	Reset Value
DPM_A35NSDS	DPM_BA+0x18	R/W	DPM A35 Non-secure Debug State Register	0xA500_000X

31	30	29	28	27	26	25	24
VCODE							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					OLOCK	ODIS	REGDEN

Bits	Description	
[31:24]	**VCODE**	**Write Verify Code and Read Verify Code** Read operation: 0xA5 = The read access for DPM_CTL is correct. Others = The read access for DPM_CTL is incorrect. Write operation: 0x5A = The write verify code, which is needed to do a valid write to DPM_CA35SDS. Others = Invalid write verify code.
[23:3]	**Reserved**	Reserved.
[2]	**OLOCK**	**Debug State OTP Lock Bit (Read Only)** Indicate the Lock bit stored in OTP.
[1]	**ODIS**	**Debug State OTP Disable Bit (Read Only)** Indicate the Disable bit stored in OTP.
[0]	**REGDEN**	**Debug State Register Enable Bit** 0 = Set the debug state to CLOSE state when current state is OPEN state. 1 = Set the debug state to OPEN state when current state is CLOSE state. **Note:** If SINFAEN.SSPCC is set to high, this bit becomes read-only by Non-Secure software.

DPM A35 Non-secure Non-invasive Debug State Register (DPM_A35NSNDS)

Register	Offset	R/W	Description	Reset Value
DPM_A35NSNDS	DPM_BA+0x1C	R/W	DPM A35 Non-secure Non-invasive Debug State Register	0xA500_000X

31	30	29	28	27	26	25	24
VCODE							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					OLOCK	ODIS	REGDEN

Bits	Description	
[31:24]	**VCODE**	**Write Verify Code and Read Verify Code** Read operation: 0xA5 = The read access for DPM_CTL is correct. Others = The read access for DPM_CTL is incorrect. Write operation: 0x5A = The write verify code, which is needed to do a valid write to DPM_CA35SDS. Others = Invalid write verify code.
[23:3]	**Reserved**	Reserved.
[2]	**OLOCK**	**Debug State OTP Lock Bit (Read Only)** Indicate the Lock bit stored in OTP.
[1]	**ODIS**	**Debug State OTP Disable Bit (Read Only)** Indicate the Disable bit stored in OTP.
[0]	**REGDEN**	**Debug State Register Enable Bit** 0 = Set the debug state to CLOSE state when current state is OPEN state. 1 = Set the debug state to OPEN state when current state is CLOSE state. **Note:** If SINFAEN.SSPCC is set to high, this bit becomes read-only by Non-Secure software.

DPM M4 Debug State Register (DPM_M4DS)

Register	Offset	R/W	Description	Reset Value
DPM_M4DS	DPM_BA+0x20	R/W	DPM M4 Debug State Register	0xA500_000X

31	30	29	28	27	26	25	24
VCODE							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					OLOCK	ODIS	REGDEN

Bits	Description	
[31:24]	**VCODE**	**Write Verify Code and Read Verify Code** Read operation: 0xA5 = The read access for DPM_CTL is correct. Others = The read access for DPM_CTL is incorrect. Write operation: 0x5A = The write verify code, which is needed to do a valid write to DPM_CA35SDS. Others = Invalid write verify code.
[23:3]	**Reserved**	Reserved.
[2]	**OLOCK**	**Debug State OTP Lock Bit (Read Only)** Indicate the Lock bit stored in OTP.
[1]	**ODIS**	**Debug State OTP Disable Bit (Read Only)** Indicate the Disable bit stored in OTP.
[0]	**REGDEN**	**Debug State Register Enable Bit** 0 = Set the debug state to CLOSE state when current state is OPEN state. 1 = Set the debug state to OPEN state when current state is CLOSE state. **Note:** If SINFAEN.SSPCC is set to high, this bit becomes read-only by Non-Secure software.

DPM M4 Non-invasive Debug State Register (DPM_M4NDS)

Register	Offset	R/W	Description	Reset Value
DPM_M4NDS	DPM_BA+0x24	R/W	DPM M4 Non-invasive Debug State Register	0xA500_000X

31	30	29	28	27	26	25	24
VCODE							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					OLOCK	ODIS	REGDEN

Bits	Description	
[31:24]	**VCODE**	**Write Verify Code and Read Verify Code** Read operation: 0xA5 = The read access for DPM_CTL is correct. Others = The read access for DPM_CTL is incorrect. Write operation: 0x5A = The write verify code, which is needed to do a valid write to DPM_CA35SDS. Others = Invalid write verify code.
[23:3]	**Reserved**	Reserved.
[2]	**OLOCK**	**Debug State OTP Lock Bit (Read Only)** Indicate the Lock bit stored in OTP.
[1]	**ODIS**	**Debug State OTP Disable Bit (Read Only)** Indicate the Disable bit stored in OTP.
[0]	**REGDEN**	**Debug State Register Enable Bit** 0 = Set the debug state to CLOSE state when current state is OPEN state. 1 = Set the debug state to OPEN state when current state is CLOSE state. **Note:** If SINFAEN.SSPCC is set to high, this bit becomes read-only by Non-Secure software.

DPM External Debug State Register (DPM_EXTDS)

Register	Offset	R/W	Description	Reset Value
DPM_EXTDS	DPM_BA+0x30	R/W	DPM External Debug State Register	0xA500_000X

31	30	29	28	27	26	25	24
VCODE							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					OLOCK	ODIS	REGDEN

Bits	Description	
[31:24]	**VCODE**	**Write Verify Code and Read Verify Code** Read operation: 0xA5 = The read access for DPM_CTL is correct. Others = The read access for DPM_CTL is incorrect. Write operation: 0x5A = The write verify code, which is needed to do a valid write to DPM_CA35SDS. Others = Invalid write verify code.
[23:3]	**Reserved**	Reserved.
[2]	**OLOCK**	**Debug State OTP Lock Bit (Read Only)** Indicate the Lock bit stored in OTP.
[1]	**ODIS**	**Debug State OTP Disable Bit (Read Only)** Indicate the Disable bit stored in OTP.
[0]	**REGDEN**	**Debug State Register Enable Bit** 0 = Set the debug state to CLOSE state when current state is OPEN state. 1 = Set the debug state to OPEN state when current state is CLOSE state. **Note:** If SINFAEN.SSPCC is set to high, this bit becomes read-only by Non-Secure software.

DPM External Tracing Debug State Register (DPM_EXTTDS)

Register	Offset	R/W	Description	Reset Value
DPM_EXTTDS	DPM_BA+0x34	R/W	DPM External Tracing Debug State Register	0xA500_000X

31	30	29	28	27	26	25	24
VCODE							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					OLOCK	ODIS	REGDEN

Bits	Description	
[31:24]	**VCODE**	**Write Verify Code and Read Verify Code** Read operation: 0xA5 = The read access for DPM_CTL is correct. Others = The read access for DPM_CTL is incorrect. Write operation: 0x5A = The write verify code, which is needed to do a valid write to DPM_CA35SDS. Others = Invalid write verify code.
[23:3]	**Reserved**	Reserved.
[2]	**OLOCK**	**Debug State OTP Lock Bit (Read Only)** Indicate the Lock bit stored in OTP.
[1]	**ODIS**	**Debug State OTP Disable Bit (Read Only)** Indicate the Disable bit stored in OTP.
[0]	**REGDEN**	**Debug State Register Enable Bit** 0 = Set the debug state to CLOSE state when current state is OPEN state. 1 = Set the debug state to OPEN state when current state is CLOSE state. **Note:** If SINFAEN.SSPCC is set to high, this bit becomes read-only by Non-Secure software.

Product Life-cycle Status Register (PLM_STS)

Register	Offset	R/W	Description	Reset Value
PLM_STS	PLM_BA+0x04	R	Product Life-cycle Status Register (TZNS)	0xA500_000X

31	30	29	28	27	26	25	24
RVCODE							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					STAGE		

Bits	Description	
[31:24]	**RVCODE**	**Read Verify Code (Read Only)** Read Verify Code should be 0xA5
[23:3]	**Reserved**	Reserved.
[2:0]	**STAGE**	**Current Product Lifecycle Stage (Read Only)** 0x1 = Develop Stage. 0x3 = Deployed Stage. 0x7 = RMA Stage. 0x5 = RMA Stage. Others = Error stage.

6.7 SSMCC

6.7.1 Overview

System Security Memory Configuration Controller, SSMCC, is the module to manage the access authority of external DRAM. SSMCC provides the access authority, including TrustZone authority and SubM region authority, which follows the pre-defined memory access policy but has more flexibility.

6.7.2 Features

- Configure security attribution of DRAM based on address range
- Two Arm TZC400 controllers controlling seven AXI channels in total
- Security violation report and interrupt
- Programmable DRAM region for SubM system
- Write Protect of security configuration to prevent accidental change.

6.7.3 Block Diagram

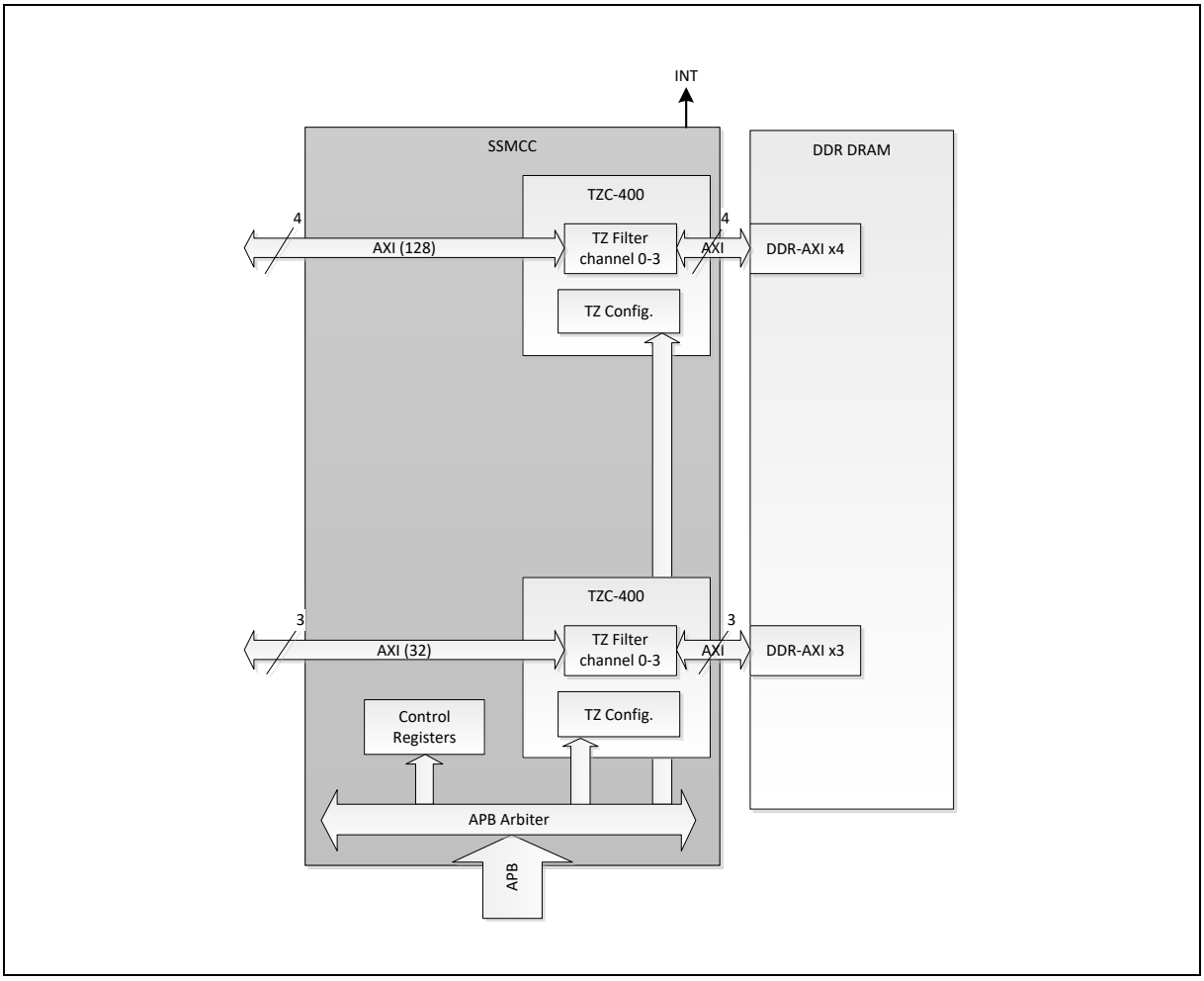


Figure 6.7-1 SSMCC Block Diagram

6.7.4 Functional Description

SSMCC is a module used to configure the security attribution of external DRAM and also check the

access policy of all accesses to external DRAM. To achieve that, SSMCC utilizes the Arm TZC400 module. By means of TZC400, the whole DRAM region can be divided into many regions with different security attribution.

SSMCC is a TZS slave and accepts TZS access only. Secure user should take the responsibility to configure SSMCC correctly.

6.7.4.1 TZC400

Based on the bus architecture, there are two TZC400 instances in SSMCC to provide 7 filter channels to check access to the DRAM region. The region setting of each TZC400 instance is independent, relying on the secure software to configure the region setting correctly and consistently among these instances.

TZC400_0 has four filter channels, servicing CA35, AXI-AP, Graphic engine (GFX), Display controller (DCUltra), and GMAC masters. Thus, the security configuration setting of TZC400_0 only has effects on these masters. The same rule applies to TZC400_2. The masters related to the filter channel of each TZC400 instance and the corresponding master ID are listed in Table 6.7-1.

The master ID is used to indicate an access is issued by which master and will be shown in TZC400 FAIL_ID register when the access violates some rule.

Filter Channel	**Accessible Master And ID**
TZC400_0 channel0	CA35: 0x00 + 6'bxxxxxx AXI-AP: 0x40 + 6'bxxxxxx
TZC400_0 channel1	GFX: 0x100 + 4'hx VC8000: 0x100 + 8'hxx
TZC400_0 channel2	DCUltra: 0x0 + 4'hx
TZC400_0 channel3	GMAC0: 0x00 + 4'hx GMAC1: 0x10 + 4'hx
TZC400_2 channel0	CCAP0: 0x1 CCAP1: 0x2
TZC400_2 channel1	CM4: 0x02 PDMA0: 0x04 PDMA1:0x05 PDMA2: 0x06 PDMA3:0x07 SDH0: 0x08 SDH1: 0x09 USBH0/ HSUSBH0: 0x0B USBH1/ HSUSBH1: 0x0C HSUSBD: 0x0D NANDC: 0x0E
TZC400_2 channel2	Crypto: 0x0

Table 6.7-1 TZC400 Filter Channels and Corresponding Masters

For more information of using TZC400, refer to "Arm CoreLink TZC400 Technical Reference Manual."

When a master intends to access DRAM, the corresponding setting of TZC400 in SSMCC should be configured correctly first. A typical flow of setting SSMCC includes the following steps.

1. Open the GATE_KEEPER

Before using the TZC400 channel to access DRAM, it is necessary to open the corresponding gate before accessing DRAM. For example, TZC400_0 channel 0, which is the access path of A35 to access DRAM, should be set to open through TZC400_0.GATE_KEEPER register.

2. Configure Regions for TZS Access

Each TZC400 region has a REGION_ATTRIBUTES register to enable/disable the region for TZS access. To allow TZS to access the region, secure software should set the REGION_ATTRIBUTES register of target region to the corresponding TZC400 instance. For example, to allow Secure A35 to access DRAM Region0. The REGION_ATTRIBUTES_0 register of TZC400_0 instance should be set to allow secure read and/or write.

3. Configure Regions for TZNS, SubM Access

The ability of TZNS and SubM to access specific region is configured through REGION_ID_ACCESS register of the corresponding region. NSAID[0] and NSAID[16] indicates the access permission for TZNS master read and write respectively while NSAID[1] and NSAID[17] indicates the access permission for SubM master read and write respectively.

Note that before programming value to a register related to a channel or channels, the clock of corresponding channel(s) must be turned on. For example, when programming the gate_keeper register of TZC400_0, the clock of GFX, DCUltra and GMAC, refer to CLK_SYSCLK0 register, must be turned on because this TZC400 instance is related to these channels.

Note that before programming a value to a register related to a channel or channels, the core reset of DRAM controller, refer to SYS_IPRST0[29] in System Manager section, must be checked to be released.

6.7.4.2 Interrupt

Based on TZC400, SSMCC provides information about security violation of the access to the DRAM region. The interrupt of SSMCC is the combination of two TZC400 interrupts, as shown in Figure 6.7-2.

In addition to security violation event, SSMCC also reports other fail event defined in TZC400 design, such as overlapping address. Refer to “Arm CoreLink TZC400 Technical Reference Manual” for more details.

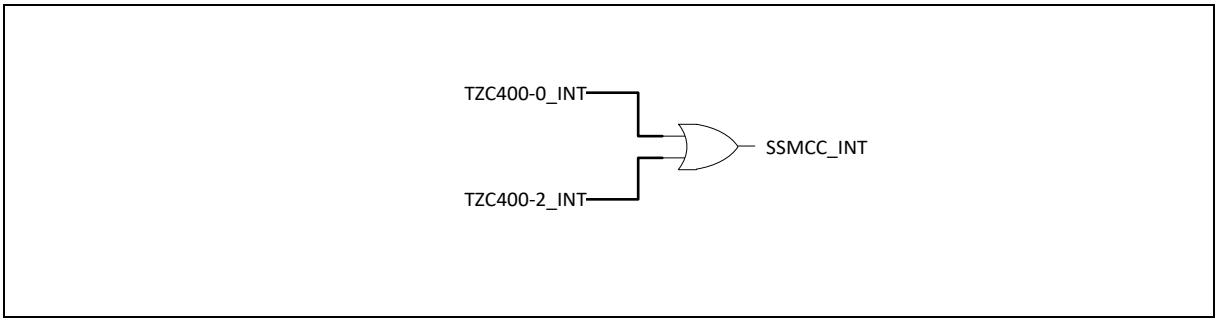


Figure 6.7-2 Interrupt Route of SSMCC

6.7.4.3 Security Configuration

SSMCC utilizes Arm TZC400 to define up to 9 DRAM regions with individual security attribution. The granularity of each region is 4 Kbytes. Each region can be set to allow Secure access or Non-secure access or SubM access by setting TZC400 registers with “REGION_” as prefix.

TZC400 provides a mechanism to set DRAM Memory flexibly. Each region can be set to allow TZS access, TZNS access or SubM access independently. Secure software should take the responsibility to set SSMCC to configure the system security correctly.

To follow memory access policy defined in SSPCC DTS, it is recommended to configure SSMCC to set DRAM to three kinds of regions: TZS, TZNS and ShareMemory. The property of each region is listed in Table 6.7-2.

Memory Region Type	Description Of Security Attribution
TZS	The memory region is TZS
TZNS	The memory region is TZNS
ShareMemory	The memory region is accessible by all masters including SubM

Table 6.7-2 Security Attribution of DRAM

A recommended partition of DRAM memory using SSMCC is listed below and shown in Table 6.7-3:

- The Region0 is set to accept TZS access which means that TZS master should be able to access all DRAM region if no additional constraint. The s_rd_en and s_wr_en bits in REGION_ATTRIBUTES register of Region0 should be set to 1 to make the region to accept TZS access.
- The Region1 is configured to range from 0x8000_0000 to 0x8040_0000, which is the first 4 Mbytes region of DRAM, and is set to accept accesses of TZS, TZNS and SubM so that Region1 has the security attribution type of "ShareMemory." The s_rd_en, s_wr_en bit in REGION_ATTRIBUTES register of Region1 should be set to 1 and filter_en bits in REGION_ATTRIBUTES register of Region1 should be set to 1 to all filter channels to let Region1 effects on all filter channels. Besides, the nsaid_wr_en[1:0] and nsaid_rd_en[1:0] bits should all be set to 1 to make Region1 to accept TZNS and SubM accesses.
- If TZNS type of memory is required, Region2 to Region8 can be used. The difference of TZNS-type memory from ShareMemory-type memory is that the nsaid_wr_en[1] and nsaid_rd_en[1] is set to 0 to disallow SubM access.

For the convenience and consistency, all regions of all TZC400 instances can use the same settings.

	Whole DRAM Address Space			
TZC400 Regions	Region0			
	Region1 (4 Mbytes)		Region2 (User-defined range)	
S_RD_EN, S_WR_EN (TZS Accessible)	Set	Set	Set	Set
nsaid_rd_en[0], nsaid_wr_en[0] (TZNS Accessible)	Set	Unset	Set	Unset
nsaid_rd_en[1], nsaid_wr_en[1] (SubM Accessible)	Set	Unset	Unset	Unset
Security Attribution	ShareMemory	TZS	TZNS	TZS

Table 6.7-3 Example of SSMCC Region Configuration

6.7.4.4 Write Protect of Security Configuration

To prevent security configuration to be changed accidently causing information leakage and/or severe fault, user can turn on the Write Protect mechanism after setting the security configuration.

The security configuration registers are listed in Table 6.7-4, which will be influenced by the Write Protect mechanism.

Offset	Register Name
0x004	ACTION
0x008	GATE_KEEPER
0x00C	SPECULATION_CTRL
0x110	REGION_ATTRIBUTES_0
0x114	REGION_ID_ACCESS_0
0x100+(0x20*n), n=1,2…8	REGION_BASE_LOW_<n>
0x104+(0x20*n), n=1,2…8	REGION_BASE_HIGH_<n>
0x108+(0x20*n), n=1,2…8	REGION_TOP_LOW_<n>
0x10C+(0x20*n), n=1,2…8	REGION_TOP_HIGH_<n>
0x110+(0x20*n), n=1,2…8	REGION_ATTRIBUTES_<n>
0x114+(0x20*n), n=1,2…8	REGION_ID_ACCESS_<n>

Table 6.7-4 Security Configuration Registers of TZC400

The Write Protect mechanism is disabled state by default after reset, and can be turned on by setting SSMCC_SCWP register. The enable bit in SSMCC_SCWP register is 0 by default, which means the Write Protect is disabled and security configurations are able to be set. When the security configurations has been initialized, user can set 1 to the enable bit, SSMCC_SCWP[0], with a valid write verify code, to enable Write Protect. When Write Protect is enabled, all security configuration registers are not able to be set before unlocking the Write Protect.

In addition to Enable bit, there is a Lock bit to enable the Write Protect. When the security configurations have been initialized and user intends to prevent further change of security configuration, the lock bit of SSMCC_SCWP[1] can be used. This bit, like SSMCC_SCWP[0], makes the ability of setting security configuration disabled. The difference is that after this bit is set to 1, it can't be cleared to 0 without a system-level reset.

6.7.5 Register Map

R: read only, **W**: write only, **R/W**: both read and write, **C**: Only value 0 can be written

Register	Offset	R/W	Description	Reset Value
SSMCC Base Address: **SSMCC_BA = 0x404E_0000** **TZC0_BA = 0x404E_1000** **TZC2_BA = 0x404E_3000**				
SSMCC_SCWP	SSMCC_BA+0x00	R/W	Security Configuration Write Protect Register	0x00000000
TZC0_BUILD_CONFIG	TZC0_BA+0x000	R	Provides information about the configuration of the TZC-400.	0x03001f08
TZC0_ACTION	TZC0_BA+0x004	R/W	Controls the interrupt and bus response signaling behavior of the TZC-400 when region permission failures occur.	0x00000000
TZC0_GATE_KEEPER	TZC0_BA+0x008	R/W	Provides control and status for the gate keeper in each filter unit implemented	0x00000000
TZC0_SPECULATION_CTRL	TZC0_BA+0x00C	R/W	Controls the read access speculation and write access speculation.	0x00000000
TZC0_INT_STATUS	TZC0_BA+0x010	R	Contains the status of the interrupt signal, TZCINT, that reports access security violations or region overlap errors.	0x00000000
TZC0_INT_CLEAR	TZC0_BA+0x014	W	Clears the interrupt.	0x00000000
TZC0_FAIL_ADDRESS_LOW_n **n=0,1,2,3**	TZC0_BA+0x020+(0x10*n)	R	Contains the lower 32 bits of the address of the first access that failed a region permission check in the associated filter unit. This occurs even if the ACTION register is set to not drive the interrupt signal. The first failing access is defined as the first permission failure in the associated filter unit after one of the following occurs: • Reset. • The last clearance of an interrupt using the INT_CLEAR register.	0x00000000
TZC0_FAIL_ADDRESS_HIGH_n **n=0,1,2,3**	TZC0_BA+0x024+(0x10*n)	R	Reserved	0x00000000
TZC0_FAIL_CONTROL_n **n=0,1,2,3**	TZC0_BA+0x028+(0x10*n)	R	Contains the control status information of the first access that failed a region permission check in the associated filter unit. This occurs even if the ACTION register is set to not drive the interrupt signal. The first failing access is defined as the first permission failure in the associated filter unit after one of the following occurs: • Reset. • The last clearance of an interrupt using the INT_CLEAR register.	0x00000000
TZC0_FAIL_ID_n **n=0,1,2,3**	TZC0_BA+0x02C+(0x10*n)	R	Contains the master ACE-Lite ARID or AWID of the first access that failed a region permission check in the associated filter unit. This occurs even if the ACTION register is set to not drive the interrupt signal. The first failing access is defined as the first permission failure in the associated filter unit after one of the following occurs:	0x00000000

			• Reset. • The last clearance of an interrupt using the INT_CLEAR register.	
TZC0_REGION_BASE_LOW_0	TZC0_BA+0x100	R	This register is read-only and is hard-wired to all zeros.	0x00000000
TZC0_REGION_BASE_HIGH_0	TZC0_BA+0x104	R	This register is read-only and is hard-wired to all zeros.	0x00000000
TZC0_REGION_TOP_LOW_0	TZC0_BA+0x108	R	This register is read-only and is hard-wired to all 1.	0xFFFFFFFF
TZC0_REGION_TOP_HIGH_0	TZC0_BA+0x10C	R	This register is read-only and is hard-wired to all zeros.	0x00000000
TZC0_REGION_ATTRIBUTES_0	TZC0_BA+0x110	R/W	Controls the permissions for Region 0.	0x00000000
TZC0_REGION_ID_ACCESS_0	TZC0_BA+0x114	R/W	Controls the Non-secure access based on the NSAID inputs for Region 0.	0x00000001
TZC0_REGION_BASE_LOW_n **n=1,2..8**	TZC0_BA+0x100+(0x20*n)	R/W	Controls the base address bits[31:12] of Region n.	0x00000000
TZC0_REGION_BASE_HIGH_n **n=1,2..8**	TZC0_BA+0x104+(0x20*n)	R/W	Reserved	0x00000000
TZC0_REGION_TOP_LOW_n **n=1,2..8**	TZC0_BA+0x108+(0x20*n)	R/W	Controls the region top address bits[31:12] of Region <n>. Region top address bits[11:0] are always 0xFFF because the TZC-400 does not permit the region size to be less than 4 Kbytes. Therefore, the region top address points to the top byte of a 4 Kbytes region.	0x00000FFF
TZC0_REGION_TOP_HIGH_n **n=1,2..8**	TZC0_BA+0x10C+(0x20*n)	R/W	Reserved	0x00000000
TZC0_REGION_ATTRIBUTES_n **n=1,2..8**	TZC0_BA+0x110+(0x20*n)	R/W	Controls the permissions and target filter region enables.	0x00000000
TZC0_REGION_ID_ACCESS_n **n=1,2..8**	TZC0_BA+0x114+(0x20*n)	R/W	Controls the Non-secure access based on the NSAID inputs.	0x00000000
TZC0_PID4	TZC0_BA+0xFD0	R	Provides the following information about the peripheral configuration • 4 Kbytes_count. • Jep106_c_code.	0x04
TZC0_PID0	TZC0_BA+0xFE0	R	The PID0 register provides the following information about the peripheral configuration: • part_number_0.	0x60
TZC0_PID1	TZC0_BA+0xFE4	R	The PID1 register provides the following information about the peripheral configuration: • part_number_1. • Jep106_id_3_0.	0xB4
TZC0_PID2	TZC0_BA+0xFE8	R	The PID2 register provides the following information about the peripheral configuration: • Jep106_id_6_4. • Revision number. • JEDEC use flag.	0x2B

TZC0_PID3	TZC0_BA+0xFEC	R	The PID3 register provides the following information about the peripheral configuration: • Mod Number. • RevAnd.	0x00
TZC0_CID0	TZC0_BA+0xFF0	R	This is one of four 8-bit registers that together hold a 32-bit component ID value.	0x0D
TZC0_CID1	TZC0_BA+0xFF4	R	This is one of four 8-bit registers, that together hold a 32-bit component ID value.	0xF0
TZC0_CID2	TZC0_BA+0xFF8	R	This is one of four 8-bit registers, that together hold a 32-bit component ID value.	0x05
TZC0_CID3	TZC0_BA+0xFFC	R	This is one of four 8-bit registers that together hold a 32-bit component ID value.	0xB1
TZC2_BUILD_CONFIG	TZC2_BA+0x000	R	Provides information about the configuration of the TZC-400.	0x03001f08
TZC2_ACTION	TZC2_BA+0x004	R/W	Controls the interrupt and bus response signaling behavior of the TZC-400 when region permission failures occur.	0x00000000
TZC2_GATE_KEEPER	TZC2_BA+0x008	R/W	Provides control and status for the gate keeper in each filter unit implemented	0x00000000
TZC2_SPECULATION_CTRL	TZC2_BA+0x00C	R/W	Controls the read access speculation and write access speculation.	0x00000000
TZC2_INT_STATUS	TZC2_BA+0x010	R	Contains the status of the interrupt signal, TZCINT, that reports access security violations or region overlap errors.	0x00000000
TZC2_INT_CLEAR	TZC2_BA+0x014	W	Clears the interrupt.	0x00000000
TZC2_FAIL_ADDRESS_LOW_n **n=0,1,2**	TZC2_BA+0x020+(0x10*n)	R	Contains the lower 32 bits of the address of the first access that failed a region permission check in the associated filter unit. This occurs even if the ACTION register is set to not drive the interrupt signal. The first failing access is defined as the first permission failure in the associated filter unit after one of the following occurs: • Reset. • The last clearance of an interrupt using the INT_CLEAR register.	0x00000000
TZC2_FAIL_ADDRESS_HIGH_n **n=0,1,2**	TZC2_BA+0x024+(0x10*n)	R	Reserved	0x00000000
TZC2_FAIL_CONTROL_n **n=0,1,2**	TZC2_BA+0x028+(0x10*n)	R	Contains the control status information of the first access that failed a region permission check in the associated filter unit. This occurs even if the ACTION register is set to not drive the interrupt signal. The first failing access is defined as the first permission failure in the associated filter unit after one of the following occurs: • Reset. • The last clearance of an interrupt using the INT_CLEAR register.	0x00000000
TZC2_FAIL_ID_n **n=0,1,2**	TZC2_BA+0x02C+(0x10*n)	R	Contains the master ACE-Lite ARID or AWID of the first access that failed a region permission check in the associated filter unit. This occurs even if the ACTION register is set to not drive the interrupt	0x00000000

			signal. The first failing access is defined as the first permission failure in the associated filter unit after one of the following occurs: • Reset. • The last clearance of an interrupt using the INT_CLEAR register.	
TZC2_REGION_BASE_LOW_0	TZC2_BA+0x100	R	This register is read-only and is hard-wired to all zeros.	0x00000000
TZC2_REGION_BASE_HIGH_0	TZC2_BA+0x104	R	This register is read-only and is hard-wired to all zeros.	0x00000000
TZC2_REGION_TOP_LOW_0	TZC2_BA+0x108	R	This register is read-only and is hard-wired to all 1.	0xFFFFFFFF
TZC2_REGION_TOP_HIGH_0	TZC2_BA+0x10C	R	This register is read-only and is hard-wired to all zeros.	0x00000000
TZC2_REGION_ATTRIBUTES_0	TZC2_BA+0x110	R/W	Controls the permissions for Region 0.	0x00000000
TZC2_REGION_ID_ACCESS_0	TZC2_BA+0x114	R/W	Controls the Non-secure access based on the NSAID inputs for Region 0.	0x00000001
TZC2_REGION_BASE_LOW_n **n=1,2..8**	TZC2_BA+0x100+(0x20*n)	R/W	Controls the base address bits[31:12] of Region n.	0x00000000
TZC2_REGION_BASE_HIGH_n **n=1,2..8**	TZC2_BA+0x104+(0x20*n)	R/W	Reserved	0x00000000
TZC2_REGION_TOP_LOW_n **n=1,2..8**	TZC2_BA+0x108+(0x20*n)	R/W	Controls the region top address bits[31:12] of Region <n>. Region top address bits[11:0] are always 0xFFF because the TZC-400 does not permit the region size to be less than 4 Kbytes. Therefore, the region top address points to the top byte of a 4 Kbytes region.	0x00000FFF
TZC2_REGION_TOP_HIGH_n **n=1,2..8**	TZC2_BA+0x10C+(0x20*n)	R/W	Reserved	0x00000000
TZC2_REGION_ATTRIBUTES_n **n=1,2..8**	TZC2_BA+0x110+(0x20*n)	R/W	Controls the permissions and target filter region enables.	0x00000000
TZC2_REGION_ID_ACCESS_n **n=1,2..8**	TZC2_BA+0x114+(0x20*n)	R/W	Controls the Non-secure access based on the NSAID inputs.	0x00000000
TZC2_PID4	TZC2_BA+0xFD0	R	Provides the following information about the peripheral configuration • 4 Kbytes_count. • Jep106_c_code.	0x04
TZC2_PID0	TZC2_BA+0xFE0	R	The PID0 register provides the following information about the peripheral configuration: • part_number_0.	0x60
TZC2_PID1	TZC2_BA+0xFE4	R	The PID1 register provides the following information about the peripheral configuration: • part_number_1. • Jep106_id_3_0.	0xB4
TZC2_PID2	TZC2_BA+0xFE8	R	The PID2 register provides the following information about the peripheral configuration: • Jep106_id_6_4.	0x2B

			• Revision number. • JEDEC use flag.	
TZC2_PID3	TZC2_BA+0xFEC	R	The PID3 register provides the following information about the peripheral configuration: • Mod Number. • RevAnd.	0x00
TZC2_CID0	TZC2_BA+0xFF0	R	This is one of four 8-bit registers that together hold a 32-bit component ID value.	0x0D
TZC2_CID1	TZC2_BA+0xFF4	R	This is one of four 8-bit registers, that together hold a 32-bit component ID value.	0xF0
TZC2_CID2	TZC2_BA+0xFF8	R	This is one of four 8-bit registers, that together hold a 32-bit component ID value.	0x05
TZC2_CID3	TZC2_BA+0xFFC	R	This is one of four 8-bit registers that together hold a 32-bit component ID value.	0xB1

6.7.6 Register Description

SSMCC Security Configuration Write Protect Register

Register	Offset	R/W	Description	Reset Value
SSMCC_SCWP	SSMCC_BA+0x00	R/W	Security Configuration Write Protect Register	0x00000000

31	30	29	28	27	26	25	24
WVCODE							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
WVCODE							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**LOCK**	**ENABLE**

Bits	Description	
[31:16]	**WVCODE**	**Write Verify Code** Read operation: Reserved, all zeros. Write operation: 0x475A = The write verify code, 0x475A, is needed to do a valid write to SSMCC_SCWP. Others = Invalid write verify code.
[15:2]	**Reserved**	Reserved.
[1]	**LOCK**	**Write Protect Enable and Lock Bit (write 1 to set)** 0 = Security Configuration Write Protect Disabled. 1 = Security Configuration Write Protect Enabled. **Note:** This bit can only be cleared by system reset.
[0]	**ENABLE**	**Write Protect Enable Bit** 0 = Security Configuration Write Protect Disabled. 1 = Security Configuration Write Protect Enabled.

BUILD CONFIG Register (BUILD_CONFIG)

Register	Offset	R/W	Description	Reset Value
TZC0_BUILD_CONFIG	TZC0_BA+0x000	R	Provides information about the configuration of the TZC-400.	0x03001f08
TZC2_BUILD_CONFIG	TZC2_BA+0x000	R	Provides information about the configuration of the TZC-400.	0x03001f08

31	30	29	28	27	26	25	24
Reserved						no_of_filters	
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		address_width					
7	6	5	4	3	2	1	0
Reserved			no_of_regions				

Bits	Description	
[31:26]	**Reserved**	Reserved.
[25:24]	**no_of_filters**	**Defines the number of filter units in the design implementation:** 0b00 = One filter unit 0b01 = Two filter units 0b10 = Reserved 0b11 = Four filter units.
[23:14]	**Reserved**	Reserved.
[13:8]	**address_width**	**Defines the width of the ACE-Lite address bus:** 0b011111 = 32 bits 0b100011 = 36 bits 0b100111 = 40 bits 0b101111 = 48 bits 0b111111 = 64 bits All other values Reserved.
[7:5]	**Reserved**	Reserved.
[4:0]	**no_of_regions**	**Defines the number of regions that the TZC-400 provides:** 0b01000 Nine regions All other values Reserved.

ACTION Register (ACTION)

Register	Offset	R/W	Description	Reset Value
TZC0_ACTION	TZC0_BA+0x004	R/W	Controls the interrupt and bus response signaling behavior of the TZC-400 when region permission failures occur.	0x00000000
TZC2_ACTION	TZC2_BA+0x004	R/W	Controls the interrupt and bus response signaling behavior of the TZC-400 when region permission failures occur.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**reaction_value**	

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1:0]	**reaction_value**	Controls how the TZC-400 uses the BRESPS[1:0], RRESPS[1:0], and TZCINT signals when a region permission failure occurs, excluding region overlap errors The settings for these bits are: 0b00 = Sets TZCINT LOW and issues an OKAY response 0b01 = Sets TZCINT LOW and issues a DECERR response 0b10 = Sets TZCINT HIGH and issues an OKAY response 0b11 = Sets TZCINT HIGH and issues a DECERR response When a region overlap for region 1 and higher occurs, this field also determines how TZCINT is set. The settings are: 0b00, 0b01 = TZCINT LOW 0b10, 0b11 = TZCINT HIGH.

GATE KEEPER Register (GATE_KEEPER)

Register	Offset	R/W	Description	Reset Value
TZC0_GATE_KEEPER	TZC0_BA+0x008	R/W	Provides control and status for the gate keeper in each filter unit implemented	0x00000000
TZC2_GATE_KEEPER	TZC2_BA+0x008	R/W	Provides control and status for the gate keeper in each filter unit implemented	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				open_status			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				open_request			

Bits	Description	
[31:20]	**Reserved**	Reserved.
[19:16]	**open_status**	The current state of the gate keeper in each filter unit The bit associations are as follows: bit 16 Filter 0 gate keeper status bit 17 Filter 1 gate keeper status bit 18 Filter 2 gate keeper status bit 19 Filter 3 gate keeper status When a bit is set to 1, the gate keeper permits access to its associated filter, that is, it is open When a bit is set to 0, the gate keeper no longer permits access to its associated filter, that is, it is closed This bit is set to 0 when both of the following conditions are fulfilled:This means that the gate keeper always waits for outstanding accesses to complete •The gate keeper no longer permits access to its associated filter •All outstanding accesses through the filter unit are complete If any of the associated filter units are not implemented, the corresponding gate keeper bits are unused,
[15:4]	**Reserved**	Reserved.
[3:0]	**open_request**	Each bit in this field requests the gate of the associated filter unit to be open or closed Each bit is associated as follows: bit 0 Filter 0 bit 1 Filter 1 bit 2 Filter 2 bit 3 Filter 3. Set the open_request bit to 1, to request the gate to be open Set the open_request bit to 0, to request the gate to be closed If any of the associated filter units are not implemented, the corresponding open_request bits are unused,

SPECULATION CTRL Register (SPECULATION_CTRL)

Register	Offset	R/W	Description	Reset Value
TZC0_SPECULATION_CTRL	TZC0_BA+0x00C	R/W	Controls the read access speculation and write access speculation.	0x00000000
TZC2_SPECULATION_CTRL	TZC2_BA+0x00C	R/W	Controls the read access speculation and write access speculation.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**write_spec_disable**	**read_spec_disable**

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**write_spec_disable**	Set this bit as follows: 0 = Enable write access speculation. This is the default setting. 1 = Disable write access speculation. **Note:**This bit is ignored and assumed to be zero at a filter unit if the corresponding QVNENABLE<x.
[0]	**read_spec_disable**	**Controlsread access speculation** **Note:**This bit is ignored and assumed to be zero at a filter unit if the corresponding QVNENABLE<x> signal is HIGH You can set this bit as follows: 0 = Enables read access speculation This is the default setting 1 = Disables read access speculation.

INT STATUS Register (INT_STATUS)

Register	Offset	R/W	Description	Reset Value
TZC0_INT_STATUS	TZC0_BA+0x010	R	Contains the status of the interrupt signal, TZCINT, that reports access security violations or region overlap errors.	0x00000000
TZC2_INT_STATUS	TZC2_BA+0x010	R	Contains the status of the interrupt signal, TZCINT, that reports access security violations or region overlap errors.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				overlap			
15	14	13	12	11	10	9	8
Reserved				overrun			
7	6	5	4	3	2	1	0
Reserved				status			

Bits	Description	
[31:20]	**Reserved**	Reserved.
[19:16]	**overlap**	The bit associations are as follows: bit 16 Filter 0 bit 17 Filter 1 bit 18 Filter 2 bit 19 Filter 3 When a bit is set to 1, it indicates a violation of the overlap region configuration rules for the associated filter unit This occurs when an access matches with two enabled regions at the same time unless the overlap is only with Region 0. This bit is set even if the ACTION register is set to not drive the interrupt When this bit is 1, the interrupt status bit is also set to 1. Clear the interrupt status of the associated bit to also clear this field
[15:12]	**Reserved**	Reserved.
[11:8]	**overrun**	The bit associations are as follows: bit 8 Filter 0 bit 9 Filter 1 bit 10 Filter 2 bit 11 Filter 3. When a bit is set to 1, it indicates the occurrence of two or more region permission or region overlapping failures at the associated filter unit after the interrupt was cleared by the associated bit This bit is set even if the ACTION register is set to not drive the interrupt Clear the interrupt status of the associated bit in the INT_CLEAR register to also clear this field.
[7:4]	**Reserved**	Reserved.

Bits	Description	
[3:0]	**status**	Each bit is associated as follows: bit 0 Filter 0 bit 1 Filter 1 bit 2 Filter 2 bit 3 Filter 3. Each bit in this field indicates the status of the interrupt from each filter unit as follows: 0 = Interrupt is not asserted 1 = Interrupt is asserted and waiting to be cleared This bit is set even if the ACTION register is set to not drive the interrupt output TZCINT HIGH Therefore, the status acts as an indicator that a region permission check failure or an overlap error has occurred at a particular filter unit.

INT CLEAR Register (INT_CLEAR)

Register	Offset	R/W	Description	Reset Value
TZC0_INT_CLEAR	TZC0_BA+0x014	W	Clears the interrupt.	0x00000000
TZC2_INT_CLEAR	TZC2_BA+0x014	W	Clears the interrupt.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**clear**			

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3:0]	**clear**	Contains the control bits to clear interrupts The bit associations are as follows: bit 0 Filter 0 bit 1 Filter 1 bit 2 Filter 2 bit 3 Filter 3. Write a 1 to any of these bits to clear the associated status, overrun, and overlap bits in the INT_STATUS register.

FAIL ADDRESS LOW <n> Register (FAIL_ADDRESS_LOW_n)

Register	Offset	R/W	Description	Reset Value
TZC0_FAIL_ADDRESS_LOW_n **n=0,1,2,3**	TZC0_BA+0x020+(0x10*n)	R	Contains the lower 32 bits of the address of the first access that failed a region permission check in the associated filter unit. This occurs even if the ACTION register is set to not drive the interrupt signal. The first failing access is defined as the first permission failure in the associated filter unit after one of the following occurs: • Reset. • The last clearance of an interrupt using the INT_CLEAR register.	0x00000000
TZC2_FAIL_ADDRESS_LOW_n **n=0,1,2**	TZC2_BA+0x020+(0x10*n)	R	Contains the lower 32 bits of the address of the first access that failed a region permission check in the associated filter unit. This occurs even if the ACTION register is set to not drive the interrupt signal. The first failing access is defined as the first permission failure in the associated filter unit after one of the following occurs: • Reset. • The last clearance of an interrupt using the INT_CLEAR register.	0x00000000

31	30	29	28	27	26	25	24
addr_status_low							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
addr_status_low							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
addr_status_low							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
addr_status_low							

Bits	Description	
[31:0]	**addr_status_low**	If a region permission check fails or a region overlap occurs, this field returns the ACE-Lite address bits [31:0. You must clear the associated interrupt status before this field can return the address of accesses of subsequent permission checks or region overlap failures This occurs even if the ACTION register does not enable the interrupt If the status flag for the filter unit in the INT_STATUS register is already set, new region permission check failures in the same filter unit do not update the associated fail status group of registers.

FAIL ADDRESS HIGH <n> Register (FAIL_ADDRESS_HIGH_<n>)

Register	Offset	R/W	Description	Reset Value
TZC0_FAIL_ADDRESS_HIGH_n **n=0,1,2,3**	TZC0_BA+0x024+(0x10*n)	R	Reserved	0x00000000
TZC2_FAIL_ADDRESS_HIGH_n **n=0,1,2**	TZC2_BA+0x024+(0x10*n)	R	Reserved	0x00000000

Bits	Description	
[31:0]	**Reserved**	Reserved.

FAIL CONTROL <n> Register (FAIL_CONTROL_<n>)

Register	Offset	R/W	Description	Reset Value
TZC0_FAIL_CONTROL_n **n=0,1,2,3**	TZC0_BA+0x028+(0x10*n)	R	Contains the control status information of the first access that failed a region permission check in the associated filter unit. This occurs even if the ACTION register is set to not drive the interrupt signal. The first failing access is defined as the first permission failure in the associated filter unit after one of the following occurs: • Reset. • The last clearance of an interrupt using the INT_CLEAR register.	0x00000000
TZC2_FAIL_CONTROL_n **n=0,1,2**	TZC2_BA+0x028+(0x10*n)	R	Contains the control status information of the first access that failed a region permission check in the associated filter unit. This occurs even if the ACTION register is set to not drive the interrupt signal. The first failing access is defined as the first permission failure in the associated filter unit after one of the following occurs: • Reset. • The last clearance of an interrupt using the INT_CLEAR register.	0x00000000

31	30	29	28	27	26	25	24
Reserved							Direction
23	22	21	20	19	18	17	16
Reserved		Non-secure	Privileged	Reserved			
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							

Bits	Description	
[31:25]	**Reserved**	Reserved.
[24]	**Direction**	If a region permission check fails or a region overlap occurs, this field indicates whether the failed access was a read or write access attempt You must clear the associated interrupt status before this field can return the direction of accesses of subsequent permission checks or region overlap failures 0 = Read access 1 = Write access If the status flag for the filter unit in the INT_STATUS register is already set, new region permission check failures in the same filter unit do not update the associated fail status group of registers.
[23:22]	**Reserved**	Reserved.

Bits	Description	
[21]	**Non-secure**	If a region permission check fails or a region overlap occurs, this field indicates whether it was a Secure or Non- secure access attempt You must clear the associated interrupt status before this field can return the direction of accesses of subsequent permission checks or region overlap failures 0 = Secure access 1 = Non-secure access If the status flag for the filter unit in the INT_STATUS register is already set, new region permission check failures in the same filter unit do not update the associated fail status group of registers.
[20]	**Privileged**	If a region permission check fails or a region overlap occurs, this field indicates whether it was an unprivileged or privileged access attempt You must clear the associated interrupt status before this field can return the values of accesses of subsequent permission checks or region overlap failures 0 = Unprivileged access 1 = Privileged access If the status flag for the filter unit in the INT_STATUS register is already set, new region permission check failures in the same filter unit do not update the associated fail status group of registers.
[19:0]	**Reserved**	Reserved.

FAIL ID <n> Register (FAIL_ID_<n>)

Register	Offset	R/W	Description	Reset Value
TZC0_FAIL_ID_n **n=0,1,2,3**	TZC0_BA+0x02C+(0x10*n)	R	Contains the master ACE-Lite ARID or AWID of the first access that failed a region permission check in the associated filter unit. This occurs even if the ACTION register is set to not drive the interrupt signal. The first failing access is defined as the first permission failure in the associated filter unit after one of the following occurs: • Reset. • The last clearance of an interrupt using the INT_CLEAR register.	0x00000000
TZC2_FAIL_ID_n **n=0,1,2**	TZC2_BA+0x02C+(0x10*n)	R	Contains the master ACE-Lite ARID or AWID of the first access that failed a region permission check in the associated filter unit. This occurs even if the ACTION register is set to not drive the interrupt signal. The first failing access is defined as the first permission failure in the associated filter unit after one of the following occurs: • Reset. • The last clearance of an interrupt using the INT_CLEAR register.	0x00000000

31	30	29	28	27	26	25	24
Reserved				vnet			
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		id					

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:24]	**vnet**	**If a region permission check fails or a region overlap occurs, this field returns the VN number of the first failed access, from either ARVNET<x> or AWVNET<x> as appropriate** If the status flag for the filter unit in the INT_STATUS register is already set, new region permission check failures in the same filter unit do not update the associated fail status group of registers.
[23:6]	**Reserved**	Reserved.
[5:0]	**id**	**If a region permission check fails or a region overlap occurs, this field returns the ACE-Lite ID values of the first failed access** If the status flag for the filter unit in the INT_STATUS register is already set, new region permission check failures in the same filter unit do not update the associated fail status group of registers.

REGION BASE LOW 0 Register (REGION_BASE_LOW_0)

Register	Offset	R/W	Description	Reset Value
TZC0_REGION_BASE_LOW_0	TZC0_BA+0x100	R	This register is read-only and is hard-wired to all zeros.	0x00000000
TZC2_REGION_BASE_LOW_0	TZC2_BA+0x100	R	This register is read-only and is hard-wired to all zeros.	0x00000000

31	30	29	28	27	26	25	24
base_address_low<n>							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
base_address_low<n>							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
base_address_low<n>				Reserved			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:12]	**base_address_low<*n*>**	**Controls the base address bits[31:12] of Region <*n*>** For Region 0, this field is read-only. The TZC-400 sets the base address of Region 0 to 0x0.
[11:0]	**Reserved**	Reserved.

REGION_BASE_HIGH_0 Register (REGION_BASE_HIGH_0)

Register	Offset	R/W	Description	Reset Value
TZC0_REGION_BASE_HIGH_0	TZC0_BA+0x104	R	This register is read-only and is hard-wired to all zeros.	0x00000000
TZC2_REGION_BASE_HIGH_0	TZC2_BA+0x104	R	This register is read-only and is hard-wired to all zeros.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							

Bits	Description	
[31:0]	**Reserved**	Reserved.

REGION_TOP_LOW_0 Register (REGION_TOP_LOW_0)

Register	Offset	R/W	Description	Reset Value
TZC0_REGION_TOP_LOW_0	TZC0_BA+0x108	R	This register is read-only and is hard-wired to all 1.	0xFFFFFFFF
TZC2_REGION_TOP_LOW_0	TZC2_BA+0x108	R	This register is read-only and is hard-wired to all 1.	0xFFFFFFFF

31	30	29	28	27	26	25	24
top_address_low<n>							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
top_address_low<n>							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
top_address_low<n>				Reserved			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:12]	**top_address_low<n>**	**Controls the region top address bits[31:12] of region *<n>*** For Region 0, this field is read-only and all bits are set HIGH.
[11:0]	**Reserved**	**0xFFF** Read only and points to the highest byte of a 4 Kbytes-aligned area.

REGION_TOP_HIGH_0 Register (REGION_TOP_HIGH_0)

Register	Offset	R/W	Description	Reset Value
TZC0_REGION_TOP_HIGH_0	TZC0_BA+0x10C	R	This register is read-only and is hard-wired to all zeros.	0x00000000
TZC2_REGION_TOP_HIGH_0	TZC2_BA+0x10C	R	This register is read-only and is hard-wired to all zeros.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:0]	**Reserved**	Reserved.

REGION ATTRIBUTES 0 Register (REGION_ATTRIBUTES_0)

Register	Offset	R/W	Description	Reset Value
TZC0_REGION_ATTRIBUTES_0	TZC0_BA+0x110	R/W	Controls the permissions for Region 0.	0x00000000
TZC2_REGION_ATTRIBUTES_0	TZC2_BA+0x110	R/W	Controls the permissions for Region 0.	0x00000000

31	30	29	28	27	26	25	24
s_wr_en	s_rd_en	Reserved					
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				filter_en			

Bits	Description	
[31]	**s_wr_en**	**Secure global write enable This control bit defines the permissions for the Secure writes in the region** Set this bit as follows: 0 = Secure write to the region is not permitted 1 = Permits Secure write to the region.
[30]	**s_rd_en**	**Secure global read enable This control bit defines the permissions for the Secure reads in the region** Set this bit as follows: 0 = Secure read to the region is not permitted 1 = Permits Secure read to the region.
[29:4]	**Reserved**	Reserved.
[3:0]	**filter_en**	Independent region enable for each filter unit Each bit is associated with a filter unit Set these bits as follows: Bit 0 For Filter 0 Bit 1 For Filter 1 Bit 2 For Filter 2 Bit 3 For Filter 3. When set HIGH, it enables the use of the current region programming for the associated filter For Region 0, all bits are set HIGH and cannot be modified.

REGION ID ACCESS 0 Register (REGION_ID_ACCESS_0)

Register	Offset	R/W	Description	Reset Value
TZC0_REGION_ID_ACCESS_0	TZC0_BA+0x114	R/W	Controls the Non-secure access based on the NSAID inputs for Region 0.	0x00000001
TZC2_REGION_ID_ACCESS_0	TZC2_BA+0x114	R/W	Controls the Non-secure access based on the NSAID inputs for Region 0.	0x00000001

31	30	29	28	27	26	25	24
nsaid_wr_en							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
nsaid_wr_en							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
nsaid_rd_en							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
nsaid_rd_en							

Bits	Description	
[31:16]	**nsaid_wr_en**	**NSAID write enables** This enables NSAID inputs to define Non-secure write access permission Each bit of the nsaid_wr_en register field is associated with a value on the NSAIDW<x> signal as follows: Bit 16 Associated with NSAIDW<x> = 0. Bit 17 Associated with NSAIDW<x> = 1. Bit 18 Associated with NSAIDW<x> = 2. Bit 31 Associated with NSAIDW<x> = 15. <x> is the filter unit number. When any bit in this field is set to 1, TZC-400 permits Non-secure write access to any access with the associated NSAIDW<x> value. For example, if NSAIDW<x> is 3 and the corresponding nsaid_wr_en[3] bit is set HIGH then TZC-400 permits the Non-secure write access to progress.
[15:0]	**nsaid_rd_en**	**NSAID read enables** This enables NSAID inputs to define Non-secure read access permission Each bit of the nsaid_rd_en register field is associated with a value on the NSAIDR<x> signal as follows: Bit 0 Associated with NSAIDR<x> = 0. Bit 1 Associated with NSAIDR<x> = 1. Bit 2 Associated with NSAIDR<x> = 2. Bit 15 Associated with NSAIDR<x> = 15. <x> is the filter unit number. When any bit in this field is set HIGH, TZC-400 permits Non-secure read access to any access with the associated NSAIDR<x> value. For example, if NSAIDR<x> is 3 and the corresponding nsaid_rd_en[3] bit is set HIGH then TZC-400 permits the Non-secure read access to progress.

REGION BASE LOW <n> Register (REGION_BASE_LOW_<n>)

Register	Offset	R/W	Description	Reset Value
TZC0_REGION_BASE_LOW_n **n=1,2..8**	TZC0_BA+0x100+(0x20*n)	R/W	Controls the base address bits[31:12] of Region n.	0x00000000
TZC2_REGION_BASE_LOW_n **n=1,2..8**	TZC2_BA+0x100+(0x20*n)	R/W	Controls the base address bits[31:12] of Region n.	0x00000000

31	30	29	28	27	26	25	24
base_address_low<n>							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
base_address_low<n>							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
base_address_low<n>				**Reserved**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:12]	**base_address_low<n>**	**Controls the base address bits[31:12] of Region <n>** For Region 0, this field is read-only The TZC-400 sets the base address of Region 0 to 0x0 The TZC-400 only permits a region to start at a 4 Kbytes aligned address and address bits[11:0] are zeros.
[11:0]	**Reserved**	Reserved.

REGION BASE HIGH <n> Register (REGION_BASE_HIGH_<n>)

Register	Offset	R/W	Description	Reset Value
TZC0_REGION_BASE_HIGH_n n=1,2..8	TZC0_BA+0x104+(0x20*n)	R/W	Reserved	0x00000000
TZC2_REGION_BASE_HIGH_n n=1,2..8	TZC2_BA+0x104+(0x20*n)	R/W	Reserved	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:0]	**Reserved**	Reserved.

REGION TOP LOW <n> Register (REGION_TOP_LOW_<n>)

Register	Offset	R/W	Description	Reset Value
TZC0_REGION_TOP_LOW_n **n=1,2..8**	TZC0_BA+0x108+(0x20*n)	R/W	Controls the region top address bits[31:12] of Region <n>. Region top address bits[11:0] are always 0xFFF because the TZC-400 does not permit the region size to be less than 4 Kbytes. Therefore, the region top address points to the top byte of a 4 Kbytes region.	0x00000FFF
TZC2_REGION_TOP_LOW_n **n=1,2..8**	TZC2_BA+0x108+(0x20*n)	R/W	Controls the region top address bits[31:12] of Region <n>. Region top address bits[11:0] are always 0xFFF because the TZC-400 does not permit the region size to be less than 4 Kbytes. Therefore, the region top address points to the top byte of a 4 Kbytes region.	0x00000FFF

31	30	29	28	27	26	25	24
top_address_low<n>							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
top_address_low<n>							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
top_address_low<n>				**Reserved**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:12]	**top_address_low<n>**	**Controls the region top address bits[31:12] of region <n>** This address points to the start of the next 4 Kbytes aligned address immediately outside the region.
[11:0]	**Reserved**	**0xFFF** Read only and points to the highest byte of a 4 Kbytes-aligned area.

REGION TOP HIGH <n> Register (REGION_TOP_HIGH_<n>)

Register	Offset	R/W	Description	Reset Value
TZC0_REGION_TOP_HIGH_n **n=1,2..8**	TZC0_BA+0x10C+(0x20*n)	R/W	Reserved	0x00000000
TZC2_REGION_TOP_HIGH_n **n=1,2..8**	TZC2_BA+0x10C+(0x20*n)	R/W	Reserved	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:0]	**Reserved**	Reserved.

REGION ATTRIBUTES <n> Register (REGION_ATTRIBUTES_<n>)

Register	Offset	R/W	Description	Reset Value
TZC0_REGION_ATTRIBUTES_n n=1,2..8	TZC0_BA+0x110+(0x20*n)	R/W	Controls the permissions and target filter region enables.	0x00000000
TZC2_REGION_ATTRIBUTES_n n=1,2..8	TZC2_BA+0x110+(0x20*n)	R/W	Controls the permissions and target filter region enables.	0x00000000

31	30	29	28	27	26	25	24
s_wr_en	s_rd_en	Reserved					
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				filter_en			

Bits	Description	
[31]	**s_wr_en**	**Secure global write enable This control bit defines the permissions for the Secure writes in the region** Set this bit as follows: 0 = Secure write to the region is not permitted 1 = Permits Secure write to the region.
[30]	**s_rd_en**	**Secure global read enable This control bit defines the permissions for the Secure reads in the region** Set this bit as follows: 0 = Secure read to the region is not permitted 1 = Permits Secure read to the region.
[29:4]	**Reserved**	**Reserved.**
[3:0]	**filter_en**	**Independent region enable for each filter unit Each bit is associated with a filter unit Set these bits as follows:** Bit 0 = For Filter 0 Bit 1 = For Filter 1 Bit 2 = For Filter 2 Bit 3 = For Filter 3. When set HIGH, it enables the use of the current region programming for the associated filter For Region 0, all bits are set HIGH and cannot be modified.

REGION ID ACCESS <n> Register (REGION_ID_ACCESS_<n>)

Register	Offset	R/W	Description	Reset Value
TZC0_REGION_ID_ACCESS_n **n=1,2..8**	TZC0_BA+0x114+(0x20*n)	R/W	Controls the Non-secure access based on the NSAID inputs.	0x00000000
TZC2_REGION_ID_ACCESS_n **n=1,2..8**	TZC2_BA+0x114+(0x20*n)	R/W	Controls the Non-secure access based on the NSAID inputs.	0x00000000

31	30	29	28	27	26	25	24
nsaid_wr_en							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
nsaid_wr_en							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
nsaid_rd_en							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
nsaid_rd_en							

Bits	Description	
[31:16]	**nsaid_wr_en**	**NSAID write enables** This enables NSAID inputs to define Non-secure write access permission Each bit of the nsaid_wr_en register field is associated with a value on the NSAIDW<x> signal as follows: Bit 16 Associated with NSAIDW<x> = 0. Bit 17 Associated with NSAIDW<x> = 1. Bit 18 Associated with NSAIDW<x> = 2. Bit 31 Associated with NSAIDW<x> = 15. <x> is the filter unit number. When any bit in this field is set to 1, TZC-400 permits Non-secure write access to any access with the associated NSAIDW<x> value. For example, if NSAIDW<x> is 3 and the corresponding nsaid_wr_en[3] bit is set HIGH then TZC-400 permits the Non-secure write access to progress.
[15:0]	**nsaid_rd_en**	**NSAID read enables** This enables NSAID inputs to define Non-secure read access permission Each bit of the nsaid_rd_en register field is associated with a value on the NSAIDR<x> signal as follows: Bit 0 Associated with NSAIDR<x> = 0. Bit 1 Associated with NSAIDR<x> = 1. Bit 2 Associated with NSAIDR<x> = 2. Bit 15 Associated with NSAIDR<x> = 15. <x> is the filter unit number. When any bit in this field is set HIGH, TZC-400 permits Non-secure read access to any access with the associated NSAIDR<x> value. For example, if NSAIDR<x> is 3 and the corresponding nsaid_rd_en[3] bit is set HIGH then TZC-400 permits the Non-secure read access to progress.

PID4 Register (PID4)

Register	Offset	R/W	Description	Reset Value
TZC0_PID4	TZC0_BA+0xFD0	R	Provides the following information about the peripheral configuration • 4 Kbytes_count. • Jep106_c_code.	0x04
TZC2_PID4	TZC2_BA+0xFD0	R	Provides the following information about the peripheral configuration • 4 Kbytes_count. • Jep106_c_code.	0x04

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
4 Kbytes_count				**jep106_c_code**			

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:4]	**4 Kbytes_count**	The number of 4 Kbytes address blocks required to access the registers, expressed in powers of 2 These bits read back as 0x0 This means that the TZC-400 occupies a single 4 Kbytes address block.
[3:0]	**jep106_c_code**	The JEP106 continuation code value represents how many 0x7F continuation characters occur in the manufacturer identity code These bits read back as 0x4 For information on the JEP106 standard, see the Additional reading section.

PID0 Register (PID0)

Register	Offset	R/W	Description	Reset Value
TZC0_PID0	TZC0_BA+0xFE0	R	The PID0 register provides the following information about the peripheral configuration: • part_number_0.	0x60
TZC2_PID0	TZC2_BA+0xFE0	R	The PID0 register provides the following information about the peripheral configuration: • part_number_0.	0x60

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
part_number_0							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**part_number_0**	These bits read back as 0x60

PID1 Register (PID1)

Register	Offset	R/W	Description	Reset Value
TZC0_PID1	TZC0_BA+0xFE4	R	The PID1 register provides the following information about the peripheral configuration: • part_number_1. • Jep106_id_3_0.	0xB4
TZC2_PID1	TZC2_BA+0xFE4	R	The PID1 register provides the following information about the peripheral configuration: • part_number_1. • Jep106_id_3_0.	0xB4

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
jep106_id_3_0				**part_number_1**			

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:4]	**jep106_id_3_0**	**JEP106 identity code [3:0]** See the JEP106, Standard Manufacturer Identification Code These bits read back as 0xB because ARM is the peripheral designer.
[3:0]	**part_number_1**	These bits read back as 0x4.

PID2 Register (PID2)

Register	Offset	R/W	Description	Reset Value
TZC0_PID2	TZC0_BA+0xFE8	R	The PID2 register provides the following information about the peripheral configuration: • Jep106_id_6_4. • Revision number. • JEDEC use flag.	0x2B
TZC2_PID2	TZC2_BA+0xFE8	R	The PID2 register provides the following information about the peripheral configuration: • Jep106_id_6_4. • Revision number. • JEDEC use flag.	0x2B

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
revision				jedec_used	jep106_id_6_4		

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:4]	**revision**	Identifies the revision of the TZC-400 For revision r0p1, this field is set to 0x2.
[3]	**jedec_used**	This indicates that the TZC-400 uses a manufacturer identity code that was allocated by JEDEC according to JEP106 This bit always reads back as 0x1.
[2:0]	**jep106_id_6_4**	**JEP106 identity code [6:4]** See the JEP106, Standard Manufacturer Identification Code These bits read back as 0b011 because ARM is the peripheral designer.

PID3 Register (PID3)

Register	Offset	R/W	Description	Reset Value
TZC0_PID3	TZC0_BA+0xFEC	R	The PID3 register provides the following information about the peripheral configuration: • Mod Number. • RevAnd.	0x00
TZC2_PID3	TZC2_BA+0xFEC	R	The PID3 register provides the following information about the peripheral configuration: • Mod Number. • RevAnd.	0x00

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RevAnd				**mod_number**			

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:4]	**RevAnd**	The top-level RTL provides a 4-bit input, USERPID3REVAND, that is normally tied LOW and provides a read value of 0x0 When silicon is available, and if metal fixes are required, the manufacturer can modify the tie-offs to indicate a revision of the silicon.
[3:0]	**mod_number**	This is set to 0x0.

CID0 Register (CID0)

Register	Offset	R/W	Description	Reset Value
TZC0_CID0	TZC0_BA+0xFF0	R	This is one of four 8-bit registers that together hold a 32-bit component ID value.	0x0D
TZC2_CID0	TZC2_BA+0xFF0	R	This is one of four 8-bit registers that together hold a 32-bit component ID value.	0x0D

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
comp_id_0							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**comp_id_0**	These bits read back as 0x0D

CID1 Register (CID1)

Register	Offset	R/W	Description	Reset Value
TZC0_CID1	TZC0_BA+0xFF4	R	This is one of four 8-bit registers, that together hold a 32-bit component ID value.	0xF0
TZC2_CID1	TZC2_BA+0xFF4	R	This is one of four 8-bit registers, that together hold a 32-bit component ID value.	0xF0

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
comp_id_0							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**comp_id_0**	These bits read back as 0xF0

CID2 Register (CID2)

Register	Offset	R/W	Description	Reset Value
TZC0_CID2	TZC0_BA+0xFF8	R	This is one of four 8-bit registers, that together hold a 32-bit component ID value.	0x05
TZC2_CID2	TZC2_BA+0xFF8	R	This is one of four 8-bit registers, that together hold a 32-bit component ID value.	0x05

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
comp_id_2							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**comp_id_2**	These bits read back as 0x05

CID3 Register (CID3)

Register	Offset	R/W	Description	Reset Value
TZC0_CID3	TZC0_BA+0xFFC	R	This is one of four 8-bit registers that together hold a 32-bit component ID value.	0xB1
TZC2_CID3	TZC2_BA+0xFFC	R	This is one of four 8-bit registers that together hold a 32-bit component ID value.	0xB1

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
comp_id_3							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**comp_id_3**	These bits read back as 0xB1

6.8 True Random Number Generator (TRNG)

6.8.1 Overview

The TRNG NIST SP800-90C is NIST SP800-90A/B/c and BSI AIS 20/31 compliant True Random Number Generator (TRNG). It generates random numbers that are intended to be statistically equivalent to a uniformly distributed random data stream. The circuit includes a NIST SP800-90B compliant noise source, a NIST SP800-90B vetted conditioning component, and a NIST SP800-90A approved Deterministic Random Bit Generator (DRBG). The noise source sends Independent and Identically Distributed (IID) noise stream to the conditioning component to produce full-entropy seed which is then fed into the DRBG to generate random numbers. A health test block is included to perform Known Answer Tests (KAT) and seven different statistical tests required by NIST SP800-90A/B/c and BSI AIS 20/31. The health test block is capable of performing start-up, on-demand, and continuous tests.

The TRNG NIST can generate random seeds from the internal ring-oscillator based noise source or can be manually seeded through a host-provided nonce. Host-provided nonce are fed into the conditioning component to increase the entropy rate using a NIST SP800-90B vetted conditioning function. If the host's nonce has enough entropy, the nonce can also be directly loaded into the DRBG to be used as a seed

6.8.2 Features

- Background noise collection to speed reseeding operations
- Internal random seeding operation
- 128-bit random number generation
- Start-up, continuous and on-demand health tests
- Compliant with NIST SP800-90A/B/c and BSI AIS 20/31
- 128-bit or 256-bit of security strength
- Ring oscillator-based Bit Generator blocks with wide system clock rate dynamic range
- Independent ring oscillator-based Bit Generator blocks

6.9 Hardware Semaphore (HWSEM)

6.9.1 Overview

The Hardware Semaphore block provides 8 hardware semaphores, which can be used to synchronize between different processors with different semaphore keys.

6.9.2 Features

- Supports 8 hardware semaphores for synchronization control between processors
- Interrupt notification for semaphore unlock
- Semaphore is locked by an 8-bit key
- Only the core locking a semaphore can unlock the semaphore with the correct key

6.9.3 Block Diagram

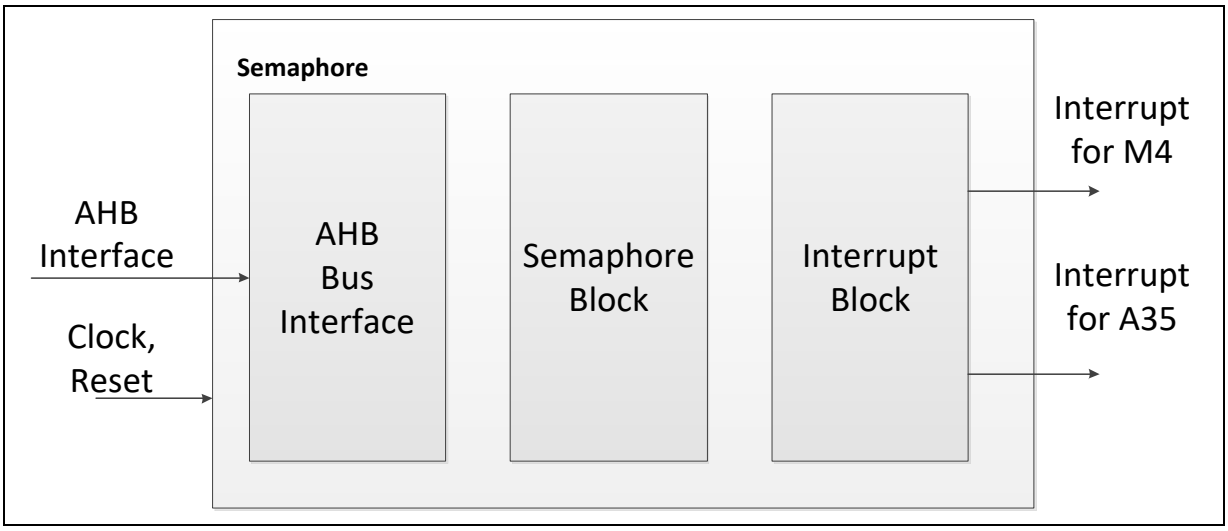


Figure 6.9-1 Semaphore Block Diagram

6.9.4 Basic Configuration

- Clock Source Configuration
  - Enable SEM peripheral clock in HWSCKEN (CLK_SYSCLK1[6]).
- Reset Configuration (Controllable by Cortex-A only)
  - Reset SEM controller in HWSEMRST (SYS_IPRST0[18]).

6.9.5 Functional Description

6.9.5.1 Lock Semaphore

To lock a semaphore, an 8-bit key has to be written into KEY(HWSEM_SEMx[15:8]) field. The steps for locking semaphore x are listed below.

1. Write 8-bit key into KEY.
2. Read SEM_SEMx. If this semaphore is successfully locked by this processor, then key field will be written into KEY, and ID (HWSEM_SEMx[3:0]) shows the processor ID.
3. Otherwise, set SEMxIEN bit and wait for the semaphore unlock event interrupt.

Note: It is not recommended to enter Power-down mode with semaphore locked.

6.9.5.2 Unlock Semaphore

The semaphore can be unlocked by the processor that holds the semaphore using the correct key. To release the semaphore, write the 8-bit key into KEY(HWSEM_SEMx[15:8]). At that time, the key field will be cleared to 0. The semaphore will not be unlocked if the key is incorrect or the processor writing the key does not hold this semaphore.

6.9.5.3 Interrupt Control

Interrupt can be used to notify semaphore unlock event. So software does not need to polling for the semaphore state. Each semaphore has its own interrupt enable bit and interrupt flag.

6.9.5.4 Synchronization between Cortex-A and Cortex-M

User can synchronize Cortex-A and Cortex- M using the semaphore (different master ID and different key).

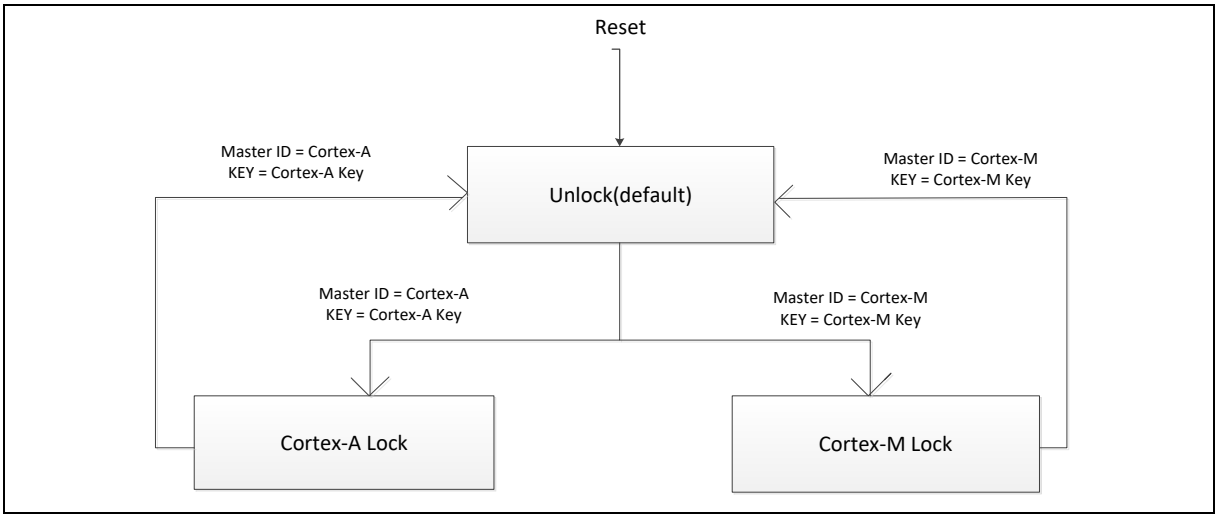


Figure 6.9-2 Synchronization between Cortex-A and Cortex-M Diagram

6.9.5.5 Synchronization between Cortex-A Cores in AMP System

User can synchronize Cortex-A cores in AMP system using the semaphore (different key).

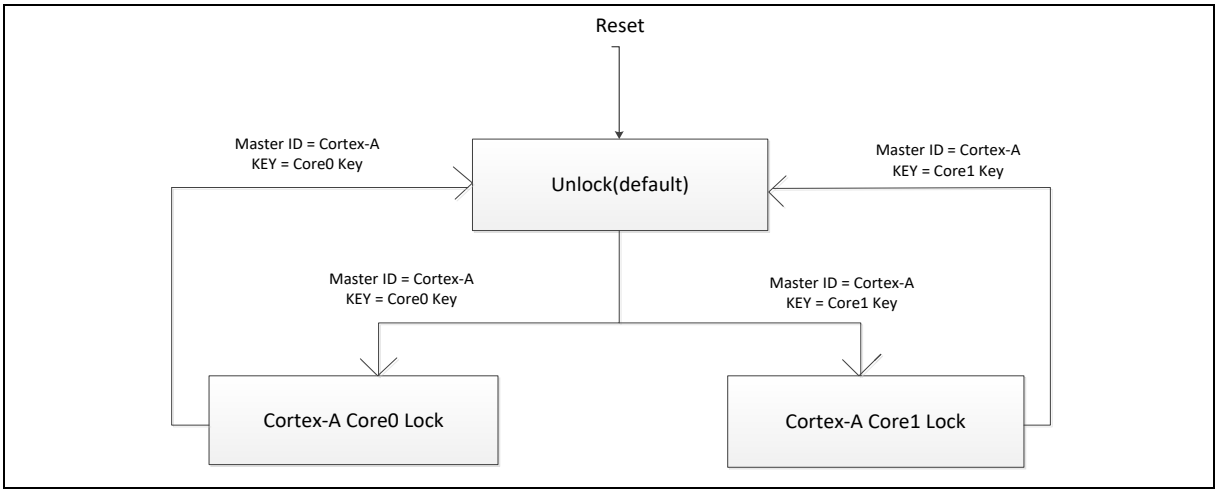


Figure 6.9-3 Synchronization between Cortex-A Cores in AMP System Diagram

6.9.5.6 Reset Semaphore (Cortex-A Only)

The Cortex-A can write SEMxRST bits in HWSEM_CTL register to reset semaphore x. The Cortex-A should only use this feature to unlock the semaphore if Cortex-M reset with semaphore held. A semaphore reset event will also trigger semaphore unlock interrupt if the semaphore is unlocked by reset event.

6.9.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
HWSEM Base Address: **HWSEM_BA = 0x40380000**				
HWSEM_CTL	HWSEM_BA+0x000	R/W	HWSEM Control Register	0x0000_0000
HWSEM_INTENA35	HWSEM_BA+0x004	R/W	HWSEM Interrupt Enable Register (A35 Can Read/Write ,M4 Can Only Read)	0x0000_0000
HWSEM_INTENM4	HWSEM_BA+0x008	R/W	HWSEM Interrupt Enable Register (M4 Can Read/Write , A35 Can Only Read)	0x0000_0000
HWSEM_INTSTSA35	HWSEM_BA+0x00C	R/W	HWSEM Interrupt Status Register (A35 Can Read/Write ,M4 Can Only Read) (Write 1 to Clear)	0x0000_0000
HWSEM_INTSTSM4	HWSEM_BA+0x010	R/W	HWSEM Interrupt Status Register (M4 Can Read/Write , A35 Can Only Read) (Write 1 to Clear)	0x0000_0000
HWSEM_SEM0	HWSEM_BA+0x020	R/W	HWSEM Semaphore 0 Register	0x0000_0000
HWSEM_SEM1	HWSEM_BA+0x024	R/W	HWSEM Semaphore 1 Register	0x0000_0000
HWSEM_SEM2	HWSEM_BA+0x028	R/W	HWSEM Semaphore 2 Register	0x0000_0000
HWSEM_SEM3	HWSEM_BA+0x02C	R/W	HWSEM Semaphore 3 Register	0x0000_0000
HWSEM_SEM4	HWSEM_BA+0x030	R/W	HWSEM Semaphore 4 Register	0x0000_0000
HWSEM_SEM5	HWSEM_BA+0x034	R/W	HWSEM Semaphore 5 Register	0x0000_0000
HWSEM_SEM6	HWSEM_BA+0x038	R/W	HWSEM Semaphore 6 Register	0x0000_0000
HWSEM_SEM7	HWSEM_BA+0x03C	R/W	HWSEM Semaphore 7 Register	0x0000_0000

6.9.7 Register Description

SEM Control Register (HWSEM_CTL)

Register	Offset	R/W	Description	Reset Value
HWSEM_CTL	HWSEM_BA+0x000	R/W	HWSEM Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SEM7RST	**SEM6RST**	**SEM5RST**	**SEM4RST**	**SEM3RST**	**SEM2RST**	**SEM1RST**	**SEM0RST**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**SEM7RST**	**Reset Semaphore 7** Set this bit to reset semaphore 7. After reset, SEM_SEM7 will reset to 0. This bit automatically clears to 0 after reset complete.
[6]	**SEM6RST**	**Reset Semaphore 6** Set this bit to reset semaphore 6. After reset, SEM_SEM6 will reset to 0. This bit automatically clears to 0 after reset complete.
[5]	**SEM5RST**	**Reset Semaphore 5** Set this bit to reset semaphore 5. After reset, SEM_SEM5 will reset to 0. This bit automatically clears to 0 after reset complete.
[4]	**SEM4RST**	**Reset Semaphore 4** Set this bit to reset semaphore 4. After reset, SEM_SEM4 will reset to 0. This bit automatically clears to 0 after reset complete.
[3]	**SEM3RST**	**Reset Semaphore 3** Set this bit to reset semaphore 3. After reset, SEM_SEM3 will reset to 0. This bit automatically clears to 0 after reset complete.
[2]	**SEM2RST**	**Reset Semaphore 2** Set this bit to reset semaphore 2. After reset, SEM_SEM2 will reset to 0. This bit automatically clears to 0 after reset complete.
[1]	**SEM1RST**	**Reset Semaphore 1** Set this bit to reset semaphore 1. After reset, SEM_SEM1 will reset to 0. This bit automatically clears to 0 after reset complete.
[0]	**SEM0RST**	**Reset Semaphore 0** Set this bit to reset semaphore 0. After reset, SEM_SEM0 will reset to 0. This bit automatically clears to 0 after reset complete.

HWSEM Interrupt Enable Register (HWSEM_INTENA35)

Register	Offset	R/W	Description	Reset Value
HWSEM_INTENA35	HWSEM_BA+0x004	R/W	HWSEM Interrupt Enable Register (A35 Can Read/Write ,M4 Can Only Read)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SEM7IEN	**SEM6IEN**	**SEM5IEN**	**SEM4IEN**	**SEM3IEN**	**SEM2IEN**	**SEM1IEN**	**SEM0IEN**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**SEM7IEN**	**Semaphore 7 Unlock Interrupt Enable Bit** 0 = Semaphore 7 unlock interrupt Disabled. 1 = Semaphore 7 unlock interrupt Enabled.
[6]	**SEM6IEN**	**Semaphore 6 Unlock Interrupt Enable Bit** 0 = Semaphore 6 unlock interrupt Disabled. 1 = Semaphore 6 unlock interrupt Enabled.
[5]	**SEM5IEN**	**Semaphore 5 Unlock Interrupt Enable Bit** 0 = Semaphore 5 unlock interrupt Disabled. 1 = Semaphore 5 unlock interrupt Enabled.
[4]	**SEM4IEN**	**Semaphore 4 Unlock Interrupt Enable Bit** 0 = Semaphore 4 unlock interrupt Disabled. 1 = Semaphore 4 unlock interrupt Enabled.
[3]	**SEM3IEN**	**Semaphore 3 Unlock Interrupt Enable Bit** 0 = Semaphore 3 unlock interrupt Disabled. 1 = Semaphore 3 unlock interrupt Enabled.
[2]	**SEM2IEN**	**Semaphore 2 Unlock Interrupt Enable Bit** 0 = Semaphore 2 unlock interrupt Disabled. 1 = Semaphore 2 unlock interrupt Enabled.
[1]	**SEM1IEN**	**Semaphore 1 Unlock Interrupt Enable Bit** 0 = Semaphore 1 unlock interrupt Disabled. 1 = Semaphore 1 unlock interrupt Enabled.
[0]	**SEM0IEN**	**Semaphore 0 Unlock Interrupt Enable Bit** 0 = Semaphore 0 unlock interrupt Disabled. 1 = Semaphore 0 unlock interrupt Enabled.

HWSEM Interrupt Enable Register (HWSEM_INTENM4)

Register	Offset	R/W	Description	Reset Value
HWSEM_INTENM4	HWSEM_BA+0x008	R/W	HWSEM Interrupt Enable Register (M4 Can Read/Write , A35 Can Only Read)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SEM7IEN	**SEM6IEN**	**SEM5IEN**	**SEM4IEN**	**SEM3IEN**	**SEM2IEN**	**SEM1IEN**	**SEM0IEN**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**SEM7IEN**	**Semaphore 7 Unlock Interrupt Enable Bit** 0 = Semaphore 7 unlock interrupt Disabled. 1 = Semaphore 7 unlock interrupt Enabled.
[6]	**SEM6IEN**	**Semaphore 6 Unlock Interrupt Enable Bit** 0 = Semaphore 6 unlock interrupt Disabled. 1 = Semaphore 6 unlock interrupt Enabled.
[5]	**SEM5IEN**	**Semaphore 5 Unlock Interrupt Enable Bit** 0 = Semaphore 5 unlock interrupt Disabled. 1 = Semaphore 5 unlock interrupt Enabled.
[4]	**SEM4IEN**	**Semaphore 4 Unlock Interrupt Enable Bit** 0 = Semaphore 4 unlock interrupt Disabled. 1 = Semaphore 4 unlock interrupt Enabled.
[3]	**SEM3IEN**	**Semaphore 3 Unlock Interrupt Enable Bit** 0 = Semaphore 3 unlock interrupt Disabled. 1 = Semaphore 3 unlock interrupt Enabled.
[2]	**SEM2IEN**	**Semaphore 2 Unlock Interrupt Enable Bit** 0 = Semaphore 2 unlock interrupt Disabled. 1 = Semaphore 2 unlock interrupt Enabled.
[1]	**SEM1IEN**	**Semaphore 1 Unlock Interrupt Enable Bit** 0 = Semaphore 1 unlock interrupt Disabled. 1 = Semaphore 1 unlock interrupt Enabled.
[0]	**SEM0IEN**	**Semaphore 0 Unlock Interrupt Enable Bit** 0 = Semaphore 0 unlock interrupt Disabled. 1 = Semaphore 0 unlock interrupt Enabled.

HWSEM Interrupt Status Register (HWSEM_INTSTSA35)

Register	Offset	R/W	Description	Reset Value
HWSEM_INTSTSA35	HWSEM_BA+0x00C	R/W	HWSEM Interrupt Status Register (A35 Can Read/Write ,M4 Can Only Read) (Write 1 to Clear)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
SEM7IF	SEM6IF	SEM5IF	SEM4IF	SEM3IF	SEM2IF	SEM1IF	SEM0IF

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**SEM7IF**	**Semaphore 7 Unlock Interrupt Flag** 0 = No semaphore 7 unlock interrupt generated. 1 = Semaphore 7 unlock interrupt generated. **Note:** Write 1 to clear this bit to 0.
[6]	**SEM6IF**	**Semaphore 6 Unlock Interrupt Flag** 0 = No semaphore 6 unlock interrupt generated. 1 = Semaphore 6 unlock interrupt generated. **Note:** Write 1 to clear this bit to 0.
[5]	**SEM5IF**	**Semaphore 5 Unlock Interrupt Flag** 0 = No semaphore 5 unlock interrupt generated. 1 = Semaphore 5 unlock interrupt generated. **Note:** Write 1 to clear this bit to 0.
[4]	**SEM4IF**	**Semaphore 4 Unlock Interrupt Flag** 0 = No semaphore 4 unlock interrupt generated. 1 = Semaphore 4 unlock interrupt generated. **Note:** Write 1 to clear this bit to 0.
[3]	**SEM3IF**	**Semaphore 3 Unlock Interrupt Flag** 0 = No semaphore 3 unlock interrupt generated. 1 = Semaphore 3 unlock interrupt generated. **Note:** Write 1 to clear this bit to 0.
[2]	**SEM2IF**	**Semaphore 2 Unlock Interrupt Flag** 0 = No semaphore 2 unlock interrupt generated. 1 = Semaphore 2 unlock interrupt generated. **Note:** Write 1 to clear this bit to 0.

[1]	**SEM1IF**	**Semaphore 1 Unlock Interrupt Flag** 0 = No semaphore 1 unlock interrupt generated. 1 = Semaphore 1 unlock interrupt generated. **Note:** Write 1 to clear this bit to 0.
[0]	**SEM0IF**	**Semaphore 0 Unlock Interrupt Flag** 0 = No semaphore 0 unlock interrupt generated. 1 = Semaphore 0 unlock interrupt generated. **Note:** Write 1 to clear this bit to 0.

HWSEM Interrupt Status Register (HWSEM_INTSTSM4)

Register	Offset	R/W	Description	Reset Value
HWSEM_INTSTSM4	HWSEM_BA+0x010	R/W	HWSEM Interrupt Status Register (M4 Can Read/Write , A35 Can Only Read) (Write 1 to Clear)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SEM7IF	**SEM6IF**	**SEM5IF**	**SEM4IF**	**SEM3IF**	**SEM2IF**	**SEM1IF**	**SEM0IF**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**SEM7IF**	**Semaphore 7 Unlock Interrupt Flag** 0 = No semaphore 7 unlock interrupt generated. 1 = Semaphore 7 unlock interrupt generated. **Note:** Write 1 to clear this bit to 0.
[6]	**SEM6IF**	**Semaphore 6 Unlock Interrupt Flag** 0 = No semaphore 6 unlock interrupt generated. 1 = Semaphore 6 unlock interrupt generated. **Note:** Write 1 to clear this bit to 0.
[5]	**SEM5IF**	**Semaphore 5 Unlock Interrupt Flag** 0 = No semaphore 5 unlock interrupt generated. 1 = Semaphore 5 unlock interrupt generated. **Note:** Write 1 to clear this bit to 0.
[4]	**SEM4IF**	**Semaphore 4 Unlock Interrupt Flag** 0 = No semaphore 4 unlock interrupt generated. 1 = Semaphore 4 unlock interrupt generated. **Note:** Write 1 to clear this bit to 0.
[3]	**SEM3IF**	**Semaphore 3 Unlock Interrupt Flag** 0 = No semaphore 3 unlock interrupt generated. 1 = Semaphore 3 unlock interrupt generated. **Note:** Write 1 to clear this bit to 0.
[2]	**SEM2IF**	**Semaphore 2 Unlock Interrupt Flag** 0 = No semaphore 2 unlock interrupt generated. 1 = Semaphore 2 unlock interrupt generated. **Note:** Write 1 to clear this bit to 0.

[1]	**SEM1IF**	**Semaphore 1 Unlock Interrupt Flag** 0 = No semaphore 1 unlock interrupt generated. 1 = Semaphore 1 unlock interrupt generated. **Note:** Write 1 to clear this bit to 0.
[0]	**SEM0IF**	**Semaphore 0 Unlock Interrupt Flag** 0 = No semaphore 0 unlock interrupt generated. 1 = Semaphore 0 unlock interrupt generated. **Note:** Write 1 to clear this bit to 0.

HWSEM Semaphore x Register (HWSEM_SEMx)

Register	Offset	R/W	Description	Reset Value
HWSEM_SEM0	HWSEM_BA+0x020	R/W	HWSEM Semaphore 0 Register	0x0000_0000
HWSEM_SEM1	HWSEM_BA+0x024	R/W	HWSEM Semaphore 1 Register	0x0000_0000
HWSEM_SEM2	HWSEM_BA+0x028	R/W	HWSEM Semaphore 2 Register	0x0000_0000
HWSEM_SEM3	HWSEM_BA+0x02C	R/W	HWSEM Semaphore 3 Register	0x0000_0000
HWSEM_SEM4	HWSEM_BA+0x030	R/W	HWSEM Semaphore 4 Register	0x0000_0000
HWSEM_SEM5	HWSEM_BA+0x034	R/W	HWSEM Semaphore 5 Register	0x0000_0000
HWSEM_SEM6	HWSEM_BA+0x038	R/W	HWSEM Semaphore 6 Register	0x0000_0000
HWSEM_SEM7	HWSEM_BA+0x03C	R/W	HWSEM Semaphore 7 Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
KEY							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				ID			

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:8]	**KEY**	**Key for Lock/Unlock Semaphore** This field holds the key to lock/unlock this semaphore.
[7:4]	**Reserved**	Reserved.
[3:0]	**ID**	**Processor ID (Read Only)** This field holds the ID of Processor which lock this semaphore. 0000 = Semaphore is unlocked. 0001 = This semaphore is locked by Cortex-A. 0010 = This semaphore is locked by Cortex-M. Others = Reserved.

6.10 Wormhole Controller (WHC)

6.10.1 Overview

This chip is equipped with two Wormhole Controllers (WHC) that facilitates inter-processor communication between Cortex-A35 and Cortex-M4, and between Cortex-A35 and TSI.

The WHC utilizes an interactive mechanism for each processor to deliver and receive data with eight unidirectional message channels. Each side can send messages and inform counterpart to read these messages. After the receiver has read these messages, the sender will be acknowledged and then be able to send next messages. Moreover, a recall function is implemented for sender to call back the sent messages.

6.10.2 Features

- One interrupt notifies reset event or power state change event to other side
- One register indicates the status of counterpart, including:
  - Power state (power down, power off)
  - Reset source
- Four general event interrupts on each side
- Eight unidirectional message channels, four for each direction. Each channel transfer 16-byte data.
- Supports message recall by Tx side.

6.10.3 Block Diagram

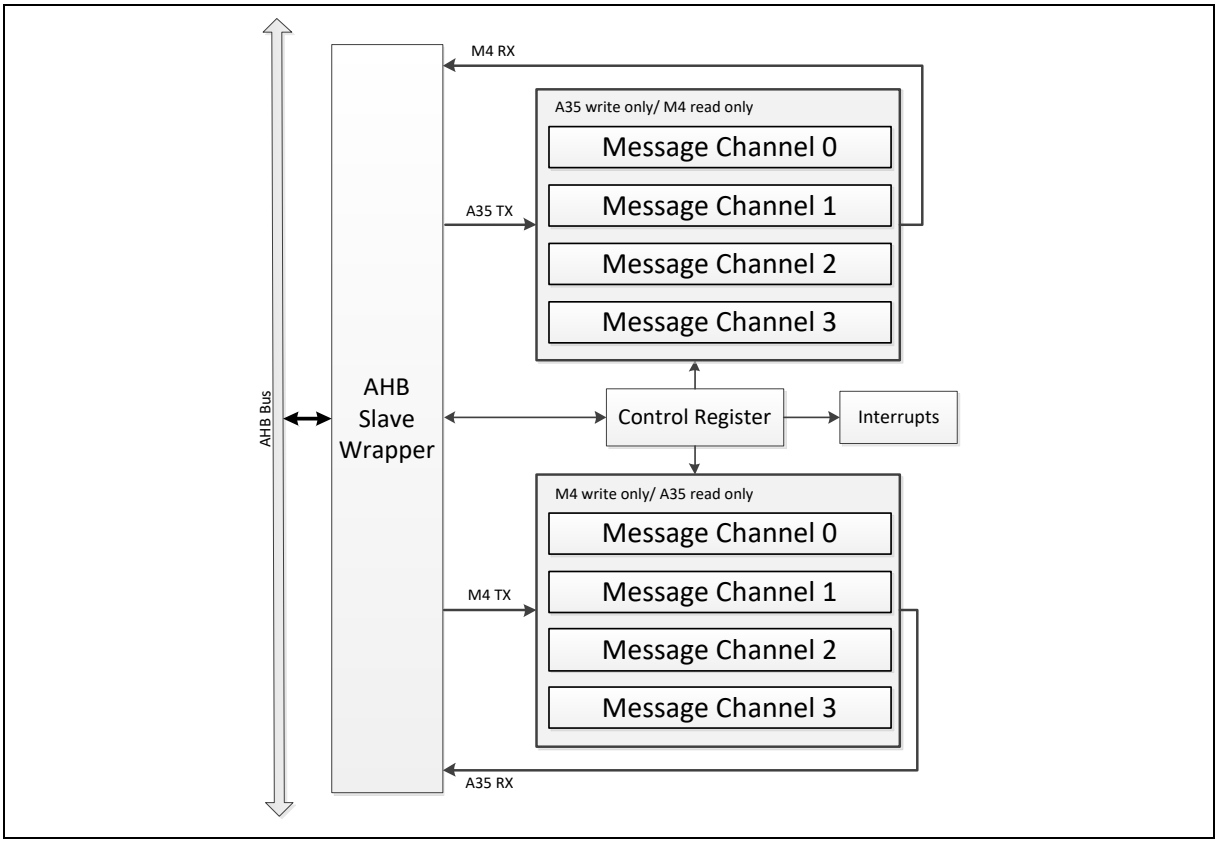


Figure 6.10-1 Wormhole Controller Block Diagram

6.10.4 Basic Configuration

- Clock Source Configuration
  - Enable WHC peripheral clock in WH0CKEN, WH1CKEN (CLK_SYSCLK1[4], CLK_SYSCLK1[5]).
- Reset Configuration (Controllable by Cortex-A only)
  - Reset WHC controller in WRHO0RST, WRHO1RST (SYS_IPRST0[14], SYS_IPRST0[15]).

6.10.5 Functional Description

6.10.5.1 Counterpart Status

The status register WHCx_CPSTS is provided to show the current power state and reset source of other CPU. Any status change can also be used to trigger interrupt or wakeup event if relative enable bits are set in WHCx_WKCTL and WHCx_WKCTL register.

6.10.5.2 General Interrupt

Four general interrupts are provided on each direction. Sender can set TRGGIx (WHCx_GINTTRG[3:0]) to trigger an interrupt on receiver side to indicate a predefined status or event that doesn't need a message body.

6.10.5.3 Message Channel

Eight unidirectional message channels are provided. Each channel provides 16-byte message body with one toggle bit to indicate the current owner. The following lists a complete flow of CA35 sending a message to CM4 through channel 0:

1. CA35 checks if CH0RDY (WHCx_TXSTS[0]) is 1 to make sure channel 0 is ready.
2. CA35 fills the message body to WHCx_TM0DATy, y = 0~3 registers.
3. CA35 sets CH0SND (WHCx_TXCTL[0]) to send the message.
4. CM4 checks CH0RDY (WHCx_RXSTS[0]) is 1 to make sure new message arrived.
5. CM4 reads the mesasge body from WHCx_RM0DATy, y = 0~3 registers.
6. CA35 sets CH0ACK (WHCx_RXCTL[0]) to indicate the message is received.

6.10.5.4 Message Recall

The message channel supports recall function. Sender can recall message by writing 1 to CHxRC (WHCx_TXCTL[19:16]). Sender can write a new message to Tx channel after recall is completed. Once a message is recalled, the CHxRDY (WHCx_TXCTL[3:0]) will be set to 1.

This feature is designed to rebuild communication if one of the controllers is reset by WDT or other signal while the other side is still operating normally. It is not recommended to use this function during normal operation.

6.10.5.5 Interrupt and Wakeup Event

All functions mentioned in previous sections supports interrupt function and wake-up event. All interrupt enable control bits could be found in WHCx_INTEN register, and wake-up event enable bits could be found in WHCx_WKCTL register. No matter the interrupt enable bit is cleared or set in WHCx_INTEN register, the corresponding status bit in WHCx_INTSTS register can always indicate the current interrupt status.

6.10.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
WHC Base Address: **WHCx_BA = 0x403A_0000 + (0x1001_0000 * x)** **x=0, 1** **WHC0 is used for the communication between A35 and M4 (RTP). (TSI writes ignored and reads as zero.)** **WHC1 is used for the communication between A35 (TZS) and M4 (TSI). (TZNS, RTP write ignored and read as zero.)**				
WHCx_WKCTL	WHCx_BA+0x000	R/W	WHC Wakeup Event Control Register	0x0000_0000
WHCx_INTEN	WHCx_BA+0x004	R/W	WHC Interrupt Enable Register	0x0000_0000
WHCx_INTSTS	WHCx_BA+0x008	R/W	WHC Interrupt Status Register	0x0000_0000
WHCx_CPSTS	WHCx_BA+0x040	R/W	WHC Counterpart Status Register	0xXX00_0XXX
WHCx_GINTTRG	WHCx_BA+0x080	W	WHC General Interrupt Trigger Register	0x0000_0000
WHCx_TXCTL	WHCx_BA+0x0C0	W	WHC TX Message Control Register	0x0000_0000
WHCx_TXSTS	WHCx_BA+0x0C4	R	WHC TX Message Status Register	0x0000_000F
WHCx_RXCTL	WHCx_BA+0x0C8	W	WHC RX Message Control Register	0x0000_0000
WHCx_RXSTS	WHCx_BA+0x0CC	R	WHC RX Message Status Register	0x0000_0000
WHCx_TM0DAT0	WHCx_BA+0x100	W	WHC TX Message Channel 0 Data 0 Register	0xXXXX_XXXX
WHCx_TM0DAT1	WHCx_BA+0x104	W	WHC TX Message Channel 0 Data 1 Register	0xXXXX_XXXX
WHCx_TM0DAT2	WHCx_BA+0x108	W	WHC TX Message Channel 0 Data 2 Register	0xXXXX_XXXX
WHCx_TM0DAT3	WHCx_BA+0x10C	W	WHC TX Message Channel 0 Data 3 Register	0xXXXX_XXXX
WHCx_TM1DAT0	WHCx_BA+0x110	W	WHC TX Message Channel 1 Data 0 Register	0xXXXX_XXXX
WHCx_TM1DAT1	WHCx_BA+0x114	W	WHC TX Message Channel 1 Data 1 Register	0xXXXX_XXXX
WHCx_TM1DAT2	WHCx_BA+0x118	W	WHC TX Message Channel 1 Data 2 Register	0xXXXX_XXXX
WHCx_TM1DAT3	WHCx_BA+0x11C	W	WHC TX Message Channel 1 Data 3 Register	0xXXXX_XXXX
WHCx_TM2DAT0	WHCx_BA+0x120	W	WHC TX Message Channel 2 Data 0 Register	0xXXXX_XXXX
WHCx_TM2DAT1	WHCx_BA+0x124	W	WHC TX Message Channel 2 Data 1 Register	0xXXXX_XXXX
WHCx_TM2DAT2	WHCx_BA+0x128	W	WHC TX Message Channel 2 Data 2 Register	0xXXXX_XXXX
WHCx_TM2DAT3	WHCx_BA+0x12C	W	WHC TX Message Channel 2 Data 3 Register	0xXXXX_XXXX
WHCx_TM3DAT0	WHCx_BA+0x130	W	WHC TX Message Channel 3 Data 0 Register	0xXXXX_XXXX
WHCx_TM3DAT1	WHCx_BA+0x134	W	WHC TX Message Channel 3 Data 1 Register	0xXXXX_XXXX
WHCx_TM3DAT2	WHCx_BA+0x138	W	WHC TX Message Channel 3 Data 2 Register	0xXXXX_XXXX
WHCx_TM3DAT3	WHCx_BA+0x13C	W	WHC TX Message Channel 3 Data 3 Register	0xXXXX_XXXX

WHCx_RM0DAT0	WHCx_BA+0x200	R	WHC RX Message Channel 0 Data 0 Register	0xXXXX_XXXX
WHCx_RM0DAT1	WHCx_BA+0x204	R	WHC RX Message Channel 0 Data 1 Register	0xXXXX_XXXX
WHCx_RM0DAT2	WHCx_BA+0x208	R	WHC RX Message Channel 0 Data 2 Register	0xXXXX_XXXX
WHCx_RM0DAT3	WHCx_BA+0x20C	R	WHC RX Message Channel 0 Data 3 Register	0xXXXX_XXXX
WHCx_RM1DAT0	WHCx_BA+0x210	R	WHC RX Message Channel 1 Data 0 Register	0xXXXX_XXXX
WHCx_RM1DAT1	WHCx_BA+0x214	R	WHC RX Message Channel 1 Data 1 Register	0xXXXX_XXXX
WHCx_RM1DAT2	WHCx_BA+0x218	R	WHC RX Message Channel 1 Data 2 Register	0xXXXX_XXXX
WHCx_RM1DAT3	WHCx_BA+0x21C	R	WHC RX Message Channel 1 Data 3 Register	0xXXXX_XXXX
WHCx_RM2DAT0	WHCx_BA+0x220	R	WHC RX Message Channel 2 Data 0 Register	0xXXXX_XXXX
WHCx_RM2DAT1	WHCx_BA+0x224	R	WHC RX Message Channel 2 Data 1 Register	0xXXXX_XXXX
WHCx_RM2DAT2	WHCx_BA+0x228	R	WHC RX Message Channel 2 Data 2 Register	0xXXXX_XXXX
WHCx_RM2DAT3	WHCx_BA+0x22C	R	WHC RX Message Channel 2 Data 3 Register	0xXXXX_XXXX
WHCx_RM3DAT0	WHCx_BA+0x230	R	WHC RX Message Channel 3 Data 0 Register	0xXXXX_XXXX
WHCx_RM3DAT1	WHCx_BA+0x234	R	WHC RX Message Channel 3 Data 1 Register	0xXXXX_XXXX
WHCx_RM3DAT2	WHCx_BA+0x238	R	WHC RX Message Channel 3 Data 2 Register	0xXXXX_XXXX
WHCx_RM3DAT3	WHCx_BA+0x23C	R	WHC RX Message Channel 3 Data 3 Register	0xXXXX_XXXX

6.10.7 Register Description

WHC Wakeup Event Control Register (WHCx_WKCTL)

Register	Offset	R/W	Description	Reset Value
WHCx_WKCTL	WHCx_BA+0x000	R/W	WHC Wakeup Event Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				RX3WKEN	RX2WKEN	RX1IWKEN	RX0WKEN
23	22	21	20	19	18	17	16
Reserved				TX3WKEN	TX2WKEN	TX1IWKEN	TX0WKEN
15	14	13	12	11	10	9	8
Reserved				GI3WKEN	GI2WKEN	GI1WKEN	GI0WKEN
7	6	5	4	3	2	1	0
Reserved		PD1WKEN	POFF1WKEN	RST1WKEN	PD0WKEN	POFF0WKEN	RST0WKEN

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27]	**RX3WKEN**	**RX Message Channel 3 Interrupt Wake-up Enable Bit** RX message channel 3 interrupt wake-up trigger event enable. 0 = Channel 3 interrupt wake-up trigger event Disabled. 1 = Channel 3 interrupt wake-up trigger event Enabled.
[26]	**RX2WKEN**	**RX Message Channel 2 Interrupt Wake-up Enable Bit** RX message Channel 2 interrupt wake-up trigger event enable. 0 = Channel 2 interrupt wake-up trigger event Disabled. 1 = Channel 2 interrupt wake-up trigger event Enabled.
[25]	**RX1WKEN**	**RX Message Channel 1 Interrupt Wake-up Enable Bit** RX message channel 1 interrupt wake-up trigger event enable. 0 = Channel 1 interrupt wake-up trigger event Disabled. 1 = Channel 1 interrupt wake-up trigger event Enabled.
[24]	**RX0WKEN**	**RX Message Channel 0 Interrupt Wake-up Enable Bit** RX message channel 0 interrupt wake-up trigger event enable. 0 = Channel 0 interrupt wake-up trigger event Disabled. 1 = Channel 0 interrupt wake-up trigger event Enabled.
[23:20]	**Reserved**	Reserved.
[19]	**TX3WKEN**	**TX Message Channel 3 Interrupt Wake-up Enable Bit** TX message channel 3 interrupt wake-up trigger event enable. 0 = Channel 3 interrupt wake-up trigger event Disabled. 1 = Channel 3 interrupt wake-up trigger event Enabled.

[18]	**TX2WKEN**	**TX Message Channel 2 Interrupt Wake-up Enable Bit** TX message channel 2 interrupt wake-up trigger event enable. 0 = Channel 2 interrupt wake-up trigger event Disabled. 1 = Channel 2 interrupt wake-up trigger event Enabled.
[17]	**TX1WKEN**	**TX Message Channel 1 Interrupt Wake-up Enable Bit** TX message channel 1 interrupt wake-up trigger event enable. 0 = Channel 1 interrupt wake-up trigger event Disabled. 1 = Channel 1 interrupt wake-up trigger event Enabled.
[16]	**TX0WKEN**	**TX Message Channel 0 Interrupt Wake-up Enable Bit** TX message channel 0 interrupt wake-up trigger event enable. 0 = Channel 0 interrupt wake-up trigger event Disabled. 1 = Channel 0 interrupt wake-up trigger event Enabled.
[15:12]	**Reserved**	Reserved.
[11]	**GI3WKEN**	**General Interrupt 3 Wake-up Enable** If this bit is set to 1, while general interrupt flag is set to 1 and interrupt enable bit is enabled, the general interrupt signal will generate a wake-up trigger event to chip. 0 = Wake-up trigger event disabled if general interrupt 3 signal generated. 1 = Wake-up trigger event enabled if general interrupt 3 signal generated.
[10]	**GI2WKEN**	**General Interrupt 2 Wake-up Enable** If this bit is set to 1, while general interrupt flag is set to 1 and interrupt enable bit is enabled, the general interrupt signal will generate a wake-up trigger event to chip. 0 = Wake-up trigger event disabled if general interrupt 2 signal generated. 1 = Wake-up trigger event enabled if general interrupt 2 signal generated.
[9]	**GI1WKEN**	**General Interrupt 1 Wake-up Enable** If this bit is set to 1, while general interrupt flag is set to 1 and interrupt enable bit is enabled, the general interrupt signal will generate a wake-up trigger event to chip. 0 = Wake-up trigger event disabled if general interrupt 1 signal generated. 1 = Wake-up trigger event enabled if general interrupt 1 signal generated.
[8]	**GI0WKEN**	**General Interrupt 0 Wake-up Enable** If this bit is set to 1, while general interrupt flag is set to 1 and interrupt enable bit is enabled, the general interrupt signal will generate a wake-up trigger event to chip. 0 = Wake-up trigger event disabled if general interrupt 0 signal generated. 1 = Wake-up trigger event enabled if general interrupt 0 signal generated.
[7:6]	**Reserved**	Reserved.
[5]	**PD1WKEN**	**A35 Second Core Power Down State Interrupt Wake-up Enable Bit** Counterpart power down interrupt wake-up trigger event enable. 0 = Counterpart power down interrupt wake-up trigger event Disabled. 1 = Counterpart power down interrupt wake-up trigger event Enabled. **Note:** This bit that indicates the second A35 core is **only available for Cortex-M4**.
[4]	**POFF1WKEN**	**A35 Second Core Power Off State Interrupt Wake-up Enable Bit** Counterpart power off interrupt wake-up trigger event enable. 0 = Counterpart power off interrupt wake-up trigger event Disabled. 1 = Counterpart power off interrupt wake-up trigger event Enabled. **Note:** This bit that indicates the second A35 core is **only available for Cortex-M4**.

[3]	**RST1WKEN**	**A35 Second Core Reset Interrupt Wake-up Enable Bit** Counterpart reset event interrupt wake-up trigger event enable. 0 = Counterpart reset event interrupt wake-up trigger event Disabled. 1 = Counterpart reset event interrupt wake-up trigger event Enabled. **Note:** This bit that indicates the second A35 core is **only available for Cortex-M4**.
[2]	**PD0WKEN**	**Counterpart First Core Power Down State Interrupt Wake-up Enable Bit** Counterpart power down interrupt wake-up trigger event enable. 0 = Counterpart power down interrupt wake-up trigger event Disabled. 1 = Counterpart power down interrupt wake-up trigger event Enabled.
[1]	**POFF0WKEN**	**Counterpart First Core Power Off State Interrupt Wake-up Enable Bit** Counterpart power off interrupt wake-up trigger event enable. 0 = Counterpart power off interrupt wake-up trigger event Disabled. 1 = Counterpart power off interrupt wake-up trigger event Enabled.
[0]	**RST0WKEN**	**Counterpart First Core Reset Interrupt Wake-up Enable Bit** Counterpart reset event interrupt wake-up trigger event enable. 0 = Counterpart reset event interrupt wake-up trigger event Disabled. 1 = Counterpart reset event interrupt wake-up trigger event Enabled.

WHC Interrupt Enable Register (WHCx_INTEN)

Register	Offset	R/W	Description	Reset Value
WHCx_INTEN	WHCx_BA+0x004	R/W	WHC Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				RX3IEN	RX2IEN	RX1IEN	RX0IEN
23	22	21	20	19	18	17	16
Reserved				TX3IEN	TX2IEN	TX1IEN	TX0IEN
15	14	13	12	11	10	9	8
Reserved				GI3IEN	GI2IEN	GI1IEN	GI0IEN
7	6	5	4	3	2	1	0
Reserved		PD1IEN	POFF1IEN	RST1IEN	PD0IEN	POFF0IEN	RST0IEN

Bits	Description	
[31:24]	**Reserved**	Reserved.
[27]	**RX3IEN**	**RX Message Channel 3 Interrupt Enable Bit** RX message channel 3 arrive or recall interrupt enable 0 = Channel 3 arrive or recall interrupt Disabled. 1 = Channel 3 arrive or recall interrupt Enabled.
[26]	**RX2IEN**	**RX Message Channel 2 Interrupt Enable Bit** RX message channel 2 arrive or recall interrupt enable. 0 = Channel 2 arrive or recall interrupt Disabled. 1 = Channel 2 arrive or recall interrupt Enabled.
[25]	**RX1IEN**	**RX Message Channel 1 Interrupt Enable Bit** RX message channel 1 arrive or recall interrupt enable. 0 = Channel 1 arrive or recall interrupt Disabled. 1 = Channel 1 arrive or recall interrupt Enabled.
[24]	**RX0IEN**	**RX Message Channel 0 Interrupt Enable Bit** RX message channel 0 arrive or recall interrupt enable 0 = Channel 0 arrive or recall interrupt Disabled. 1 = Channel 0 arrive or recall interrupt Enabled.
[23:20]	**Reserved**	Reserved.
[19]	**TX3IEN**	**TX Message Channel 3 Interrupt Enable Bit** TX message channel 3 ACK interrupt enable 0 = Channel 3 ACK interrupt Disabled. 1 = Channel 3 ACK interrupt Enabled.
[18]	**TX2IEN**	**TX Message Channel 2 Interrupt Enable Bit** TX message channel 2 ACK interrupt enable. 0 = Channel 2 ACK interrupt Disabled. 1 = Channel 2 ACK interrupt Enabled.

[17]	**TX1IEN**	**TX Message Channel 1 Interrupt Enable Bit** TX message channel 1 ACK interrupt enable. 0 = Channel 1 ACK interrupt Disabled. 1 = Channel 1 ACK interrupt Enabled.
[16]	**TX0IEN**	**TX Message Channel 0 Interrupt Enable Bit** TX message channel 0 ACK interrupt enable 0 = Channel 0 ACK interrupt Disabled. 1 = Channel 0 ACK interrupt Enabled.
[15:12]	**Reserved**	Reserved.
[11]	**GI3IEN**	**General Interrupt 3 Enable Bit** General interrupt 3 interrupt enable. 0 = General interrupt 3 interrupt Disabled. 1 = General interrupt 3 interrupt Enabled.
[10]	**GI2IEN**	**General Interrupt 2 Enable Bit** General interrupt 2 interrupt enable. 0 = General interrupt 2 interrupt Disabled. 1 = General interrupt 2 interrupt Enabled.
[9]	**GI1IEN**	**General Interrupt 1 Enable Bit** General interrupt 1 interrupt enable. 0 = General interrupt 1 interrupt Disabled. 1 = General interrupt 1 interrupt Enabled.
[8]	**GI0IEN**	**General Interrupt 0 Enable Bit** General interrupt 0 interrupt enable. 0 = General interrupt 0 interrupt Disabled. 1 = General interrupt 0 interrupt Enabled.
[7:6]	**Reserved**	Reserved.
[5]	**PD1IEN**	**A35 Second Core Enter/Exit Power-down mode Interrupt Enable Bit** Counterpart Enter/Exit Power-down mode Interrupt Enable 0 = Counterpart power down status change interrupt Disabled. 1 = Counterpart power down status change interrupt Enabled. **Note:** This bit that indicates the second A35 core is **only available for Cortex-M4**.
[4]	**POFF1IEN**	**A35 Second Core Enter/Exit Power Off Mode Interrupt Enable Bit** Counterpart Enter/Exit Power Off Mode Interrupt Enable 0 = Counterpart power Off status change interrupt Disabled. 1 = Counterpart power Off status change interrupt Enabled. **Note:** This bit that indicates the second A35 core is **only available for Cortex-M4**.
[3]	**RST1IEN**	**A35 Second Core Reset Event Interrupt Enable Bit** Counterpart reset event interrupt enable. 0 = Counterpart status change interrupt Disabled. 1 = Counterpart status change interrupt Enabled. **Note:** This bit that indicates the second A35 core is **only available for Cortex-M4**.
[2]	**PD0IEN**	**Counterpart First Core Enter/Exit Power-down mode Interrupt Enable Bit** Counterpart Enter/Exit Power-down mode Interrupt Enable 0 = Counterpart power down status change interrupt Disabled. 1 = Counterpart power down status change interrupt Enabled.

[1]	**POFF0IEN**	**Counterpart First Core Enter/Exit Power Off Mode Interrupt Enable Bit** Counterpart Enter/Exit Power Off Mode Interrupt Enable 0 = Counterpart power Off status change interrupt Disabled. 1 = Counterpart power Off status change interrupt Enabled.
[0]	**RST0IEN**	**Counterpart First Core Reset Event Interrupt Enable Bit** Counterpart reset event interrupt enable. 0 = Counterpart status change interrupt Disabled. 1 = Counterpart status change interrupt Enabled.

WHC Interrupt Status Register (WHCx_INTSTS)

Register	Offset	R/W	Description	Reset Value
WHCx_INTSTS	WHCx_BA+0x008	R/W	WHC Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				RX3IF	RX2IF	RX1IF	RX0IF
23	22	21	20	19	18	17	16
Reserved				TX3IF	TX2IF	TX1IF	TX0IF
15	14	13	12	11	10	9	8
Reserved				GI3IF	GI2IF	GI1IF	GI0IF
7	6	5	4	3	2	1	0
Reserved		PD1IF	POFF1IF	RST1IF	PD0IF	POFF0IF	RST0IF

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27]	**RX3IF**	**RX Message Channel 3 Interrupt Flag** RX message channel 3 arrive or recall interrupt flag. 0 = No channel 3 arrive or recall interrupt generated. 1 = Channel 3 arrive or recall interrupt generated. **Note:** Write 1 to clear this bit to 0.
[26]	**RX2IF**	**RX Message Channel 2 Interrupt Flag** Short message channel 2 arrive or recall interrupt flag. 0 = No channel 2 arrive or recall interrupt generated. 1 = Channel 2 arrive or recall interrupt generated. **Note:** Write 1 to clear this bit to 0.
[25]	**RX1IF**	**RX Message Channel 1 Interrupt Flag** Short message channel 1 arrive or recall interrupt flag. 0 = No channel 1 arrive or recall interrupt generated. 1 = Channel 1 arrive or recall interrupt generated. **Note:** Write 1 to clear this bit to 0.
[24]	**RX0IF**	**RX Message Channel 0 Interrupt Flag** Short message channel 0 arrive or recall interrupt flag. 0 = No channel 0 arrive or recall interrupt generated. 1 = Channel 0 arrive or recall interrupt generated. **Note:** Write 1 to clear this bit to 0.
[23:20]	**Reserved**	Reserved.

[19]	**TX3IF**	**TX Message Channel 3 Interrupt Flag** Tx message channel 3 ACK interrupt flag. 0 = No channel 3 ACK interrupt generated. 1 = Channel 3 ACK interrupt generated. **Note:** Write 1 to clear this bit to 0.
[18]	**TX2IF**	**TX Message Channel 2 Interrupt Flag** Tx message channel 2 ACK interrupt flag. 0 = No channel 2 ACK interrupt generated. 1 = Channel 2 ACK interrupt generated. **Note:** Write 1 to clear this bit to 0.
[17]	**TX1IF**	**TX Message Channel 1 Interrupt Flag** Tx message channel 1 ACK interrupt flag. 0 = No channel 1 ACK interrupt generated. 1 = Channel 1 ACK interrupt generated. **Note:** Write 1 to clear this bit to 0.
[16]	**TX0IF**	**TX Message Channel 0 Interrupt Flag** Tx message channel 0 ACK interrupt flag. 0 = No channel 0 ACK interrupt generated. 1 = Channel 0 ACK interrupt generated. **Note:** Write 1 to clear this bit to 0.
[15:12]	**Reserved**	Reserved.
[11]	**GI3IF**	**General Event 3 Flag** General event 3 interrupt flag. 0 = No general event 3 interrupt generated. 1 = General event 3 interrupt generated. **Note:** Write 1 to clear this bit to 0.
[10]	**GI2IF**	**General Event 2 Flag** General event 2 interrupt flag. 0 = No general event 2 interrupt generated. 1 = General event 2 interrupt generated. **Note:** Write 1 to clear this bit to 0.
[9]	**GI1IF**	**General Event 1 Flag** General event 1 interrupt flag. 0 = No general event 1 interrupt generated. 1 = General event 1 interrupt generated. **Note:** Write 1 to clear this bit to 0.
[8]	**GI0IF**	**General Event 0 Flag** General event 0 interrupt flag. 0 = No general event 0 interrupt generated. 1 = General event 0 interrupt generated. **Note:** Write 1 to clear this bit to 0.
[7:6]	**Reserved**	Reserved.

[5]	**PD1IF**	**A35 Second Core Enter/Exit Power-down mode Interrupt Flag** Counterpart Enter/Exit Power-down mode Interrupt Flag 0 = No counterpart Power-down mode status change interrupt generated. 1 = Counterpart Power-down mode status change interrupt generated. **Note 1:** Write 1 to clear this bit to 0. **Note 2:** This bit that indicates the second A35 core is **only available for Cortex-M4**.
[4]	**POFF1IF**	**A35 Second Core Enter/Exit Power Off Mode Interrupt Flag** Counterpart Enter/Exit Power Off Mode Interrupt Flag 0 = No counterpart power off mode status change interrupt generated. 1 = Counterpart power off mode status change interrupt generated. **Note 1:** Write 1 to clear this bit to 0. **Note 2:** This bit that indicates the second A35 core is **only available for Cortex-M4**.
[3]	**RST1IF**	**A35 Second Core Reset Event Interrupt Flag** Counterpart reset event interrupt flag. 0 = No counterpart reset status change interrupt generated. 1 = Counterpart reset status change interrupt generated. **Note 1:** Write 1 to clear this bit to 0. **Note 2:** This bit that indicates the second A35 core is **only available for Cortex-M4**.
[2]	**PD0IF**	**Counterpart First Core Enter/Exit Power-down mode Interrupt Flag** Counterpart Enter/Exit Power-down mode Interrupt Flag 0 = No counterpart Power-down mode status change interrupt generated. 1 = Counterpart Power-down mode status change interrupt generated. **Note:** Write 1 to clear this bit to 0.
[1]	**POFF0IF**	**Counterpart First Core Enter/Exit Power Off Mode Interrupt Flag** Counterpart Enter/Exit Power Off Mode Interrupt Flag 0 = No counterpart power off mode status change interrupt generated. 1 = Counterpart power off mode status change interrupt generated. **Note:** Write 1 to clear this bit to 0.
[0]	**RST0IF**	**Counterpart First Core Reset Event Interrupt Flag** Counterpart reset event interrupt flag. 0 = No counterpart reset status change interrupt generated. 1 = Counterpart reset status change interrupt generated. **Note:** Write 1 to clear this bit to 0.

WHC Counterpart Status Register (WHCx_CPSTS)

Register	Offset	R/W	Description	Reset Value
WHCx_CPSTS	WHCx_BA+0x040	R/W	WHC Counterpart Status Register	0xXX00_0XXX

31	30	29	28	27	26	25	24
OPMODE1				OPMODE0			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							CPULKRF
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CPURF	Reserved	SYSRF	Reserved		WDTRF	Reserved	

Bits	Description	
[31:28]	**OPMODE1**	**Operating Mode of Other Side Core 1, Representing A35 Core 1 (Read Only)** This bit field indicates the operating mode of the core 1 of the other side. 0000 = Run mode. 0001 = Power off mode. 0010 = Power-down mode. Others = Reserved. **Note:** This field can only be accessed by Cortex-M4.
[27:24]	**OPMODE0**	**Operating Mode of Other Side Core 0, Representing A35/M4 Core 0 (Read Only)** This bit field indicates the operating mode of the core 0 of the other side. 0000 = Run mode. 0001 = Power off mode. 0010 = Power-down mode. Others = Reserved.
[23:9]	**Reserved**	Reserved.
[8]	**CPULKRF**	**CPU Lockup Reset Flag (Read Only)** 0 = No reset from CPU lockup happened. 1 = The Cortex-M4 lockup happened and chip is reset. **Note 1:** Write 1 to clear this bit to 0. **Note 2:** When CPU lockup happened under ICE is connected, this flag will set to 1 but chip will not reset. **Note 3:** From Cortex-M4 side, this bit set to 1 cannot be seen.
[7]	**CPURF**	**CPU Reset Flag (Read Only)** The CPU reset flag is set by hardware if software writes CPURST (SYS_IPRST0[3:1]) 1 to reset counterpart CPU (any of A35 Core or M4 Core). 0 = No reset from CPU. 1 = The counterpart Core is reset by software setting CPURST to 1. **Note:** Write 1 to clear this bit to 0.
[6]	**Reserved**	Reserved.

[5]	**SYSRF**	**System Reset Flag (Read Only)** The system reset flag is set by the “Reset Signal” from the Cortex-M4 core to indicate the previous reset source. 0 = No reset from Cortex-M4. 1 = The Cortex-M4 had issued the reset signal to reset the system by writing 1 to the bit SYSRESETREQ(AIRCR[2], Application Interrupt and Reset Control Register, address = 0xE000ED0C) in system control registers of Cortex-M4 core. **Note 1:** Write 1 to clear this bit to 0. **Note 2:** From Cortex-M4 side, this bit set to 1 cannot be seen.
[4:3]	**Reserved**	Reserved.
[2]	**WDTRF**	**WDT Reset Flag (Read Only)** The WDT reset flag is set by the “Reset Signal” from the Watchdog Timer or Window Watchdog Timer to indicate the previous reset source. 0 = No reset from watchdog timer or window watchdog timer. 1 = The watchdog timer or window watchdog timer had issued the reset signal to reset the system. **Note 1:** Write 1 to clear this bit to 0. **Note 2:** Watchdog Timer register RSTF(WDTx_CTL[2]) bit is set if the system has been reset by WDT time-out reset. Window Watchdog Timer register WWDTRF(WWDTx_STATUS[1]) bit is set if the system has been reset by WWDT time-out reset.
[1:0]	**Reserved**	Reserved.

WHC General Interrupt Trigger Register (WHCx_GINTTRG)

Register	Offset	R/W	Description	Reset Value
WHCx_GINTTRG	WHCx_BA+0x080	W	WHC General Interrupt Trigger Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				TRGGI3	TRGGI2	TRGGI1	TRGGI0

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3]	**TRGGI3**	**Trigger General Interrupt 3** Write this bit 1 to trigger general interrupt 3.
[2]	**TRGGI2**	**Trigger General Interrupt 2** Write this bit 1 to trigger general interrupt 2.
[1]	**TRGGI1**	**Trigger General Interrupt 1** Write this bit 1 to trigger general interrupt 1.
[0]	**TRGGI0**	**Trigger General Interrupt 0** Write this bit 1 to trigger general interrupt 0.

WHC TX Message Channel Control Register (WHCx_TXCTL)

Register	Offset	R/W	Description	Reset Value
WHCx_TXCTL	WHCx_BA+0x0C0	W	WHC TX Message Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				CH3RC	CH2RC	CH1RC	CH0RC
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				CH3SND	CH2SND	CH1SND	CH0SND

Bits	Description	
[31:20]	**Reserved**	Reserved.
[19]	**CH3RC**	**Recall Channel 3 Message** Write 1 to this bit to recall TX channel 3 message. Before write 1 to this bit, sender should check the corresponding CHxRDY bit in TXSTS is 0 or not. **Note:** Write 0 to this bit has no effect. Write 1 to SND and RC bits of the same channel simultaneously will yield unpredictable result.
[18]	**CH2RC**	**Recall Channel 2 Message** Write 1 to this bit to recall TX channel 2 message. Before write 1 to this bit, sender should check the corresponding CHxRDY bit in TXSTS is 0 or not. **Note:** Write 0 to this bit has no effect. Write 1 to SND and RC bits of the same channel simultaneously will yield unpredictable result.
[17]	**CH1RC**	**Recall Channel 1 Message** Write 1 to this bit to recall TX channel 1 message. Before write 1 to this bit, sender should check the corresponding CHxRDY bit in TXSTS is 0 or not. **Note:** Write 0 to this bit has no effect. Write 1 to SND and RC bits of the same channel simultaneously will yield unpredictable result.
[16]	**CH0RC**	**Recall Channel 1 Message** Write 1 to this bit to recall TX channel 1 message. Before write 1 to this bit, sender should check the corresponding CHxRDY bit in TXSTS is 0 or not. **Note:** Write 0 to this bit has no effect. Write 1 to SND and RC bits of the same channel simultaneously will yield unpredictable result.
[15:4]	**Reserved**	Reserved.
[3]	**CH3SND**	**Send Channel 3 Message** Write 1 to this bit to send TX channel 3 message. Before write 1 to this bit, sender should check the corresponding CHxRDY bit in TXSTS is 1 or not. Otherwise, the receiver may get unpredictable data. **Note:** Write 0 to this bit has no effect. Write 1 to SND and RC bits of the same channel simultaneously will yield unpredictable result.

[2]	**CH2SND**	**Send Channel 2 Message** Write 1 to this bit to send TX channel 2 message. Before write 1 to this bit, sender should check the corresponding CHxRDY bit in TXSTS is 1 or not. Otherwise, the receiver may get unpredictable data. **Note:** Write 0 to this bit has no effect. Write 1 to SND and RC bits of the same channel simultaneously will yield unpredictable result.
[1]	**CH1SND**	**Send Channel 1 Message** Write 1 to this bit to send TX channel 1 message. Before write 1 to this bit, sender should check the corresponding CHxRDY bit in TXSTS is 1 or not. Otherwise, the receiver may get unpredictable data. **Note:** Write 0 to this bit has no effect. Write 1 to SND and RC bits of the same channel simultaneously will yield unpredictable result.
[0]	**CH0SND**	**Send Channel 0 Message** Write 1 to this bit to send TX channel 0 message. Before write 1 to this bit, sender should check the corresponding CHxRDY bit in TXSTS is 1 or not. Otherwise, the receiver may get unpredictable data. **Note:** Write 0 to this bit has no effect. Write 1 to SND and RC bits of the same channel simultaneously will yield unpredictable result.

WHC TX Message Status Register (WHCx_TXSTS)

Register	Offset	R/W	Description	Reset Value
WHCx_TXSTS	WHCx_BA+0x0C4	R	WHC TX Message Status Register	0x0000_000F

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**CH3RDY**	**CH2RDY**	**CH1RDY**	**CH0RDY**

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3]	**CH3RDY**	**Channel 3 Message Ready** 0 = No recall event. 1 = Message channel is available to send new data. This bit automatically cleared to 0 after write 1 to CH3SND. Automatically set to 1 after recall complete or receiver notify message read.
[2]	**CH2RDY**	**Channel 2 Message Ready** 0 = No recall event. 1 = Message channel is available to send new data. This bit automatically cleared to 0 after write 1 to CH2SND. Automatically set to 1 after recall complete or receiver notify message read.
[1]	**CH1RDY**	**Channel 1 Message Ready** 0 = No recall event. 1 = Message channel is available to send new data. This bit automatically cleared to 0 after write 1 to CH1SND. Automatically set to 1 after recall complete or receiver notify message read.
[0]	**CH0RDY**	**Channel 0 Message Ready** 0 = No recall event. 1 = Message channel is available to send new data. This bit automatically cleared to 0 after write 1 to CH0SND. Automatically set to 1 after recall complete or receiver notify message read.

WHC RX Message Control Register (WHCx_RXCTL)

Register	Offset	R/W	Description	Reset Value
WHCx_RXCTL	WHCx_BA+0x0C8	W	WHC RX Message Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				CH3ACK	CH2ACK	CH1ACK	CH0ACK

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3]	**CH3ACK**	**RX Message Channel 3 ACK** Write 1 to notify sender that channel 3 message has been read. Write 0 to this bit has no effect. Write 1 will clear CH3RDY.
[2]	**CH2ACK**	**RX Message Channel 2 ACK** Write 1 to notify sender that channel 2 message has been read. Write 0 to this bit has no effect. Write 1 will clear CH2RDY.
[1]	**CH1ACK**	**RX Message Channel 1 ACK** Write 1 to notify sender that channel 1 message has been read. Write 0 to this bit has no effect. Write 1 will clear CH1RDY.
[0]	**CH0ACK**	**RX Message Channel 0 ACK** Write 1 to notify sender that channel 0 message has been read. Write 0 to this bit has no effect. Write 1 will clear CH0RDY.

WHC RX Message Status Register (WHCx_RXSTS)

Register	Offset	R/W	Description	Reset Value
WHCx_RXSTS	WHCx_BA+0x0CC	R	WHC RX Message Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				CH3RDY	CH2RDY	CH1RDY	CH0RDY

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3]	**CH3RDY**	**RX Channel 3 Message Ready** 0 = RX channel 3 message data is not ready. 1 = RX channel 3 message data is ready. **Note:** This bit is automatically set to 1 after sender writes 1 to CH3SND. It is automatically cleared to 0 after recall complete or write 1 to CH3ACK.
[2]	**CH2RDY**	**RX Channel 2 Message Ready** 0 = RX channel 2 message data is not ready. 1 = RX channel 2 message data is ready. **Note:** This bit is automatically set to 1 after sender writes 1 to CH2SND. It is automatically cleared to 0 after recall complete or write 1 to CH2ACK.
[1]	**CH1RDY**	**RX Channel 1 Message Ready** 0 = RX channel 1 message data is not ready. 1 = RX channel 1 message data is ready. **Note:** This bit is automatically set to 1 after sender writes 1 to CH1SND. It is automatically cleared to 0 after recall complete or write 1 to CH1ACK.
[0]	**CH0RDY**	**RX Channel 0 Message Ready** 0 = RX channel 0 message data is not ready. 1 = RX channel 0 message data is ready. **Note:** This bit is automatically set to 1 after sender writes 1 to CH0SND. It is automatically cleared to 0 after recall complete or write 1 to CH0ACK.

WHC TX Message Channel 0-3 Data x Register (WHCx_TMxDATx)

Register	Offset	R/W	Description	Reset Value
WHCx_TM0DAT0	WHCx_BA+0x100	W	WHC TX Message Channel 0 Data 0 Register	0xXXXX_XXXX
WHCx_TM0DAT1	WHCx_BA+0x104	W	WHC TX Message Channel 0 Data 1 Register	0xXXXX_XXXX
WHCx_TM0DAT2	WHCx_BA+0x108	W	WHC TX Message Channel 0 Data 2 Register	0xXXXX_XXXX
WHCx_TM0DAT3	WHCx_BA+0x10C	W	WHC TX Message Channel 0 Data 3 Register	0xXXXX_XXXX
WHCx_TM1DAT0	WHCx_BA+0x110	W	WHC TX Message Channel 1 Data 0 Register	0xXXXX_XXXX
WHCx_TM1DAT1	WHCx_BA+0x114	W	WHC TX Message Channel 1 Data 1 Register	0xXXXX_XXXX
WHCx_TM1DAT2	WHCx_BA+0x118	W	WHC TX Message Channel 1 Data 2 Register	0xXXXX_XXXX
WHCx_TM1DAT3	WHCx_BA+0x11C	W	WHC TX Message Channel 1 Data 3 Register	0xXXXX_XXXX
WHCx_TM2DAT0	WHCx_BA+0x120	W	WHC TX Message Channel 2 Data 0 Register	0xXXXX_XXXX
WHCx_TM2DAT1	WHCx_BA+0x124	W	WHC TX Message Channel 2 Data 1 Register	0xXXXX_XXXX
WHCx_TM2DAT2	WHCx_BA+0x128	W	WHC TX Message Channel 2 Data 2 Register	0xXXXX_XXXX
WHCx_TM2DAT3	WHCx_BA+0x12C	W	WHC TX Message Channel 2 Data 3 Register	0xXXXX_XXXX
WHCx_TM3DAT0	WHCx_BA+0x130	W	WHC TX Message Channel 3 Data 0 Register	0xXXXX_XXXX
WHCx_TM3DAT1	WHCx_BA+0x134	W	WHC TX Message Channel 3 Data 1 Register	0xXXXX_XXXX
WHCx_TM3DAT2	WHCx_BA+0x138	W	WHC TX Message Channel 3 Data 2 Register	0xXXXX_XXXX
WHCx_TM3DAT3	WHCx_BA+0x13C	W	WHC TX Message Channel 3 Data 3 Register	0xXXXX_XXXX

31	30	29	28	27	26	25	24
DATA							
23	22	21	20	19	18	17	16
DATA							
15	14	13	12	11	10	9	8
DATA							
7	6	5	4	3	2	1	0
DATA							

Bits	Description	
[31:0]	**DATA**	**TX Message Data** This field contains the TX message data field. Sender should only write these registers while ready bit in TXSTS register is 1. Otherwise, the receiver can read unpredictable data.

WHC RX Message Channel 0-3 Data x Register (WHCx_RMxDATx)

Register	Offset	R/W	Description	Reset Value
WHCx_RM0DAT0	WHCx_BA+0x200	R	WHC RX Message Channel 0 Data 0 Register	0xXXXX_XXXX
WHCx_RM0DAT1	WHCx_BA+0x204	R	WHC RX Message Channel 0 Data 1 Register	0xXXXX_XXXX
WHCx_RM0DAT2	WHCx_BA+0x208	R	WHC RX Message Channel 0 Data 2 Register	0xXXXX_XXXX
WHCx_RM0DAT3	WHCx_BA+0x20C	R	WHC RX Message Channel 0 Data 3 Register	0xXXXX_XXXX
WHCx_RM1DAT0	WHCx_BA+0x210	R	WHC RX Message Channel 1 Data 0 Register	0xXXXX_XXXX
WHCx_RM1DAT1	WHCx_BA+0x214	R	WHC RX Message Channel 1 Data 1 Register	0xXXXX_XXXX
WHCx_RM1DAT2	WHCx_BA+0x218	R	WHC RX Message Channel 1 Data 2 Register	0xXXXX_XXXX
WHCx_RM1DAT3	WHCx_BA+0x21C	R	WHC RX Message Channel 1 Data 3 Register	0xXXXX_XXXX
WHCx_RM2DAT0	WHCx_BA+0x220	R	WHC RX Message Channel 2 Data 0 Register	0xXXXX_XXXX
WHCx_RM2DAT1	WHCx_BA+0x224	R	WHC RX Message Channel 2 Data 1 Register	0xXXXX_XXXX
WHCx_RM2DAT2	WHCx_BA+0x228	R	WHC RX Message Channel 2 Data 2 Register	0xXXXX_XXXX
WHCx_RM2DAT3	WHCx_BA+0x22C	R	WHC RX Message Channel 2 Data 3 Register	0xXXXX_XXXX
WHCx_RM3DAT0	WHCx_BA+0x230	R	WHC RX Message Channel 3 Data 0 Register	0xXXXX_XXXX
WHCx_RM3DAT1	WHCx_BA+0x234	R	WHC RX Message Channel 3 Data 1 Register	0xXXXX_XXXX
WHCx_RM3DAT2	WHCx_BA+0x238	R	WHC RX Message Channel 3 Data 2 Register	0xXXXX_XXXX
WHCx_RM3DAT3	WHCx_BA+0x23C	R	WHC RX Message Channel 3 Data 3 Register	0xXXXX_XXXX

31	30	29	28	27	26	25	24
DATA							
23	22	21	20	19	18	17	16
DATA							
15	14	13	12	11	10	9	8
DATA							
7	6	5	4	3	2	1	0
DATA							

Bits	Description	
[31:0]	**DATA**	**RX Message Data** This field contains the RX message data field. Receiver should only read these registers while ready bit in RXSTS register is 1. Otherwise, the receiver can read unpredictable data.

6.11 DDR Memory Controller (UMCTL2)

6.11.1 Overview

Portions Copyright (c) Synopsys®, Inc. All rights reserved. Used with permission.

The uMCTL2 is a DDR memory controller and specific hardware configuration of Synopsys uMCTL2 DDR controller v3.80a.

In this chip, the uMCTL2 only supports the following three SDRAM types, and other SDRAM types are not supported:

- DDR2
- DDR3
- DDR3L

On the host side, the uMCTL2 can accept memory access requests from up to 7 application-side host ports. The configuration registers are programmed through the AMBA 3.0 APB software interface.

6.11.2 Features

- Complete, integrated, single-vendor DDR2, DDR3, DDR3L solution when combined with the Synopsys DWC DDR PHY
- Supports a range of DDR3/DDR3L SDRAM speeds, from DDR3-667/DDR3L-667 through DDR3-1066/DDR3L-1066
- Supports a range of DDR2 SDRAM speeds, from DDR2-667 through DDR2-1066
- For DDR3 configurations, direct software request control or programmable internal control for ZQ short calibration cycles
- For DDR3 configurations, support for ZQ long calibration after self-refresh exit
- Dynamic scheduling to optimize bandwidth and latency
- Read and write buffers in fully associative CAMs, configurable in powers of two
- Delayed writes for optimum performance on SDRAM data bus
- For maximum SDRAM efficiency, commands are executed out-of-order
- Programmable SDRAM parameters
- 16 bits SDRAM data-bus width
- Supported SDRAM Burst Length of 8
- Supports maximum 2 memory ranks
- Not support all DIMM modes
- Control options to avoid starvation of lower priorities
- Guaranteed coherency for write-after-read (WAR) and read-after-write (RAW) hazards
- Write combine to allow multiple writes to the same address to be combined into a single write to SDRAM; supported for same starting address
- Paging policy selectable by configuration registers as any of the following:
  - Leave pages open after accesses, or
  - Close page when there are no further accesses available in the controller for that page, or
  - Auto-precharge with each access, with an optimization for page-close mode which leaves the page open after a flush for read-write and write-read collision cases

- Supports automatic SDRAM power-down entry and exit caused by lack of transaction arrival for a programmable time
- Supports automatic uMCTL2 low power mode operation caused by lack of transaction arrival for a programmable time through the Hardware Low Power Interface
- Supports self-refresh entry and exit as follows:
  - Support for automatic self-refresh entry and exit caused by lack of transaction arrival for a programmable time
  - Support for self-refresh entry and exit under software control
  - Support for self-refresh entry and exit using dedicated DDRC hardware low power interface control (similar to the AMBA 3 AXI protocol low power control interface)
- Support for explicit SDRAM mode register updates under software control
- Flexible address mapper logic to allow application specific mapping of row, column, bank, and rank bits
- Programmable support for 1T or 2T timing
- Selectable refresh control options:
  - Controller-generated auto-refreshes at programmable average intervals
  - In multi-rank designs, an offset can be applied to each rank's refresh timer to allow rank refreshes to expire at different times
  - Ability to group up to 8 controller-generated refreshes together to be issued consecutively
  - When controller-generated refreshes are grouped, some refreshes can be issued speculatively when the controller is idle for a programmable period of time
  - Ability to disable controller-generated auto-refreshes
  - Ability to issue a refresh through direct software request
  - When controller-generated refreshes are grouped, some refreshes can be issued speculatively when the controller is idle does not have any HIF transactions to the SDRAM rank/bank address for a programmable period of time
- Advanced power-saving design includes no unnecessary toggling of command, address, and data pins (RAS/CAS/WE/BA/A hold last state after each command; DQ does not transition on writes when bytes are disabled)
- Leverages out of order requests with CAM to maximize throughput
- APB interface for the uMCTL2 software accessible registers
- 7 host ports using AMBA AXI

6.11.3 Block Diagram

Figure 6.11-1 is a system-level block diagram of the uMCTL2 in a multi-port configuration.

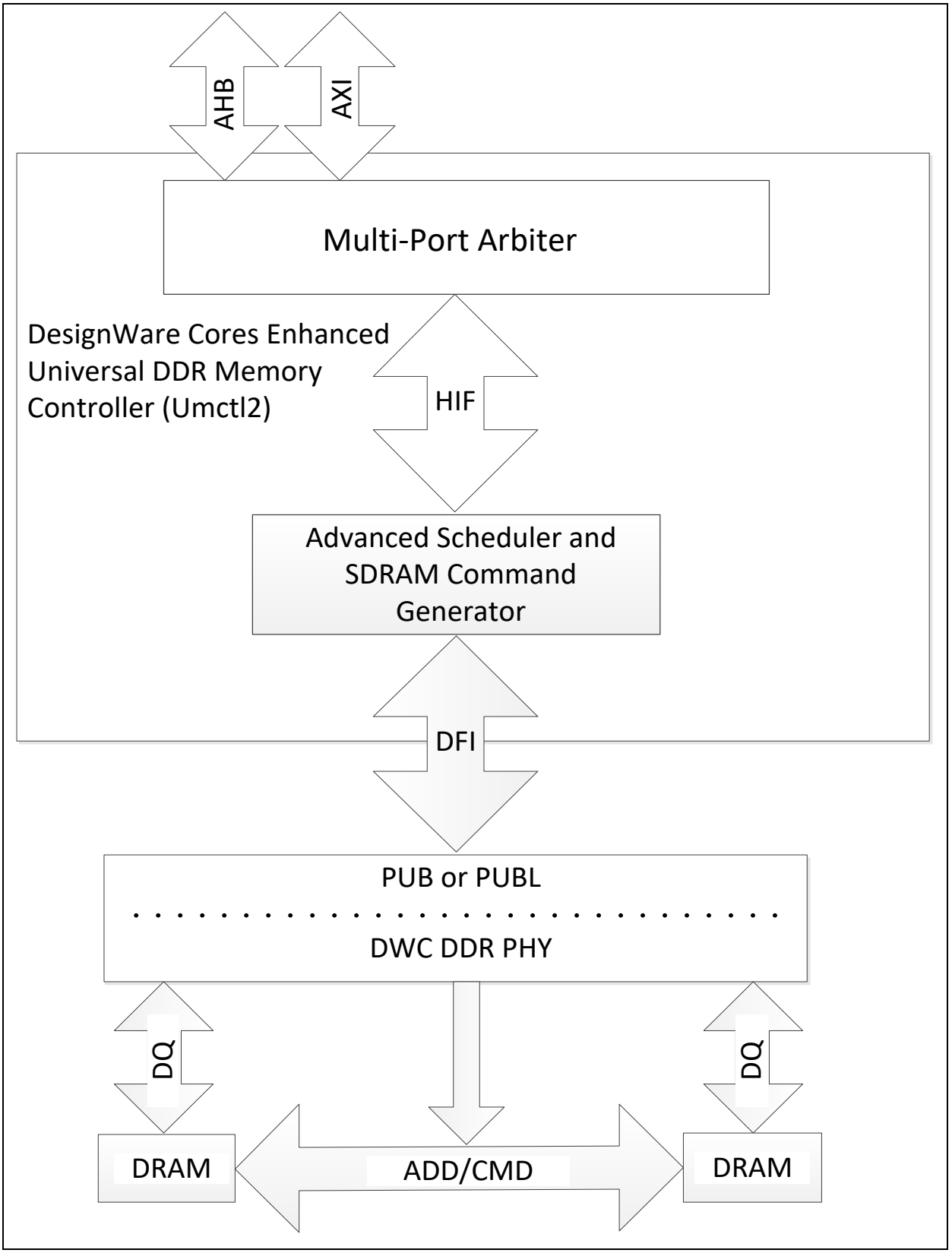


Figure 6.11-1 uMCTL2 (AXI Configuration) in System Level Diagram

The uMCTL2 controller consists of the following main architectural blocks:

- AXI Port Interface (XPI)
- Port Arbiter (PA)
- DDR Controller (DDRC)
- Register Block

The uMCTL2 controller converts system bus transactions into memory commands on the DFI interface that are compliant with the DDR protocols. Figure 6.11-2 is the block diagram of the uMCTL2.

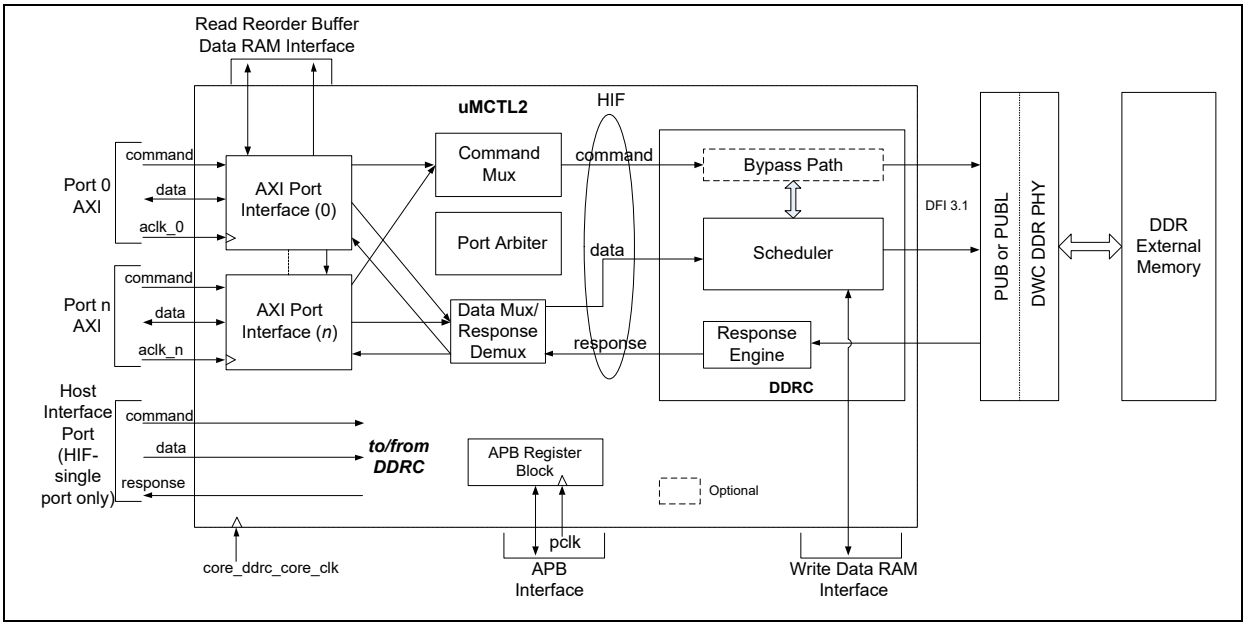


Figure 6.11-2 uMCTL2 Block Diagram (AXI)

- The AXI Port Interface (XPI) block: This block provides the interface to the application ports. It provides bus protocol handling, data buffering and reordering for read data, data bus size conversion (upsizing or downsizing), and memory burst address alignment. Read data is stored in a SRAM, read re-order buffer and returned in order, to the AXI ports. The SRAM may be instantiated as embedded memory external to the uMCTL2 or implemented as flip-flops within the uMCTL2.
- The Port Arbiter (PA) block: This block provides latency sensitive, priority based arbitration between the addresses issued by the XPIs (by the ports).
- The DDR Controller (DDRC) block: This block contains a logical CAM (Content Addressable Memory), which can be synthesized using standard cells. This holds information on the commands, which is used by the scheduling algorithms to optimally schedule commands to be sent to the PHY, based on priority, bank/rank status and DDR timing constraints. Write data is stored in a flip-flops until its associated command is issued to the PHY. Read data is handled by the response engine in the DDRC and is returned in the order of scheduled read commands on the HIF.
- The APB Register Block: This block contains the software accessible registers.

6.11.4 DDR Memory Controller Architecture Overview

6.11.4.1 AXI Port Interface (XPI)

The AXI protocol is burst-based with read and write request channels that specify the host ID for the

request, start byte address, burst length, burst size, and burst type. This information is processed by the interface and is used subsequently by the uMCTL2.

The XPI interfaces the AXI application port to the uMCTL2 and performs the following main functions:

- Read address generation
- Write address generation
- Write data generation
- Read data and response generation
- Write response generation

The XPI converts AXI bursts into read and write requests, which are forwarded to the Port Arbiter (PA). In the opposite direction, the XPI converts the responses from the DDRC into appropriate AXI responses. The interface between XPI and PA is same as HIF (but with independent channels for read and write commands).

- The AXI read address channel has the following features:
  - The read transaction can be of any length up to 256 for incremental bursts, and 16 for wrapping bursts.
  - The burst types supported are incremental and wrapping.
  - The burst start address can be unaligned to the AXI data width boundaries.
  - The size of the burst can be less than the full width of the AXI data bus (also known as sub-sized transfers).
  - The signals on the read address channel are registered into the XPI in accordance with the AXI valid/ready handshaking protocol and are synchronous with the AXI clock (aclk).
  - Read requests are accepted if the request can be written into the Read Address Queue (RAQ). This is used to store all the read address requests.
  - The depth of RAQ is set to 4.
- The AXI write address channel has the following features:
  - The write transaction can be of any length up to 256 for incremental bursts, and 16 for wrapping bursts.
  - The burst types supported are incremental and wrapping.
  - The burst start address can be unaligned to the AXI data width boundaries.
  - The size of the burst can be less than the full width of the AXI data bus (also known as sub-sized transfers). Write Address Queue (WAQ) is used to store all the addresses for write requests from a given port. There is a single queue for all AXI IDs from a given port. The write address channel behavior is similar to the read address channel.
  - The depth of WAQ is set to 4.
- Read Data and Response Channel:
  - The XPI handles the common response interface to the DDRC to process the read data from the memory.
  - AXI read data and response channel has a single data storage queue (RDQ - Read Data Queue) with two clocks—AXI clock for reading and uMCTL2 clock for writing.
  - Data from different IDs are stored in the same queue and are returned in the order of read address acceptance.
  - The controller provides an OKAY response for each read, except for exclusive read transactions. The uMCTL2 controller provides an EXOKAY response.

- The read data can be returned from the DDRC in a different order from which the read commands are forwarded from the XPI (due to the re-ordering of read commands in the DDRC to maximize SDRAM bandwidth). A read reorder buffer is implemented in each port to reorder the read data for that port to the same order (per AXI ID) as the order of the AXI read commands.
- The depth of RDQ is set to 16.

6.11.4.2 Port Arbiter (PA)

The Port Arbiter (PA) block arbitrates command requests from up to 7 AXI ports to the HIF of the DDR Controller (DDRC). PA is comprised of multiple tiers of arbitration stages which include:

- Read/write arbitration.
- 2-priority level arbitration based on port aging and expired-VPR/expired-VPW commands (timeout- priority0).
- 2-priority level arbitration for read requests based on DDRC read priorities (HPR/LPR-VPR).
- Round-robin arbitration to resolve ports having the same priority after passing all stages of arbitration.

6.11.4.3 Host Interface (HIF)

HIF is an internal interface when the uMCTL2 is configured to include a multi-port arbiter. In this case, the AXI Port Interface and/or Port Arbiter handle all signals on this interface, and effectively acts as the SoC core.

Read, write, and RMW requests as well as write data are received through the HIF. Following the receipt and acceptance of a write or RMW request, the DDRC requests write data from the interface. The interface subsequently provides the independently timed write data in the requested order (matches the order in which requests are made to the DDRC). Read, write, and RMW requests are provided through a common command bus.

- Requests to the DDRC are throttled in two ways:
  - A credit mechanism that ensures that buffer space is available for any request the SoC makes to the DDRC, prior to the request being made.
  - An independent stall mechanism that throttles requests when address collisions take place or if entering self-refresh through the software self-refresh or hardware low power self-refresh.
- Credit Mechanism:

The DDRC employs a credit mechanism to ensure that buffers do not overflow. The interface making the request to the DDRC, can only request commands for which it has been granted “credits” to issue.

Credits are tracked separately for the following three command types:

- High Priority Read (HPR)1
- Low Priority Read (LPR)
- Write

- Read Requests:

When the LPR or HPR credit count is greater than zero, the interface can issue read requests to the DDRC accordingly.

- Write Requests and Data:

When the WR credit count is greater than zero, the interface can issue write requests to the DDRC

accordingly.

6.11.4.4 DDR Scheduler (DDRC)

Details on DDRC are as follows:

- Write data is stored in an SRAM until its associated command is issued to the PHY.
- Read data is handled by the response engine in the DDRC and is returned in the order of scheduled read commands on the HIF.

6.11.4.5 APB Interface (APB)

The uMCTL2 provides a dedicated APB 3.0 bus interface that is used to access the uMCTL2 software programmable registers. The uMCTL2 converts APB reads and writes into accesses to the internal register file. The APB data width is fixed to 32 bits for compatibility reasons with the Synopsys DWC DDR PHYs. The APB address width is 12 bits. All APB interface signals (p*) are synchronous to pclk. pclk can be asynchronous to the core_ddrc_core_clk.

There are three classes of registers in the uMCTL2:

- Dynamic Registers
- Quasi Dynamic Registers
- Static Registers

Dynamic registers can be written at any time during operation. Static registers can be written only when the controller is in reset. Quasi dynamic registers can be written when the controller is in reset and some specific conditions outside reset. To write them, SWCTL.sw_done has to be set to 0 at the beginning of the programming sequence and set back to '1' at the end. After that, SWSTAT.sw_done_ack has to be polled for acknowledge.

6.11.4.6 DFI Interface

The uMCTL2 controller contains a DFI MC (Memory Controller) interface, which is used to connect to a DFI-compliant PHY, and to transfer address, control, and data to the PHY. In case of a Synopsys DWC DDR PHY, the uMCTL2 interfaces to a PUB block, which acts as the DFI PHY interface. The uMCTL2 controller including its DFI Interface, runs at Half data rate (HDR) clock (referred to as 1:2 frequency ratio mode in the DFI Specification).

The DFI interface is sub-divided into the following interface groups:

- Control Interface:

The control interface is a reflection of the SDRAM control interface including address, bank, chip select, row strobe, column strobe, write enable, clock enable, and ODT control, as applicable for the memory technology.

- Write Data Interface:

The write data interface handles the transmission of write data across the DFI interface. The DFI Specification defines signals and timing relationships. The timing of this interface is configurable, to support all DFI-compliant PHYs through the DFITMG0.dfi_tphy_wrlat and DFITMG0.dfi_tphy_wrdata registers.

- Read Data Interface:

The read data interface handles the return of read data across the DFI interface. The DFI Specification defines signals and timing relationships. The timing of this interface is configurable, to support all DFI-compliant PHYs through the DFITMG0.t_rddata_en register. Update Interface: During system operation, the system may require updates to the internal settings to compensate for environmental conditions. To ensure that updates do not interfere with signals on the SDRAM interface, the DFI interface supports update modes where the DFI read, write, and control interface is suspended from the normal activity.

- Status Interface:

The DFI interface requires status information for initialization and clock control to the SDRAM devices. These signals are used to convey information between the MC and PHY.

- Training (PHY-Independent mode supported by the uMCTL2):

Write and read leveling function performed in a DFI-compliant PHY with no uMCTL2 interaction.

- Low Power Control Interface:

This interface consists of signals that are used to inform the PHY of a low power mode opportunity, as well as how quickly the uMCTL2 requires the PHY to resume normal operation.

The uMCTL2 controller, including its DFI interface runs at the half data rate (HDR) clock (referred to as 1:2 frequency ratio mode in the DFI Specification), so all DFI command and address signals are twice the width of the equivalent DDR SDRAM signals. DFI data signals are four times the width of the equivalent DDR SDRAM signals. The uMCTL2 controller operates with an HDR clock. The PHY handles HDR-to-SDR conversion on the address bus to memory and HDR/DDR conversion on the data buses. In 1:2 frequency ratio mode, core_ddrc_core_clk runs at half the frequency of CK/CK#.

6.11.5 Basic Configuration

6.11.5.1 UMCTL2 Basic Configuration

- Clock Source Configuration
  - Set DDR PLL control registers in CLK_PLL2CTL0, CLK_PLL2CTL1, and CLK_PLL2CTL2.
- Reset Configuration
  - Reset APB interface of UMCTL2 in MCTLPRST (SYS_IPRST0[28]).
  - Reset core of UMCTL2 in MCTLCRST (SYS_IPRST0[29]).

6.11.6 Functional Description

6.11.6.1 Clock and Reset Requirements

The main clock for the uMCTL2 is core_ddrc_core_clk. The core_ddrc_core_clk clock is synchronous to the PHY clock, and it is provided from an external clock tree driving the uMCTL2 and PHY clocks together. All application port clocks (aclk_n) are independently configurable as asynchronous with respect to the main uMCTL2 clock (core_ddrc_core_clk). The APB interface clock (pclk) is normally asynchronous to the core_ddrc_core_clk. The interfaces that need to be asynchronously configured, incur latency and area increase due to the additional clock domain crossing circuitry.

Figure 6.11-3 Example with Three Clock Domains shows a typical example with three clock domains—a core_ddrc_core_clk domain, a pclk domain, and aclk domain.

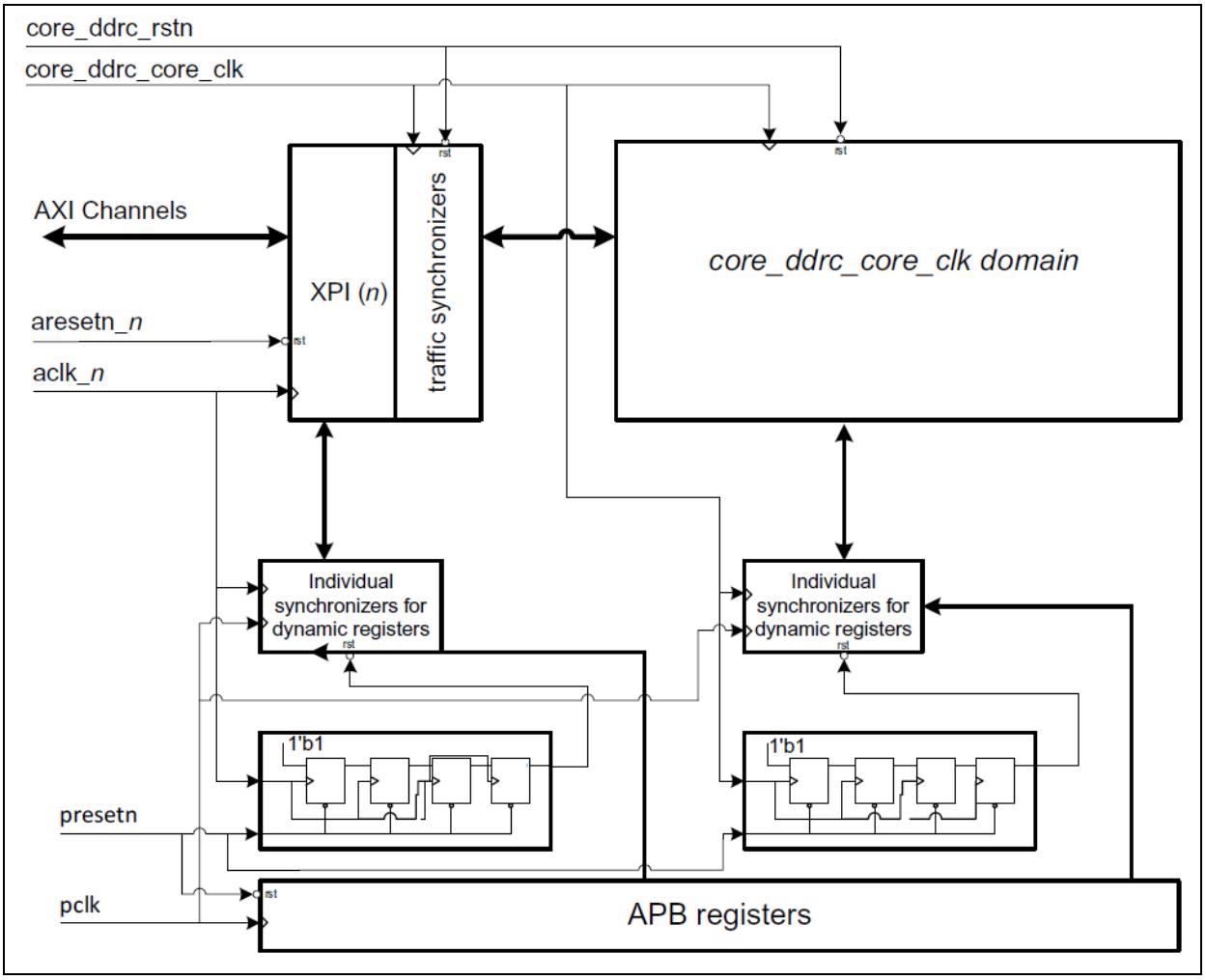


Figure 6.11-3 Example with Three Clock Domains

Every input signal is assumed to be synchronous to a specific clock unless otherwise specified. Port related signals are always synchronous to the correspondent aclk_n, while DDRC inputs are synchronous to core_ddrc_core_clk. The core_ddrc_rstn reset should always be asserted before the presetn reset.

6.11.6.2 Register Types

Registers are accessed through the APB interface. The list of registers is provided in the section of Register Description. The following sections outline under what circumstances the APB registers can be written. Most registers are intended to be initialized while the uMCTL2 controller is in reset (core_ddrc_rstn = 0), and should not need to be changed afterwards. The exceptions to this rule are listed in the following sections. core_ddrc_core_clk must be brought up and running before uMCTL2 controller is brought out of reset (core_ddrc_rstn is de-asserted).

The "Programming Mode" field in the description column of the register tables specifies the type of register, whether static, dynamic, or quasi dynamic.

6.11.6.3 Dynamic Registers

The dynamic registers can be written at any time during the operation of the uMCTL2. As the APB block is typically in a different clock domain from the uMCTL2 controller, these signals are synchronized to the uMCTL2 controller clock domain. For the remaining registers which are considered static, synchronization is not required. To know if a particular register is of the type dynamic, see the

"Programming Mode" field in the description column of the register tables in section of Register Description.

6.11.6.4 Refresh Related Registers

The refresh related registers are dynamic but to update them the following must be done:

1. Change the refresh associated register as desired.
2. After the changed register is known stable, toggle the RFSHCTL3.refresh_update_level signal.

The SDRAM controller notes the refresh_update_level signal change and updates all refresh-related register values accordingly. This mechanism is needed to avoid sampling errors in the target clock domain as well as to allow the controller to provide special handling (such as issuing an additional refresh and resetting the refresh timer if needed) when a refresh-related timing register has changed.

Refresh related registers that can be updated are denoted as "Dynamic - Refresh Related" in the "Programming Mode" field in section of Register Description. Most of the refresh related registers are dynamic. However RFSHCTL1.refresh_timer*_start_value_x32 can only be programmed during initialization.

6.11.6.5 Quasi Dynamic Registers

In addition to the dynamic registers, the following categories of registers can be written after reset:

1. Group 1: Registers that can be written when No Read/Write Traffic is Present at the DFI
2. Group 2: Registers that can be written in Self-refresh, DPD, and MPSM Modes
3. Group 3: Registers that can be written When Controller is Empty

Each category requires specific conditions for the registers to be programmed. Once the programming conditions are met, SWCTL.sw_done register must be programmed to 1’b0 to enable the software programming. If UMCTL2_OCCAP_EN=1 && OCCAPCFG.occap_en=1, the SWSTAT.sw_done_ack must be read as 1’b0. Once the programming is completed, SWCTL.sw_done must be set to 1’b1 and SWSTAT.sw_done_ack must be read as 1’b1. This is to ensure that quasi dynamic registers are propagated correctly to the destination clocks. The protocol flowchart is shown in Figure 6.11-4 and Figure 6.11-5.

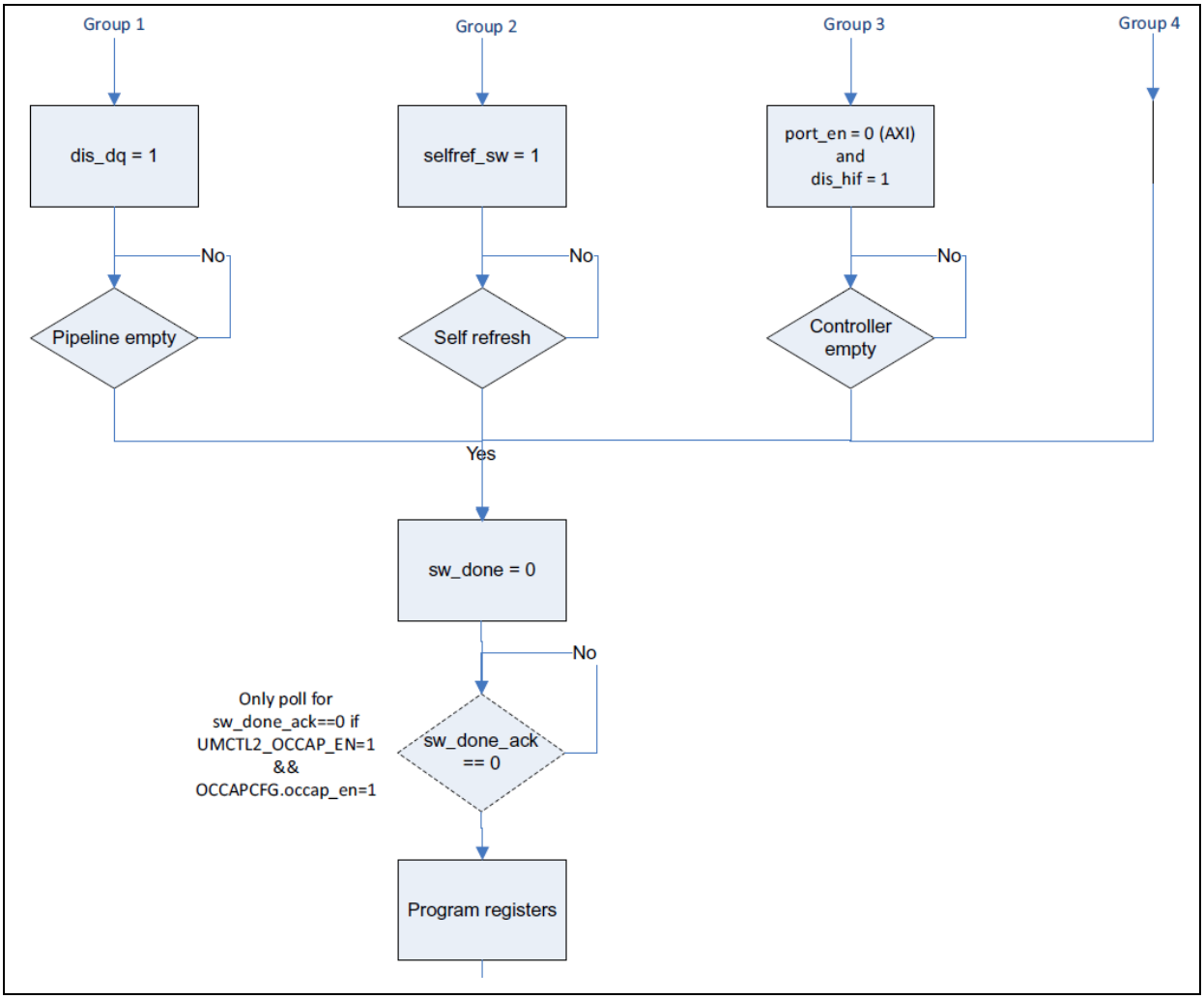


Figure 6.11-4 Software Handshake Protocol

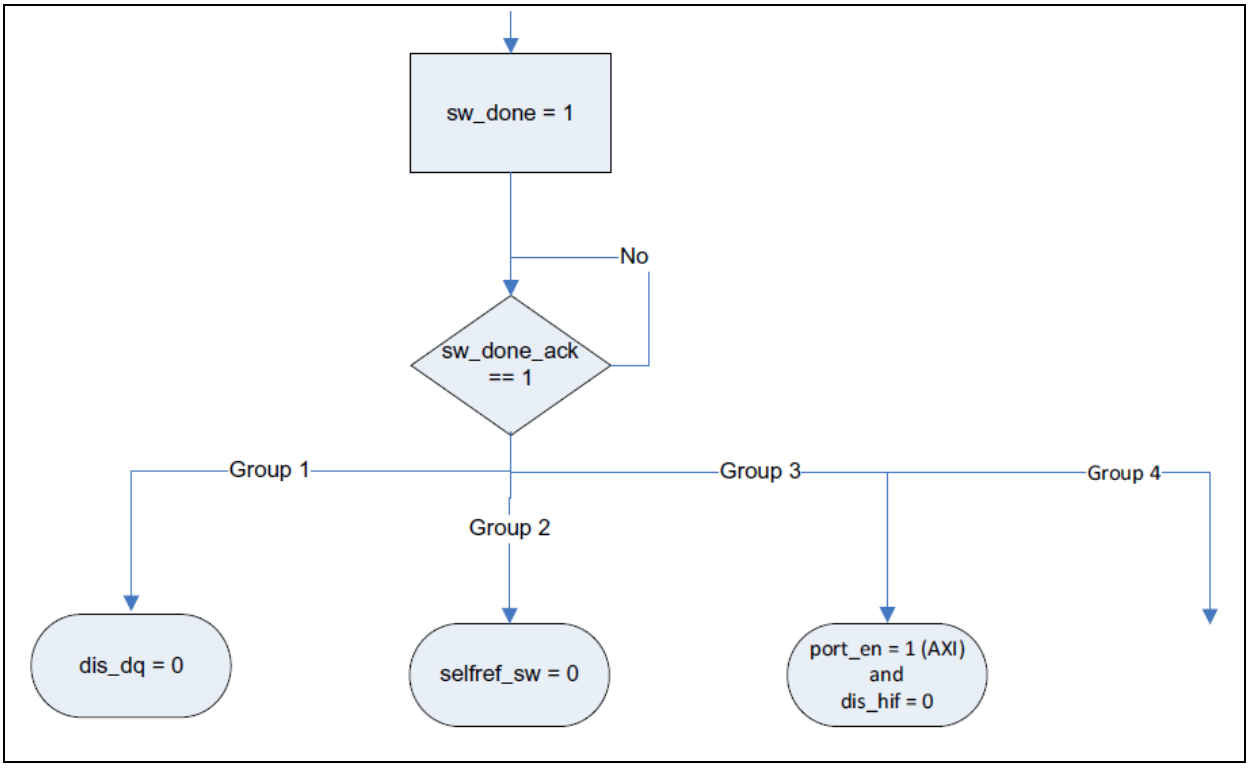


Figure 6.11-5 Software Handshake Protocol (Continuous)

The "Programming Mode" field in the description column of the register tables in section of Register Description specifies the quasi dynamic type registers along with the group. Traffic has to be enabled again depending on the register category as described in the following sections.

6.11.6.6 Group 1: Registers that can be Written when No Read/Write Traffic is Present at the DFI

By setting DBG1.dis_dq register and polling DBGCAM.wr_data_pipeline_empty and DBGCAM.rd_data_pipeline_empty1, it is possible to prevent any read or write traffic from being sent on the DFI. In this mode, it is safe to write to the group 1 registers. Re-enable the traffic by writing DBG1.dis_dq to 1'b0.

6.11.6.7 Group 2: Registers that can be Written in Self-refresh, DPD, and MPSM Modes

When the uMCTL2 has entered self-refresh mode through software (PWRCTL.selfref_sw), the DFI bus is idle until software exits self-refresh. For self-refresh, ensure that self-refresh is not caused by "Automatic Self-refresh only" by checking the STAT.operating_mode = 3'b011 and STAT.selfref_type = 2'b10. In this mode, it is safe to write the group 2 registers. Re-enable the traffic by writing PWRCTL.selfref_sw to 1'b0.

6.11.6.8 Group 3: Registers that can be Written When Controller is Empty

For multi-port configurations, PCTRL_n.port_en is used to enable or disable the input traffic per port. The Controller idleness can be polled first from PSTAT register (wr_port_busy_n and rd_port_busy_n bit fields) and should read as PSTAT==32'b0 (not busy). DBG1.dis_hif is used to enable or disable the input traffic to the controller. Then, DDRC CAM/pipeline empty status must be polled ((DBGCAM.dbg_wr_q_empty==1'b1)&&(DBGCAM.dbg_rd_q_empty==1'b1)&&(DBGCAM.wr_data_pipeline_empty==1'b11) && (DBGCAM.rd_data_pipeline_empty==1'b1)).

In this mode, it is safe to write the group 3 registers. Enable the traffic by writing 1'b1 to PCTRL_n.port_en or to DBG1.dis_hif if using HIF only.

6.11.6.9 Notes on Timing Registers

1. Note 1: Computation for all the registers in units of 32, 1024, or 4096 clocks

This note refers to register fields whose names include "x32", "x1024", or "x4096". The uMCTL2 contains a timer which issues a pulse once every 32 clock cycles, another which issues a pulse once every 1024 clock cycles and another which issues a pulse once every 4096 clock cycles. "x32" register fields count pulses of the 32-cycle timer, "x1024" register fields count pulses of the 1024-cycle timer, and "x4096" register fields count pulses of the 4096-cycle timer. These timers are shared by the logic for all of these register fields, and the various periods start without any guaranteed relation to the phase of this timer.

- Therefore for "x32" register fields:
  - A programmed value of 0 gives 0 clock cycles
  - A programmed value of 1 gives a 1 to 32 cycle delay
  - A programmed value of 2 gives a 33 to 64 cycle delay and so on
- For "x1024" register fields:
  - A programmed value of 0 gives no delay
  - A programmed value of 1 gives a 1 to 1,024 cycle delay
  - A programmed value of 2 gives a 1,025 to 2,048 cycle delay and so on
- For "x4096" register fields:
  - A programmed value of 0 gives no delay
  - A programmed value of 1 gives a 1 to 4,096 cycle delay
  - A programmed value of 2 gives a 4,097 to 8,192 cycle delay and so on

For minimum delays, the control signal must be set to the smallest number for which the resulting delay is ALWAYS greater than or equal to the required delay; for maximum delays or nominal refresh, this must be set such that the delay is ALWAYS lesser than or equal to the required delay.

This scheme is used to reduce the gate count that would be required to enforce these constraints more precisely.

2. Note 2:

Most of the SDRAM timing registers are in terms of clock cycles. But the value given in an SDRAM Datasheet may be in terms of ns/ps. These are primarily minimum timings. Before using them to calculate value to program in the register, it is expected that these are divided by tCK (SDRAM clock period) and rounded up to the next integer value if division does not result in a whole integer value.

Exceptions to this rule are related to maximum timings, where these should be rounded down rather than rounded up, if division does not result in a whole integer value.

These include:

- RFSHTMG.t_rfc_nom_x32
- DRAMTMG0.t_ras_max

Certain SDRAM timing registers clarify this by using the following functions in their descriptions:

- RoundUp (X/Y) means that if X divided by Y does not result in a whole integer value, the result should be rounded up to the next integer value. For example, RoundUp(1000/150) = 7, RoundUp(1050/150) = 7.
- RoundDown (X/Y) means that if X divided by Y does not result in a whole integer value, the result should be rounded down to the next integer value. For example, RoundDown(1000/150) = 6, Round- Down(1050/150) = 7.

6.11.6.10 Register Groups

APB registers are accessed through the APB interface, but take effect in the core_ddrc_core_clk or aclk domain (destination domain). All APB registers must be initialized in reset mode (core_ddrc_rstn is asserted). The behavior of the uMCTL2 is unpredictable if these registers contain illegal values when the core_ddrc_rstn signal is de-asserted, or any time thereafter. Some APB registers can be changed at any time. They are defined as dynamic registers as shown in Figure 6.11-3. Other APB registers must not be changed after the core_ddrc_rstn is de-asserted. Group 1: Registers that can be Written when No Read/Write Traffic is Present at the DFI, Group 2: Registers that can be written in Self-refresh, DPD, and MPSM Modes, Refresh Related Registers are exceptions to this.

The APB registers are reset by the signal presetn, which is de-asserted synchronously with pclk. Static registers are sampled directly in the destination domains (core_ddrc_core_clk/aclk domains) without synchronizing circuits. The de-assertion of the signal presetn is synchronized to the destination domain clocks and used to de-assert the reset on the static registers in their destination domains. Therefore, the static registers must be programmed when core_ddrc_core_clk/aclk_n and pclk are stable, presetn is de-asserted (at least for 4 cycles of clock in the destination domain) and core_ddrc_rstn/aresetn_n are asserted.

Synchronizing circuits are used to transfer the signals for dynamic registers to their destination domains. While these registers must be initialized with core_ddrc_rstn asserted, they can be changed thereafter with core_ddrc_rstn de-asserted.

6.11.6.11 Controller Initialization

The following sections describe the programming sequence that must be executed at power-up to bring the DWC_ddr_umctl2 controller, the PHY, and the memories into a state where HIF reads and writes can be performed. Figure 6.11-6 shows the relationship between clocks and resets for initialization of registers. Power and clocks must be in accordance with the requirements of full ASIC/SoC design requirements, and the SDRAM power and clocks must in accordance with the SDRAM specifications.

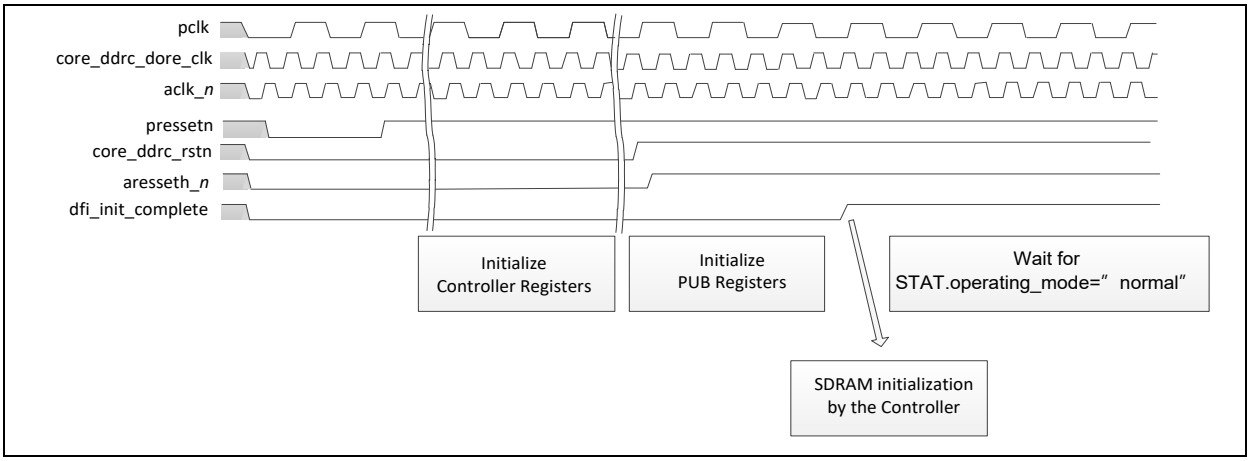


Figure 6.11-6 Simplified Relationship between Clocks and Resets for Register Initialization

After power-up:

1. Assert the core_ddrc_rstn and aresetn_n resets
2. Assert presetn
3. Start the clocks (pclk, core_ddrc_core_clk, aclk_n)
4. De-assert presetn once the clocks are active and stable
5. Allow 128 cycles for synchronization of presetn to core_ddrc_core_clk and aclk domains and to permit initialization of end logic

6. Initialize the registers
7. De-assert the resets (core_ddrc_rstn and aresetn_n) Any subsequent access to the programming interface of the registers which may be accessed outside of reset must happen when both pclk and core_ddrc_core_clk are active and stable

6.11.6.12 PHY Initialization

Because the Synopsys DWC DDR PHY is used, the input DFIMISC.dfi_init_complete_en must be set to 0 at this point. This is because the DWC DDR PHY toggles the phy_dfi_init_complete signal several times during its initialization routine, and the DWC_ddr_umctl2 controller must be made insensitive to this toggling during the PHY initialization phase. After all the registers are programmed, de-assert core_ddrc_rstn. Before the memory can be accessed by the DWC_ddr_umctl2 controller, the PHY must be initialized. When using the DWC DDR3/2 PHY, the PHY Utility Block (PUB) can be used for initialization. For an outline of the full initialization sequence for a DWC DDR3/2 PHY using the PUB, refer to Table 6.11-1. The following tables summarize the initialization procedure when using the Synopsys DWC DDR3/2 PHY. The tables assume SDRAM initialization is performed by DWC_ddr_umctl2. If this is not the case (INIT0.skip_dram_init=1) the SDRAM initialization must be performed by the PUB.

Step	Application	SVTB Task	Notes
1	Program the DWC_ddr_umctl2 registers		Ensure that DFIMISC.dfi_init_com plete_en is set to 0
2	De-assert core_ddrc_rstn	reset_dut	
3	Start PHY initialization by accessing relevant PUB registers (for example, DXnGCR, DCR, PTR*, MR*, DTPR*)	phy_init	
4	Monitor PHY initialization status by polling the PUB register PGSR0.IDONE	phy_init	
5	Indicate to the PUB that the controller performs SDRAM initialization by setting PIR.INIT and PIR.CTLDINIT, and poll PGSR0.IDONE	phy_init	
6	Set SWCTL.sw_done to ‘0’ If UMCTL2_OCCAP_EN=1 && OCCAPCFG.occap_en=1, then also poll for SWSTAT.sw_done_ack=0	phy_init	
7	Set DFIMISC.dfi_init_complete_en to ‘1’	phy_init	DWC_ddr_umctl2 waits for phy_dfi_init_complet e to be asserted
8	Set SWCTL.sw_done to ‘1’	phy_init	
9	Wait for SWSTAT.sw_done_ack to become ‘1’	phy_init	
10	Wait for DWC_ddr_umctl2 to move to “normal” operating mode by monitoring STAT.operating_mode signal	reset_dut	DWC_ddr_umctl2 performs SDRAM initialization automatically
11	Set SWCTL.sw_done to ‘0’ If UMCTL2_OCCAP_EN=1 && OCCAPCFG.occap_en=1, then also poll for SWSTAT.sw_done_ack=0		
12	Disable auto-refreshes and power down by setting RFSHCTL3.dis_auto_refresh = 1 and PWRCTL.powerdown_en = 0, and set DFIMISC.dfi_init_complete_en to ‘0’	pub_training	Mask transitions in phy_dfi_init_complet e during PUB training
13	Set SWCTL.sw_done to ‘1’		
14	Program PUB’s DTCR to enable refreshes during training and to specify which ranks to train	pub_training	

15	Configure PUB's PIR register to specify which training steps to run	pub_training/ poll_pub_training	
16	Monitor PUB's PGSR0.IDONE to poll for completion of training sequence	pub_training/ poll_pub_training	
17	Set back registers in step 12 to the original values if desired. When training is complete if there is data in the SDRAM that needs to refreshed the programming of RFSHCTL3.dis_auto_refresh = 0 should happen quickly so that the maximum supported time without Refresh command is not violated. Refer to relevant JEDEC protocol standard for further details on the Refresh command requirements, for example, for DDR3, this is 9*tREFI.		

Table 6.11-1 DWC_ddr_umctl2 and Memory Initialization with PUB

6.11.6.13 SDRAM Initialization

The DWC_ddr_umctl2 controller performs SDRAM initialization automatically when it has detected that the PHY has completed its initialization. Therefore, the DFIMISC.dfi_init_complete_en register must be asserted at this point, so that the SDRAM initialization sequence in the DWC_ddr_umctl2 controller is triggered. The SDRAM initialization sequence can be skipped by programming INIT0.skip_dram_init = 2'b01/2'b11. In this case SDRAM initialization must be performed by the PUB.

The controller performs the SDRAM initialization sequence, depending on the SDRAM protocol. A specific command sequence must be followed to properly initialize SDRAM modules. Most of the sequence is standard, but parts of it must be controlled differently based on the system configuration, clock speeds, and specific SDRAM parts in use. For this reason, the uMCTL2 provides a great deal of flexibility in how the initialization sequence is performed. When you/SoC appropriately sets the Registers Related to SDRAM Initialization, the uMCTL2 automatically performs the necessary sequence to properly initialize the DDR SDRAM devices. These registers must be set before bringing the uMCTL2 out of reset.

6.11.6.14 DDR3 Initialization Sequence

For DDR3, the uMCTL2 initialization state machine executes the following initialization sequence:

1. Power-up.
2. Maintain dfi_reset_n low for duration specified by INIT1.dram_rstn_x1024. Specification requires at least 200 us with stable power.
3. Issue NOP/deselect for duration specified by INIT0.pre_cke_x1024. Specification requires at least 500 us.
4. Assert CKE and issue NOP/deselect for INIT0.post_cke_x1024 (specification requires at least tXPR).
5. Issue MRS (mode register set) command to load MR2 with INIT4.emr2 value followed by NOP/dese-lect for duration of DRAMTMG3.t_mrd.
6. Issue MRS command to load MR3 with INIT4.emr3 followed by NOP/deselect for duration of DRAMTMG3.t_mrd.
7. Issue MRS command to load MR1 with INIT4.emr followed by NOP/deselect for duration of DRAMTMG3.t_mrd.
8. Issue MRS command to load MR0 with INIT3.mr followed by NOP/deselect for duration of DRAMTMG3.t_mod.
9. Issue ZQCL command to start ZQ calibration and wait for INIT5.dev_zqinit_x32.
10. Wait for INIT5.dev_zqinit_x32 counting to finish. Ensure wait from step 8 is larger than tDLLK.

11. The uMCTL2 controller is now ready for normal operation.

6.11.6.15 DDR2 Initialization Sequence

For DDR2, the uMCTL2 initialization state machine executes the following initialization sequence:

1. Power-up.
2. Issue NOP/deselect for duration specified by INIT0.pre_cke_x1024. Specification requires at least 200 us with stable power and clock.
3. Assert CKE and issue NOP/deselect for INIT0.post_cke_x1024 (specification requires at least 400 ns).
4. Issue precharge all followed by NOP/deselect for DRAMTMG4.t_rp cycles.
5. Program EMR2 to the INIT4.emr2 value followed by NOP/deselect for DRAMTMG3.t_mrd cycles.
6. Program EMR3 to the INIT4.emr3 value followed by NOP/deselect for DRAMTMG3.t_mrd cycles.
7. Enable DLL by programming EMR to the INIT3.emr value followed by NOP/deselect for DRAMTMG3.t_mrd cycles.
8. Issue DLL reset by programming MR to the INIT3.mr value followed by NOP/deselect for DRAMTMG3.t_mrd cycles.
9. Issue precharge all followed by NOP/deselect for (DRAMTMG4.t_rp+1) cycles.
10. Issue refresh followed by NOP/deselect for RFSHTMG.t_rfc_min cycles. Repeat it for 9 times.
11. Program MR without resetting the DLL by setting MR to INIT3.mr value with bit 8 set to ‘1’.
12. The uMCTL2 controller is now ready for normal operation.

6.11.6.16 SDRAM Timing Parameters

In DDR memory controller, there are many timings parameters as programmed in registers DDRCTRL_DRAMTMG* according to used SDRAM specification. The following registers are related to timings parameters of SDRAM.

- DRAMTMG0 to DRAMTMG14
- ODTTMG
- DFITMG0 and DFITMG1

6.11.6.17 Power Saving Mode

The uMCTL2 supports various methods to save power within the system:

1. Power saving opportunities within the SDRAM. The uMCTL2 supports various SDRAM power-saving modes such as, precharge power-down, self-refresh, and support for disabling clock to the DRAM through dfi_dram_clk_disable.
2. Power saving opportunities within the PHY.
3. Power saving opportunities from an external SoC low power controller driven through an external hardware low power interface.

6.11.6.18 SDRAM Power Saving Features

In multi-rank systems, these power-saving modes cannot be applied on a per-rank basis. If applied, they are always applied globally. When enabled, the uMCTL2 automatically enters and exits precharge power-down mode based on a programmable idle timeout period.

Self-refresh can be entered/exited through three ways:

1. Based on a programmable idle timeout period (similar to precharge power-down idle timeout)
2. Explicitly controlled through software
3. Through the hardware low power interfaces

In addition, dfi_dram_clk_disable can be asserted to disable the clocks to the DRAM. This can be done in the following modes:

- Self-refresh
- Power-down

Note: It is valid to enable any combination of Power-down and Self-refresh modes simultaneously:

- Power-down: PWRCTL.powerdown_en = 1
- Automatic self-refresh: PWRCTL.selfref_en = 1
- Software self-refresh: PWRCTL.selfref_sw = 1

Enabling assertion of ddrc_dfi_dram_clk_disable is valid in combination with any of the power-saving modes.

6.11.6.19 Precharge Power-down

Entering Precharge Power-down

When PWRCTL.powerdown_en = 1, uMCTL2 automatically enters precharge power-down when the period specified by PWRTMG.powerdown_to_x32 has passed while the uMCTL2 is idle (except for issuing refreshes).

Entering precharge power-down mode involves the following steps:

1. If there is a self-refresh exit previously, wait for at least one refresh command to all active ranks. Auto-refresh logic must be enabled, or refresh must be issued using direct software requests of refresh command through DBGCMD.rank*_refresh.
2. Precharging (closing) all open pages. Pages are closed one-at-a-time in no specified order.
3. Waiting for tRP (row precharge) idle period.
4. Issuing the command to enter precharge power-down (NOP/deselect with CKE = 0). For multi-rank systems, all chip-selects are asserted so that all ranks enter precharge power-down simultaneously.
5. This step occurs only if DFI low power interface for power-down is enabled (DFILPCFG0.dfi_lp_en_pd). Attempts an entry to low power mode through DFI low power interface and with dfi_lp_wakeup set by DFILPCFG0.dfi_lp_wakeup_pd.

Note: If the uMCTL2 receives a read or write request from the SoC during step2 or step3, the power-down entry is immediately aborted. The same is true if PWRCTL.powerdown_en is driven to ‘0’ during step 2 or step 3.

Once the uMCTL2 has put the DDR SDRAM devices in precharge power-down mode, the uMCTL2 automatically performs the precharge power-down exit sequence for any of the following reasons:

- A refresh cycle is required to any rank in the system.
- The uMCTL2 receives a new request from the SoC core.
- A self-refresh entry is requested.
- PWRCTL.powerdown_en is set to 0.

The uMCTL2 follows these steps when exiting precharge power-down mode:

1. Inserting any NOP/deselect commands required to satisfy the tCKE requirement after entering precharge power-down.
2. This step occurs only if DFI low power mode entry during power-down entry is successful. Performs an exit from DFI low power mode. DFI low power mode is exited after the wakeup time specified by DFILPCFG0.dfi_lp_wakeup_pd, but not earlier than DFITMG1.dfi_t_dram_clk_enable + DRAMTMG6.t_ckpdx clock cycles.
3. Issuing the power-down exit command (NOP/deselect with CKE = 1). For multi-rank systems, all chip-selects are asserted so that all ranks exit precharge power-down simultaneously.
4. Issuing NOP/deselect for the period defined by tXP.

6.11.6.20 Self-refresh Mode

Entering Self Refresh

The uMCTL2 puts the DDR SDRAM devices into Self Refresh mode in the following cases:

1. When the PWRCTL. selfref_en bit is set and no reads or writes are pending in the uMCTL2 for the period specified by PWRTMG.selfref_to_x32. This is referred to as automatic Self Refresh.
2. When the PWRCTL.selfref_sw bit is set, this is referred to as the software self-refresh entry. You can decide whether the existing controller commands in DDRC are performed or not before the entering the Self Refresh mode sequence occurs. This allows you to differentiate between cases when the clock is removed from the uMCTL2 upon entering Self Refresh and cases when it is not removed. If it is not removed, you can speed up Self Refresh entry by not draining the CAMs. This can be done by setting software register PWRCTL.dis_cam_drain_selfref.
3. When a hardware low power entry request occurs (on csysreq_ddrc/csysack_ddrc) with cactive_in_ddrc = 0, no outstanding commands, and as long as the uMCTL2 is not in initialization or deep power-down or maximum power saving mode. This is referred to as an accepted hardware low power self-refresh entry. When accepted, the DDRC's hif_cmd_stall is driven high to stop new commands from being accepted and existing controller commands in the DDRC are performed before the following sequence occurs.

Note that STAT.selfref_type register field is 2'b11 if automatic self-refresh feature is the only cause of self-refresh. If software self-refresh or hardware low power self-refresh occurs, STAT.selfref_type = 2'b10.

Automatic self-refresh has the lowest priority followed by both software and hardware low power self-refresh. A software self-refresh entry means that a self-refresh exit occurs only if a software self-refresh exit occurs. Similarly, a hardware low power self-refresh entry means a self-refresh exit occurs only if a hardware low power self-refresh exit occurs. If both software and hardware low power self-refresh entry occurs, self-refresh exit occurs only if both software and hardware low power self-refresh exits occur.

Exiting Self-refresh

The uMCTL2 takes the DDR SDRAM out of self-refresh mode under the following conditions:

1. Whenever the PWRCTL.selfref_en input is de-asserted, or new commands are received by the uMCTL2, as long as automatic Self Refresh is only cause for self-refresh entry (STAT.selfref_auto_flag = 1).
2. Whenever the PWRCTL.selfref_sw bit is de-asserted. This is referred to as software Self Refresh exit.
3. When a hardware low power exit request occurs (on csysreq_ddrc/csysack_ddrc). This is referred to as hardware low power self-refresh exit.

Assertion of dfi_dram_clk_disable

Assertion of dfi_dram_clk_disable occurs only if PWRCTL.en_dfi_dram_clk_disable = 1. In

DDR2/DDR3, dfi_dram_clk_disable can be asserted only in self-refresh mode.

Power Saving in PHY Through DFI Low Power Interface

Based on whether SDRAM is in Precharge Power Down, Self-Refresh, the PHY can be placed in power saving modes through the DFI low power interface. In a DDR memory subsystem, it is advantageous to place the PHY in a low power state when the uMCTL2 has knowledge that the memory subsystem remains idle for a period of time. Depending on the state of the system, the uMCTL2 communicates state information to the PHY allowing the PHY to enter the appropriate power saving state. This interface consists of signals that are used to inform the PHY of a low power mode opportunity, as well as how quickly the uMCTL2 requires the PHY to resume normal operation.

Software control is provided in DFILPCFG0 registers (see UMCTL2_REGS Registers on chapter 6.11.8) for the following:

- This is an optional interface for a DFI-compliant PHY. The interface signals always exist in the uMCTL2. But the software can enable this interface in self-refresh and/or power-down mode - DFILPCFG0.dfi_lp_en_sr and DFILPCFG0.dfi_lp_en_pd respectively.
- Software control over what is driven on dfi_lp_wakeup signal (and its associated timing) in self-refresh or power-down mode - DFILPCFG0.dfi_lp_wakeup_sr and DFILPCFG0.dfi_lp_wakeup_pd respectively.

Hardware Low Power Interfaces

Power saving opportunities from an external SoC low power controller driven through an external hardware low power interface (based on AMBA 3 AXI protocol low power control interface). There is a separate hardware low power interface for each AXI port and dedicated hardware low interface for the DDRC.

The hardware low power interface can be used to enter/exit self-refresh. For this 4-signal interface, the functionality is enabled by setting HWLPCTL.hw_lp_en = 1. Once this bit is set, the input csysreq_ddrc can trigger the entry into the self-refresh operating mode. A hardware low power entry trigger is ignored/denied if any bit of the input cactive_in_ddrc = 1, or if the DDRC is not empty, or if a controller driven R/W/RMW command is on the controller interface (hif_cmd_valid1 = 1). Otherwise, it can be accepted depending on the current state of the uMCTL2. If accepted, the DDRC's hif_cmd_stall = 1 to stop new commands from being accepted, existing controller commands in DDRC are performed before self-refresh entry occurs. cactive_in_ddrc is 1 bit per port, and each bit is driven by the cactive_n of each port. Also, cactive_in_ddrc is internally used to evaluate if self-refresh entry is allowed or not. It might have more registering stages than csysreq_ddrc. This is to avoid CDC issues.

When HWLPCTL.hw_lp_en = 1, the outputs csysack_ddrc and cactive_ddrc provide feedback as required by the AXI low power interface specification (this interface operation is defined by the AXI Specification). The csysack_ddrc signal acknowledges the request to go into self-refresh operating mode, and cactive_ddrc indicates when the uMCTL2 is in self-refresh operating mode.

The cactive_ddrc output can also be used by an external low power controller to decide when to request a transition to self-refresh. When HWLPCTL.hw_lp_idle_x32 > 0, cactive_ddrc = 0 indicates that the uMCTL2 is idle for at least HWLPCTL.hw_lp_idle_x32 * 32 * core_ddrc_core_clk cycles.

In self-refresh mode, the cactive_ddrc output can be used by an external low power controller to trigger a self-refresh exit. Always, cactive_ddrc is driven high when any bit of cactive_in_ddrc = 1 or hif_cmd_valid = 1 or if there are outstanding commands in the DDRC. The low power controller should re-enable the clocks when cactive_ddrc is driven high while in the self-refresh state.

The behavior of the uMCTL2 with respect to automatic precharge power-down or automatic self-refresh and cactive_in_ddrc is selectable through software (HWLPCTL.hw_lp_exit_idle). If HWLPCTL.hw_lp_exit_idle = 1, automatic precharge power-down and automatic self-refresh is exited (if cactive_in_ddrc = 1) and they do not occur until cactive_in_ddrc = 0. Note that it does not cause the exit of self-refresh by DDRC hardware low power interface and/or software (PWRCTL.selfref_sw). This improves memory utilization performance. cactive_in_ddrc informs DDRC that there is RD/WR command imminent, and DDRC exits the various SDRAM power saving modes early so that it can

perform the RD/WR sooner.

- Removal of DDRC Clock Through AXI Hardware Low Power Interface Handshaking

The core_ddrc_core_clk can be removed through AXI hardware low power interface handshaking. An entry to Self-Refresh occurs as shown in Figure 6.11-7. In this example, the system initiates the low power state entry through de-assertion of csysreq_ddrc.

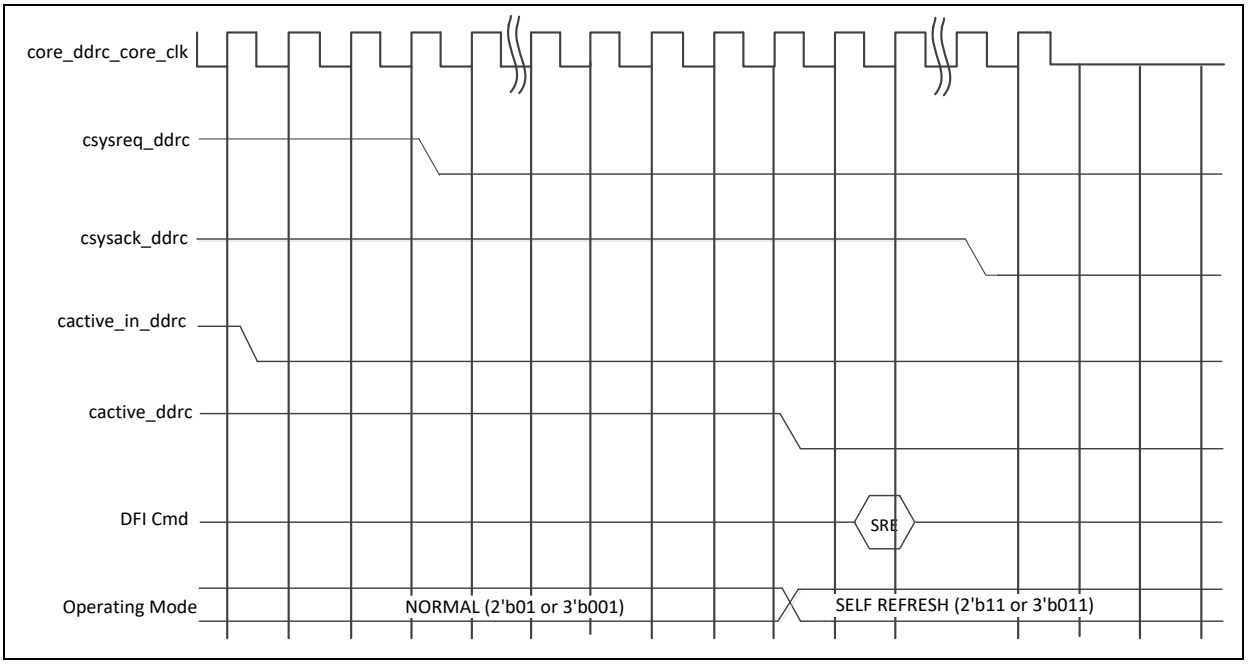


Figure 6.11-7 Entry to Self-refresh through Hardware Low Power Interface of DDRC

An exit from self-refresh occurs as shown in Figure 6.11-8. In this example, the system initiates the low power state exit by asserting csysreq_ddrc.

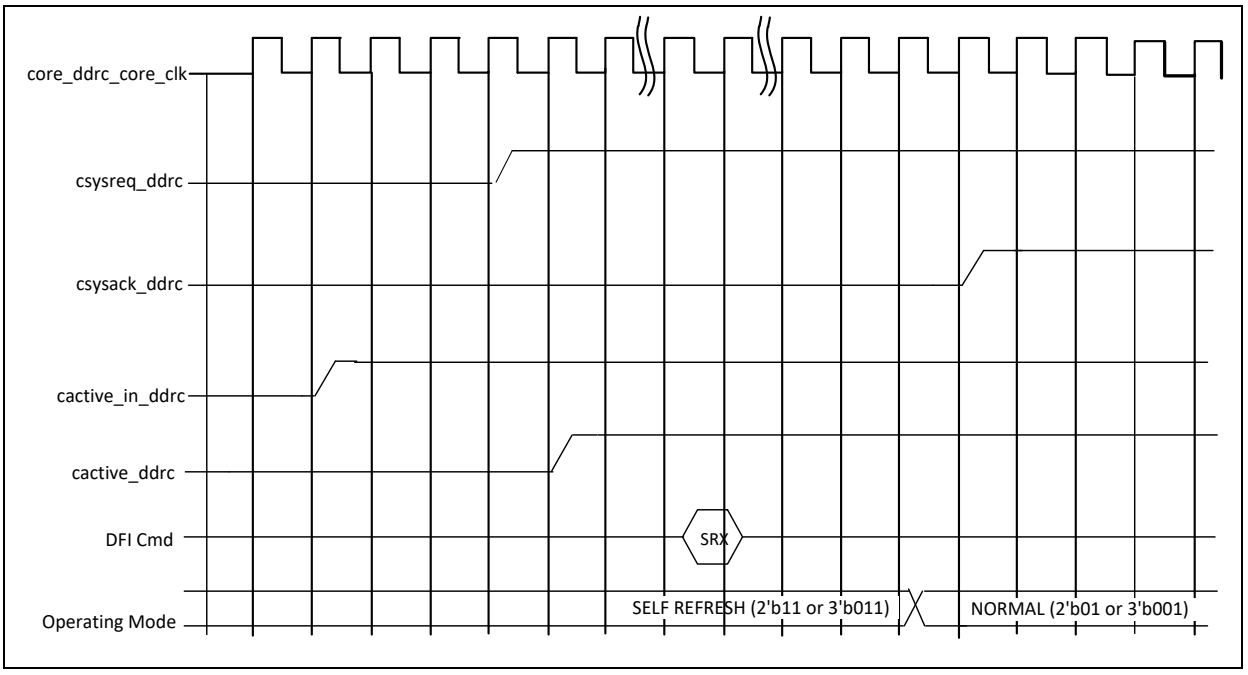


Figure 6.11-8 Exit from Self-refresh through Hardware Low Power Interface of DDRC

Removal of DDRC Clock based on Status of cactive_ddrc and stat_ddrc_reg_selfref_type

Figure 6.11-9 is an example of a clock gating circuit that could be used to remove the DDRC clock based on the status of cactive_ddrc and stat_ddrc_reg_selfref_type.

The DDRC clock can be removed if the following are true for ddrc_auto_sr_dly cycles:

- cactive_ddrc = 1'b0
- stat_ddrc_reg_selfref_type = 2'b11 (SDRAM is in self-refresh and self-refresh is caused by automatic self-refresh only)

The ddrc_auto_sr_dly cycles are required before gating the clock to allow for the self-refresh status to propagate to the APB domain so the STAT.selfref_type register field also reflects the status. A value of 31 cycles is normally sufficient. It is recommended to bypass/disable the gating circuitry through software (as indicated by the signal ddr_cg_enable_reg in Figure 6.11-9). It is assumed that the following registers are programmed as mentioned:

- PWRCTL.selfref_en is programmed to 1
- HWLPCTL.hw_lp_en is programmed to 1
- HWLPCTL.hw_lp_exit_idle_en is programmed to 1

The DDRC clock must be enabled if either of the following is true:

- cactive_ddrc = 1'b1
- stat_ddrc_reg_selfref_type != 2'b11 (SDRAM is in self-refresh and self-refresh is caused by automatic self-refresh only).

Figure 6.11-9 is block diagram of DDR top wrapper module, and it adds a clock gating circuit that could be used to remove the DDRC clock based on the status of cactive_ddrc and stat_ddrc_reg_selfref_type.

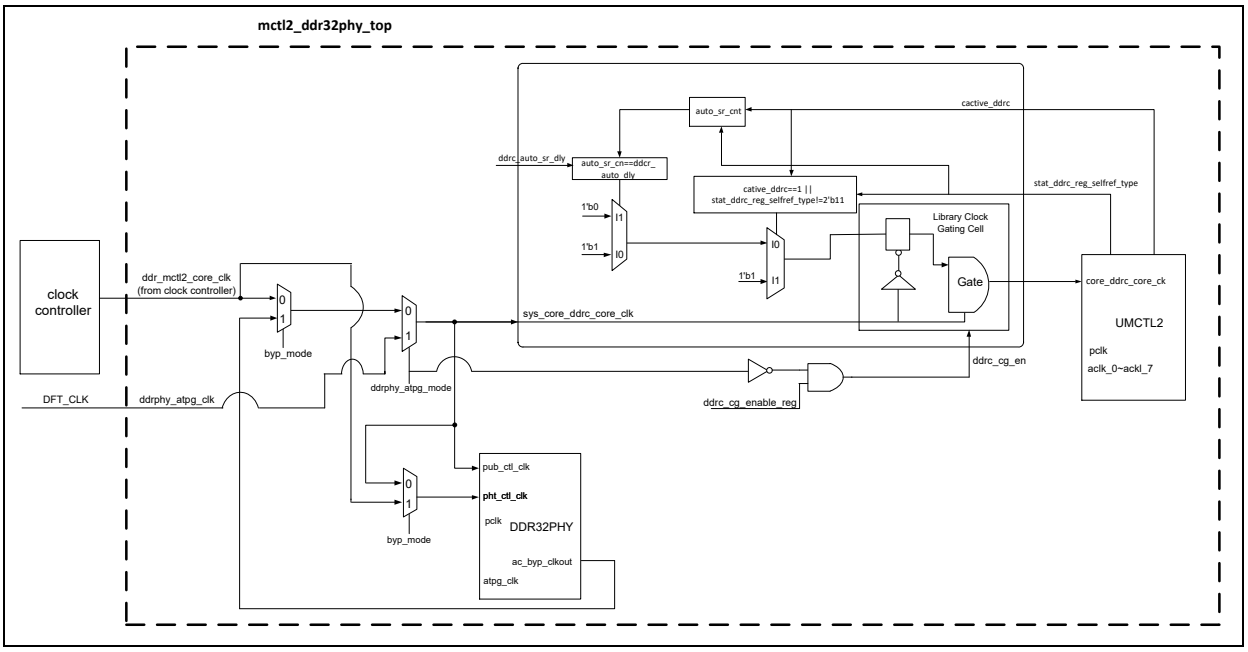


Figure 6.11-9 Block Diagram of DDR Top Wrapper Module

AXI Low Power Interface

The hardware low power interface can be used to enter/exit a low power state for the port. The input csysreq_n can trigger the entry into the low power state. A hardware low power entry trigger is accepted if the port is idle. Idle means there are no outstanding read or write commands from port AXI port N and there is no data accepted in the Write Data Queue. Otherwise, the request is rejected.

The outputs csysack_n and cactive_n provide feedback as required by the AXI low power interface specification (this interface operation is defined by the AXI Specification). The port csysack_n acknowledges the request to go into a low power state. The port csysack_n goes low for one cycle after csysreq_n goes low. The port cactive_n indicates whether the request is accepted or denied.

If the low power request is accepted, the awready_n and arready_n signals are driven low and they stay low for one cycle, after cactive_n goes high. The cactive_n output can also be used by an external low power controller to decide when to request a transition to the low power state for a port. When cactive_n is low, it indicates that there are no outstanding read or write commands from port n, there is no data in the Write Data Queue, and the port is idle.

In the low power state, the cactive_n output can be used by an external low power controller to trigger a low power state exit. Always, cactive_n is driven high when arvalid_n or awvalid_n is high, or if there are outstanding commands in the port. The paths from arvalid_n, awvalid_n, and wvalid_n to cactive_n are asynchronous. Therefore, cactive_n asserts, even if the clocks are removed. The low power controller should re-enable the clocks when cactive_n is driven high while in the low power state. The cactive_n signals are registered on aclk_n and input to the DDRC (cactive_in_ddrc).

Removal of AXI Clock through AXI Hardware Low Power Interface Handshaking

Removal of clocks using the AXI hardware low power interface is preferred, as this method places the ports in a low power state. awready_n and arready_n signals are driven low once the low power request is accepted. The low power interface of all the AXI ports is independent. aclk_n for a port can be removed once the low power requested has been accepted by the port.

Clock removal using the AXI hardware low power interface is shown in Figure 6.11-10. In this example, the system initiates the low power state exit though assertion of csysreq_n.

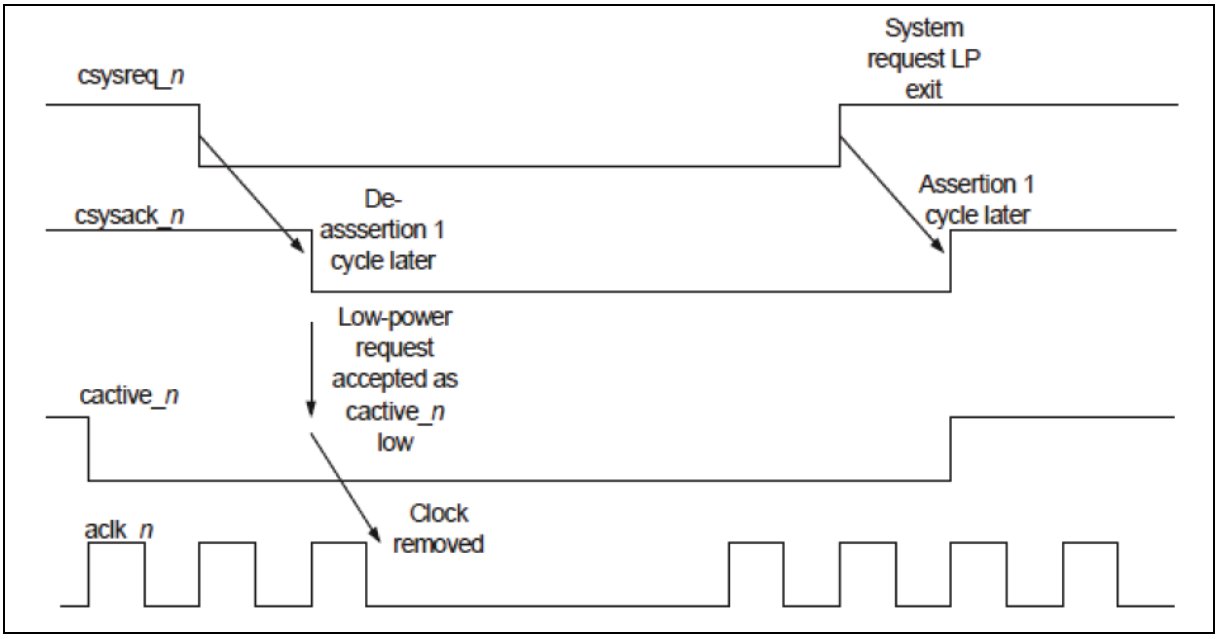


Figure 6.11-10 Clock Removal Using Hardware Low Power Interface, Exit Triggered by Assertion of csysreq_n

Clock removal using the AXI hardware low power interface is shown in Figure 6.11-11. In this example, the controller initiates the low power state exit though assertion of cactive_n.

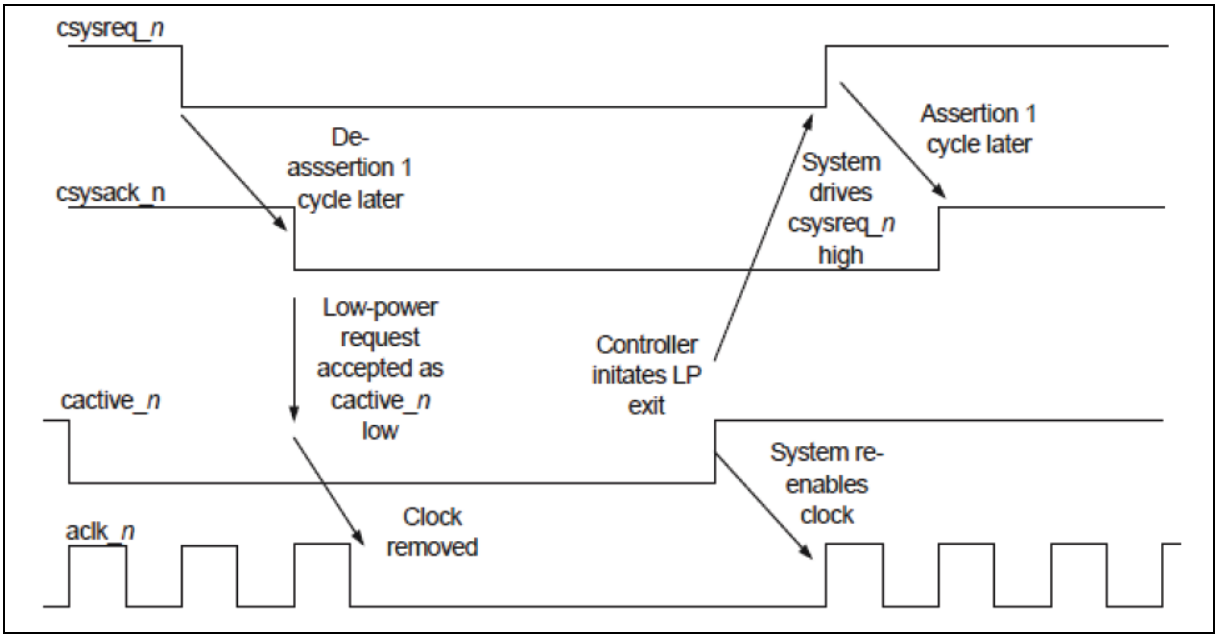


Figure 6.11-11 Clock Removal Using Hardware Low Power Interface, Exit Triggered by Assertion of cactive_n

Rejection of the request is shown in Figure 6.11-12. In this case, aclk_n cannot be removed.

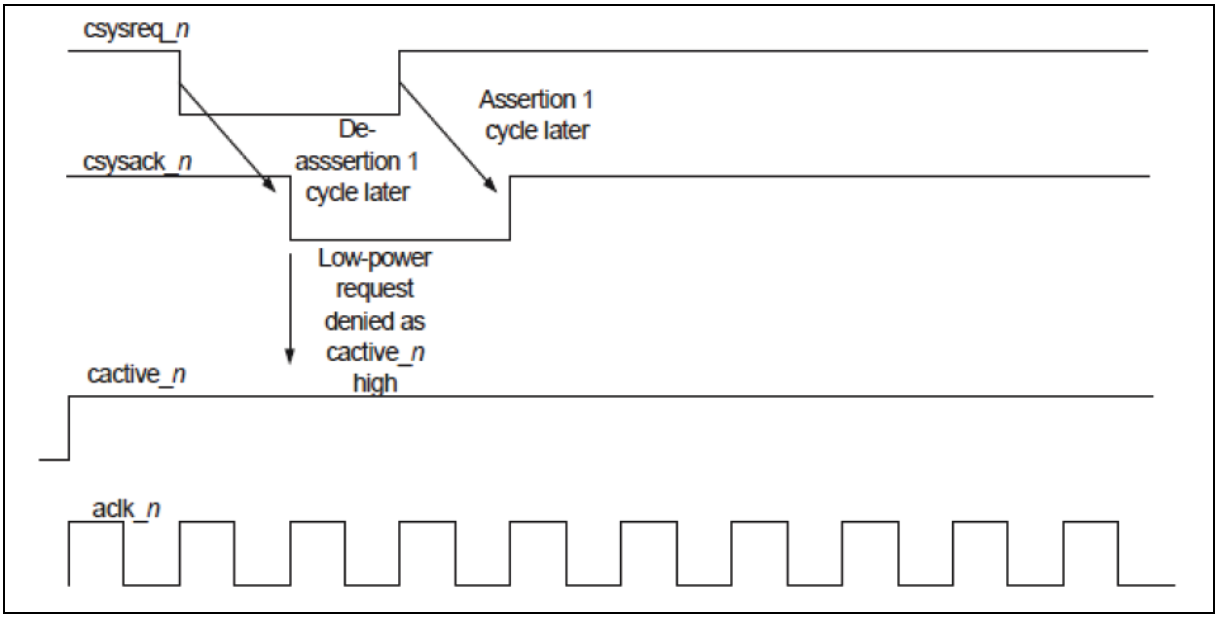


Figure 6.11-12 Rejection of Low Power State Request

With respect to interaction with “DDRC Hardware Low Power Interface”, it is recommended to stagger a low power entry of the DDRC until after all port's AXI Low Power Interface have successfully entered low power. This is to allow the cactive_n to propagate to cactive_ddrc_in of DDRC Hardware Low Power Interface.

Software Sequence for Removal of Clocks

Software can be used to keep the SDRAM in self-refresh. The AXI and DDRC clocks can be removed when in self-refresh by following the sequence described in Table 6.11-2. Dynamic and quasi dynamic registers cannot be programmed if any of the clocks has been removed. Clocks must be turned back on

before starting the programming sequence. Also the clock gating logic must ensure there are no glitches on the clocks when there are removed/enabled.

Step	Description	Comment
1	Write 0 to PCTRL_n.port_en	Blocks AXI ports from taking anymore transactions
2	Poll PSTAT.rd_port_busy_n = 0 Poll PSTAT.wr_port_busy_n = 0	Waits until all AXI ports are idle
3	Write 1 to PWRCTL.selfref_sw	Causes the system to move to Self Refresh state
4	Poll STAT.selfref_type= 2'b10	Waits until Self Refresh state is entered
5	Remove AXI clocks	
6	Remove DDRC controller clock	

Table 6.11-2 Software Clock Removal Sequence

The clocks must be re-enabled by following the sequence described in Table 6.11-3.

Step	Description	Comment
1	Enable AXI clocks	
2	Enable DDRC controller clock	
3	Write 0 to PWRCTL.selfref_sw	Cause system to exit from self-refresh state
4	Poll STAT.selfref_type = 2'b00	Wait until self-refresh state is exited
5	Write 1 to PCTRL_n.port_en	AXI ports are no longer blocked from taking transactions

Table 6.11-3 Re-enabling the Clocks

6.11.6.21 Address Mapper

Read and write requests are provided to the uMCTL2 with a system address. The system address is the command address of a transaction as presented on one of the data ports. The uMCTL2 converts this system address to a physical address. It maps the system address to the SDRAM rank, bank, row, and column addresses.

If the system address regions are enabled, the first part of the mapping is conversion of system address to AXI byte address. The controller maps disjoint address regions to internal consecutive addresses. Otherwise, the controller assumes that the DRAM is always mapped as a monolithic block. The second part of this mapping is conversion of AXI byte address to HIF word address. This is performed in the XPI block, and is applicable to AXI configurations only. The last part is the conversion of HIF word address to SDRAM address. A flexible address mapper maps the HIF word address to the SDRAM rank/bank/bank group/row/ column address. This address mapper is located within the DDRC. The address mapping to be used depends on the use-case. The uMCTL2 provides a set of registers that allows flexible re-programming of logical to physical address mapping.

System Address Regions

The system address regions add the capability to define up to four disjoint memory regions mapping to

the DRAM as consecutive addresses. The number of regions and minimum block size are determined by the hardware parameters UMCTL2_A_NSAR and UMCTL2_SARMINSIZE. Each region is defined by:

- Base Address: Starting address of the region aligned to the minimum block size.
- Number of Blocks: Size of the region in multiples of minimum block size.

Note: In our chip, hardware parameters UMCTL2_A_NSAR is set to 1, and hardware parameters UMCTL2_SARMINSIZE is set to 256 Mbytes.

The base address of each region is specified by the register SARBASEn.base_addr and the total number of blocks is specified by the register SARSIZEn.nblocks. System address region specification is the same for all ports. Error response is not generated by the controller for addresses falling outside the specified address regions and the same address translation is applied from one base address to the next base address.

HIF Address to SDRAM Address Mapping

The address mapper maps HIF word addresses to SDRAM addresses by selecting the HIF address bit that maps to each and every applicable SDRAM address bit. While it is possible to map HIF address bits to a SDRAM address in any desired manner, the full available address space is accessible only when no two SDRAM address bits are determined by the same HIF address bit. Each SDRAM address bit has an associated register vector to determine its source. Registers ADDRMAPx (x = 0 to 11) are used to program the address mapper.

The HIF address bit number is determined by adding the internal base of the ADDRMAPx (x = 0 to 11) register to the programmed value for that register, as described in the following equation:

HIF address bit number = [internal base] + [register value]

For example, for ADDRMAP3.addrmap_col_b7, the internal base is 7. When full data bus is in use, column bit 7 is determined by the following equation:

7 + [register value]

If this register is programmed to 2, the HIF address bit can be calculated by using the following equation:

[HIF address bit number] = 7 + 2 = 9

In other words, the column address bit 7 sent to SDRAM would always be equal to hif_cmd_addr[9] of the corresponding HIF source address.

Table 6.11-4 shows some examples of address mapping schemes which might be used. This example assumes a full bus width, using an SDRAM with 10 column address bits (denoted “c” in the diagram), 3 bank address bits (denoted “b”) and 12 row address bits (denoted “r”). The non-interleaved example shows a simple address map where the lowest HIF address bits map to the SDRAM column addresses, the next three to the bank addresses, and the highest HIF address bits map to the row addresses.

	HIF Address Bits	**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**
	MEMC_BURST_LENGTH																									
Non-interleaved	8	r11	r10	r9	r8	r7	r6	r5	r4	r3	r2	r1	r0	b2	b1	b0	c9	c8	c7	c6	c5	c4	c3	c2	c1	c0
	4	r11	r10	r9	r8	r7	r6	r5	r4	r3	r2	r1	r0	b2	b1	b0	c9	c8	c7	c6	c5	c4	c3	c2	c1	c0
Non-interleaved	8	r11	r10	r9	r8	r7	r6	r5	r4	r3	r2	r1	r0	b2	b1	b0	c9	c8	c7	c6	c5	c4	c3	c2	c1	c0

Interleaved-example 1	8	r11	r10	r9	r8	r7	r6	r5	r4	r3	r2	r1	r0	c9	c8	c7	c6	c5	c4	b2	b1	b0	c3	c2	c1	c0
	4	r11	r10	r9	r8	r7	r6	r5	r4	r3	r2	r1	r0	c9	c8	c7	c6	c5	c4	b2	b1	b0	c3	c2	c1	c0
Interleaved-example 1	8	r11	r10	r9	r8	r7	r6	r5	r4	r3	r2	r1	r0	c9	c8	c7	c6	c5	c4	b2	b1	b0	c3	c2	c1	c0
Interleaved-example 2	8	r11	r10	r9	r8	r7	r6	r5	r4	r3	r2	c6	c7	b2	c8	c9	r0	r1	b1	c3	c5	c4	b0	c2	c1	c0
Interleaved-example 3	4	r11	r10	r9	r8	r7	r6	r5	r4	r3	r2	c9	r0	r1	b0	c7	c8	c5	b1	c6	b2	c4	c3	c2	c1	c0

Table 6.11-4 Logical to Physical Address Mapping (Examples for DDR3)

6.11.6.22 Refresh Controls

Refresh can be issued using the auto-refresh feature in the uMCTL2 or using the direct software request of the refresh command. The RFSHCTL3.dis_auto_refresh register bit selects the refresh method. When this bit is set to 1, uMCTL2 uses direct software request of refresh command. When it is set to 0, the internal auto-refresh feature is used.

The purpose of refresh control is to:

(1) Reduce the bandwidth impact of refresh cycles

(2) Increase the likelihood of refreshes being serviced during idle periods

(3) Provide fine-gain control of the trading-off the previous benefits (to gather refreshes) versus the increased worst case latencies associated with gathering refreshes

(4) Allow traffic to flow to other ranks while a given rank is being refreshed (for multi-rank configurations of the uMCTL2 only)

Refresh Using Direct Software Request of Refresh Command

(1) Set the RFSHCTL3.dis_auto_refresh bit to 1. When the register bit set, the uMCTL2 checks for any pending refreshes. Any pending refreshes are issued right away using the 'critical refresh' feature inside the uMCTL2. After these refreshes are issued, all the refresh timers inside the uMCTL2 are reset to 0. They are re-activated only when the auto-refresh feature is enabled.

(2) The SoC must keep track of the refresh requirements of the SDRAM.

(3) The refresh command can be issued by setting the register bits DBGCMD.rank*_refresh to 1. When the rank*_refresh request is stored in the uMCTL2, the corresponding register bit is automatically cleared. The SoC can initiate a rank*_refresh operation only if DBGSTAT.rank*_refresh_busy is low. The uMCTL2 issues refresh to the SDRAM at the earliest.

(4) Software-driven refresh commands for each rank are loaded into a 9-entry buffer, and are issued by the uMCTL2 on the DFI as soon as it is legal to do so (the uMCTL2 controller must wait tRFC(min) between each refresh request). If the buffer saturates, the DBGSTAT.rank*_refresh_busy remains asserted to prevent software from initiating further refreshes.

Refresh Using Auto Refresh Feature Inside the uMCTL2

The uMCTL2 provides advanced refresh controls. Besides fully-configurable refresh constraints (tRFC(min) and tREFI), the uMCTL2 can also be programmed to gather refreshes to each rank of

SDRAM to reduce the bandwidth consumed by refreshes and to increase the likelihood that refreshes can be serviced during an idle period.

Fine-grain control of the refreshes ensures these benefits can be balanced against worst case latencies associated with servicing refreshes together. Staggered refresh timers for multi-rank configurations of the uMCTL2 allow transactions to continue to other ranks while refreshes are taking place to just one rank. To minimize the worst case impact of a forced refresh cycle, the uMCTL2 can be programmed to issue single refreshes at a time by forcing RFSHCTL0.refresh_burst = 0. For DDR2 and DDR3 modes, it can be programmed to burst up to 8 refreshes (RFSHCTL0.refresh_burst = 7).

Single Refresh

When using single refresh (RFSHCTL0.refresh_burst = 0), the uMCTL2 issues refreshes every time the refresh timer (tREFI) expires. This is the optimal mode of operation for systems that minimize the maximum latency associated with refresh cycles.

Burst Refresh

When burst refresh is enabled (RFSHCTL0.refresh_burst > 0), the uMCTL2 issues refreshes in bursts of (RFSHCTL0.refresh_burst+1) refreshes at one time. Bursting refreshes reduces the total latency associated with those refreshes by reducing the number of pre-charges and activates required for refresh, as banks must be precharged only once to perform the entire group of refreshes, instead of once for each refresh.

Speculative Refresh

When tREFI has expired at least once, the uMCTL2 can also perform speculative refreshes. This is done by automatically inserting refreshes when there are no transactions pending in the CAM to a rank/bank address.

The RFSHCTL0.refresh_to_x1_x32 register determines how long the transactions must not be stored in the CAM before considering inserting these speculative refreshes. Each time a speculative refresh is performed, the count of tREFI expirations is decremented, and thereby increasing the time before a critical refresh is required. This also ensures that speculative refreshes never occur more often than is required to keep the SDRAM properly refreshed.

If a new read or write transaction is accepted by the uMCTL2 during speculative refresh, the uMCTL2 services it as soon as legally possible. Most often it entails waiting for the required NOP cycles after a refresh before performing an activate and then servicing the read or write. If the uMCTL2 has begun closing pages for a speculative refresh but has not yet issued the refresh when the new transaction arrives, the speculative refresh is canceled.

6.11.6.23 ZQ Calibration

This feature is applicable to DDR3. The uMCTL2 controller uses ZQ calibration command to calibrate SDRAM RON (Resistor ON) and ODT (On-Die Termination) values over PVT (Process, Voltage, and Temperature). DDR3 SDRAMs need more time to calibrate RON and ODT at initialization and relatively less time to perform periodic calibrations.

To perform the initial calibration during the power-up initialization sequence. This command is allowed a timing period of tZQinit determined by INIT5. dev_zqinit_x32 to perform full calibration and the transfer of values.

To perform long ZQ Calibration after exiting from self-refresh mode. This command is allowed a timing period of tZQOPER, determined by ZQCTL0.t_zq_long_nop. This command is automatically issued when ZQCTL0.dis_srx_zqcl is set to 0. To disable issuing ZQCL after self-refresh exit, set ZQCTL0.dis_srx_zqcl to 1.

ZQCS (ZQ Calibration Short) command is used to perform periodic calibration to account for VT (Voltage/Temperature) variations. A shorter timing window is provided to perform calibration and transfer of values as defined by the timing parameter tZQCS (determined by QCTL0.t_zq_short_nop). ZQCS can be performed automatically on a regular interval or through direct software request.

In DDR3 mode, the ZQ Calibration commands are sent out encoded on the DFI command bus as mentioned in the JEDEC Specification. The uMCTL2 performs no other activities for the duration of tZQinit, tZQOPER and tZQCS. The quiet time on the SDRAM channel helps in accurate calibration of SDRAM RON and ODT. All banks are precharged and tRP met before the uMCTL2 issues the ZQ Calibration commands.

6.11.7 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
UMCTL2 Base Address: **UMCTL2_BA = 0x404d_0000**				
MSTR	UMCTL2_BA+0x000	R/W	Master Register0	0x0304_0001
STAT	UMCTL2_BA+0x004	R	Operating Mode Status Register	0x0000_0000
MRCTRL0	UMCTL2_BA+0x010	R/W	Mode Register Read/Write Control Register 0	0x0000_0030
MRCTRL1	UMCTL2_BA+0x014	R/W	Mode Register Read/Write Control Register 1	0x0000_0000
MRSTAT	UMCTL2_BA+0x018	R	Mode Register Read/Write Status Register	0x0000_0000
PWRCTL	UMCTL2_BA+0x030	R/W	Low Power Control Register	0x0000_0000
PWRTMG	UMCTL2_BA+0x034	R/W	Low Power Timing Register	0x0040_0010
HWLPCTL	UMCTL2_BA+0x038	R/W	Hardware Low Power Control Register	0x0000_0003
RFSHCTL0	UMCTL2_BA+0x050	R/W	Refresh Control Register 0	0x0021_0000
RFSHCTL1	UMCTL2_BA+0x054	R/W	Refresh Control Register 1	0x0000_0000
RFSHCTL3	UMCTL2_BA+0x060	R/W	Refresh Control Register 3	0x0000_0000
RFSHTMG	UMCTL2_BA+0x064	R/W	Refresh Timing Register	0x0062_008c
CRCPARCTL0	UMCTL2_BA+0x0c0	R/W	CRC Parity Control Register0	0x0000_0000
CRCPARSTAT	UMCTL2_BA+0x0cc	R	CRC Parity Status Register	0x0000_0000
INIT0	UMCTL2_BA+0x0d0	R/W	SDRAM Initialization Register 0	0x0002_004e
INIT1	UMCTL2_BA+0x0d4	R/W	SDRAM Initialization Register 1	0x0000_0000
INIT3	UMCTL2_BA+0x0dc	R/W	SDRAM Initialization Register 3	0x0000_0510
INIT4	UMCTL2_BA+0x0e0	R/W	SDRAM Initialization Register 4	0x0000_0000
INIT5	UMCTL2_BA+0x0e4	R/W	SDRAM Initialization Register 5	0x0010_0000
DIMMCTL	UMCTL2_BA+0x0f0	R/W	DIMM Control Register	0x0000_0000
RANKCTL	UMCTL2_BA+0x0f4	R/W	Rank Control Register	0x0000_066f
DRAMTMG0	UMCTL2_BA+0x100	R/W	SDRAM Timing Register 0	0x0f10_1b0f
DRAMTMG1	UMCTL2_BA+0x104	R/W	SDRAM Timing Register 1	0x0008_0414
DRAMTMG2	UMCTL2_BA+0x108	R/W	SDRAM Timing Register 2	0x0000_060d
DRAMTMG3	UMCTL2_BA+0x10c	R/W	SDRAM Timing Register 3	0x0000_400c
DRAMTMG4	UMCTL2_BA+0x110	R/W	SDRAM Timing Register 4	0x0504_0405
DRAMTMG5	UMCTL2_BA+0x114	R/W	SDRAM Timing Register 5	0x0505_0403

DRAMTMG8	UMCTL2_BA+0x120	R/W	SDRAM Timing Register 8	0x0000_4405
DRAMTMG15	UMCTL2_BA+0x13c	R/W	SDRAM Timing Register 15	0x0000_0000
ZQCTL0	UMCTL2_BA+0x180	R/W	ZQ Control Register 0	0x0200_0040
ZQCTL1	UMCTL2_BA+0x184	R/W	ZQ Control Register 1	0x0000_0100
DFITMG0	UMCTL2_BA+0x190	R/W	DFI Timing Register 0	0x0702_0002
DFITMG1	UMCTL2_BA+0x194	R/W	DFI Timing Register 1	0x0000_0404
DFILPCFG0	UMCTL2_BA+0x198	R/W	DFI Low Power Configuration Register 0	0x0700_0000
DFIUPD0	UMCTL2_BA+0x1a0	R/W	DFI Update Register 0	0x0040_0003
DFIUPD1	UMCTL2_BA+0x1a4	R/W	DFI Update Register 1	0x0001_0001
DFIUPD2	UMCTL2_BA+0x1a8	R/W	DFI Update Register 2	0x8000_0000
DFIMISC	UMCTL2_BA+0x1b0	R/W	DFI Miscellaneous Control Register	0x0000_0001
DFISTAT	UMCTL2_BA+0x1bc	R	DFI Status Register	0x0000_0000
DFIPHYMSTR	UMCTL2_BA+0x1c4	R/W	DFI PHY Master	0x0000_0001
ADDRMAP0	UMCTL2_BA+0x200	R/W	Address Map Register 0	0x0000_0000
ADDRMAP1	UMCTL2_BA+0x204	R/W	Address Map Register 1	0x0000_0000
ADDRMAP2	UMCTL2_BA+0x208	R/W	Address Map Register 2	0x0000_0000
ADDRMAP3	UMCTL2_BA+0x20c	R/W	Address Map Register 3	0x0000_0000
ADDRMAP4	UMCTL2_BA+0x210	R/W	Address Map Register 4	0x0000_0000
ADDRMAP5	UMCTL2_BA+0x214	R/W	Address Map Register 5	0x0000_0000
ADDRMAP6	UMCTL2_BA+0x218	R/W	Address Map Register 6	0x0000_0000
ADDRMAP9	UMCTL2_BA+0x224	R/W	Address Map Register 9	0x0000_0000
ADDRMAP10	UMCTL2_BA+0x228	R/W	Address Map Register 10	0x0000_0000
ADDRMAP11	UMCTL2_BA+0x22c	R/W	Address Map Register 11	0x0000_0000
ODTCFG	UMCTL2_BA+0x240	R/W	ODT Configuration Register	0x0400_0400
ODTMAP	UMCTL2_BA+0x244	R/W	ODT/Rank Map Register	0x0000_2211
SCHED	UMCTL2_BA+0x250	R/W	Scheduler Control Register	0x0000_1004
SCHED1	UMCTL2_BA+0x254	R/W	Scheduler Control Register 1	0x0000_0000
PERFHPR1	UMCTL2_BA+0x25c	R/W	High Priority Read CAM Register 1	0x0f00_0001
PERFLPR1	UMCTL2_BA+0x264	R/W	Low Priority Read CAM Register 1	0x0f00_007f
PERFWR1	UMCTL2_BA+0x26c	R/W	Write CAM Register 1	0x0f00_007f
DBG0	UMCTL2_BA+0x300	R/W	Debug Register 0	0x0000_0000

DBG1	UMCTL2_BA+0x304	R/W	Debug Register 1	0x0000_0000
DBGCAM	UMCTL2_BA+0x308	R	CAM Debug Register	0x3600_0000
DBGCMD	UMCTL2_BA+0x30c	R/W	Command Debug Register	0x0000_0000
DBGSTAT	UMCTL2_BA+0x310	R	Status Debug Register	0x0000_0000
SWCTL	UMCTL2_BA+0x320	R/W	Software Register Programming Control Enable	0x0000_0001
SWSTAT	UMCTL2_BA+0x324	R	Software Register Programming Control Status	0x0000_0001
SWCTLSTATIC	UMCTL2_BA+0x328	R/W	Static Registers Write Enable	0x0000_0000
POISONCFG	UMCTL2_BA+0x36c	R/W	AXI Poison Configuration Register. Common for all AXI ports.	0x0011_0011
POISONSTAT	UMCTL2_BA+0x370	R	AXI Poison Status Register	0x0000_0000
PSTAT	UMCTL2_BA+0x3fc	R	Port Status Register	0x0000_0000
PCCFG	UMCTL2_BA+0x400	R/W	Port Common Configuration Register	0x0000_0000
PCFGR_0	UMCTL2_BA+0x404	R/W	Port 0 Configuration Read Register	0x0000_4000
PCFGW_0	UMCTL2_BA+0x408	R/W	Port 0 Configuration Write Register	0x0000_4000
PCTRL_0	UMCTL2_BA+0x490	R/W	Port 0 Control Register	0x0000_0000
PCFGQOS0_0	UMCTL2_BA+0x494	R/W	Port 0 Read QoS Configuration Register 0	0x0000_0000
PCFGR_1	UMCTL2_BA+0x4b4	R/W	Port 1 Configuration Read Register	0x0000_4000
PCFGW_1	UMCTL2_BA+0x4b8	R/W	Port 1 Configuration Write Register	0x0000_4000
PCTRL_1	UMCTL2_BA+0x540	R/W	Port 1 Control Register	0x0000_0000
PCFGQOS0_1	UMCTL2_BA+0x544	R/W	Port 1 Read QoS Configuration Register 0	0x0000_0000
PCFGR_2	UMCTL2_BA+0x564	R/W	Port 2 Configuration Read Register	0x0000_4000
PCFGW_2	UMCTL2_BA+0x568	R/W	Port 2 Configuration Write Register	0x0000_4000
PCTRL_2	UMCTL2_BA+0x5f0	R/W	Port 2 Control Register	0x0000_0000
PCFGQOS0_2	UMCTL2_BA+0x5f4	R/W	Port 2 Read QoS Configuration Register 0	0x0000_0000
PCFGR_3	UMCTL2_BA+0x614	R/W	Port 3 Configuration Read Register	0x0000_4000
PCFGW_3	UMCTL2_BA+0x618	R/W	Port 3 Configuration Write Register	0x0000_4000
PCTRL_3	UMCTL2_BA+0x6a0	R/W	Port 3 Control Register	0x0000_0000
PCFGQOS0_3	UMCTL2_BA+0x6a4	R/W	Port 3 Read QoS Configuration Register 0	0x0000_0000
PCFGR_4	UMCTL2_BA+0x6c4	R/W	Port 4 Configuration Read Register	0x0000_4000
PCFGW_4	UMCTL2_BA+0x6c8	R/W	Port 4 Configuration Write Register	0x0000_4000
PCTRL_4	UMCTL2_BA+0x750	R/W	Port 4 Control Register	0x0000_0000

PCFGQOS0_4	UMCTL2_BA+0x754	R/W	Port 4 Read QoS Configuration Register 0	0x0000_0000
PCFGR_5	UMCTL2_BA+0x774	R/W	Port 5 Configuration Read Register	0x0000_4000
PCFGW_5	UMCTL2_BA+0x778	R/W	Port 5 Configuration Write Register	0x0000_4000
PCTRL_5	UMCTL2_BA+0x800	R/W	Port 5 Control Register	0x0000_0000
PCFGQOS0_5	UMCTL2_BA+0x804	R/W	Port 5 Read QoS Configuration Register 0	0x0000_0000
PCFGR_6	UMCTL2_BA+0x824	R/W	Port 6 Configuration Read Register	0x0000_4000
PCFGW_6	UMCTL2_BA+0x828	R/W	Port 6 Configuration Write Register	0x0000_4000
PCTRL_6	UMCTL2_BA+0x8b0	R/W	Port 6 Control Register	0x0000_0000
PCFGQOS0_6	UMCTL2_BA+0x8b4	R/W	Port 6 Read QoS Configuration Register 0	0x0000_0000
SARBASE0	UMCTL2_BA+0xf04	R/W	SAR Base Address Register n	0x0000_0000
SARSIZE0	UMCTL2_BA+0xf08	R/W	SAR Size Register n	0x0000_0000
UMCTL2_VER_NUMBER	UMCTL2_BA+0xff0	R	UMCTL2 Version Number Register	0x3338_302a
UMCTL2_VER_TYPE	UMCTL2_BA+0xff4	R	UMCTL2 Version Type Register	0x6761_2a2a

6.11.8 Register Description

UMCTL2 Master Register0 (MSTR)

Register	Offset	R/W	Description	Reset Value
MSTR	UMCTL2_BA+0x000	R/W	Master Register0	0x0304_0001

31	30	29	28	27	26	25	24
Reserved						active_ranks	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				burst_rdwr			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
dll_off_mode	Reserved	data_bus_width		Reserved	en_2t_timing_mode	burstchop	Reserved
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							ddr3

Bits	Description	
[31:26]	**Reserved**	Reserved.
[25:24]	**active_ranks**	**Active Ranks** 01 = One rank. 11 = Two ranks. Others = Reserved. **Programming Mode:** Static
[23:20]	**Reserved**	Reserved.
[19:16]	**burst_rdwr**	**Burst Length for Read and Write** Indicates SDRAM burst length used: 0010 = Burst length of 4. 0100 = Burst length of 8. All other values are reserved. This bit controls the burst size used to access the SDRAM. This must match the burst length mode register setting in the SDRAM. (For BC4/8 on-the-fly mode of DDR3, set this field to 0x0100) Burst length of 2 is not supported with AXI ports when MEMC_BURST_LENGTH is 8. Burst length of 2 is only supported when the controller is operating in 1:1 frequency mode. For DDR3, this must be set to 0x0100 (BL8). **Programming Mode:** Static
[15]	**dll_off_mode**	**DLL Off Mode Enable Bit** 1 = When the uMCTL2 and DRAM has to be put in DLL-off mode for low frequency operation. 0 = To put uMCTL2 and DRAM in DLL-on mode for normal frequency operation. **Programming Mode:** Quasi-dynamic Group 2
[14]	**Reserved**	Reserved.
[13:12]	**data_bus_width**	**DQ Bus Width** Selects proportion of DQ bus width that is used by the SDRAM.

		00 = Full DQ bus width to SDRAM. 01 = Half DQ bus width to SDRAM. 10 = Reserved. 11 = Reserved. Note that half bus width mode is only supported when the SDRAM bus width is a multiple of 16. Bus width refers to DQ bus width (excluding any ECC width). **Programming Mode:** Static
[11]	**Reserved**	Reserved.
[10]	**en_2t_timing_mode**	**2T Timing Mode Enable Bit** If 1, then uMCTL2 uses 2T timing, otherwise uses 1T timing. In 2T timing, all command signals (except chip select) are held for 2 clocks on the SDRAM bus. Chip select is asserted on the second cycle of the command. **Programming Mode:** Static
[9]	**burstchop**	**Burst-Chop Enable Bit** When this bit is set, enables burst-chop (BC4 or 8 on-the-fly) in DDR3. Burst Chop for reads is exercised only: • If in full bus width mode (MSTR.data_bus_width = 00). Burst Chop for writes is exercised only: • If CRC is disabled (CRCPARCTL1.crc_enable = 0). BC4 (fixed) mode is not supported. **Programming Mode:** Static
[8:1]	**Reserved**	Reserved.
[0]	**ddr3**	**DDR3 SDRAM Type Enable Bit** Selects DDR3 SDRAM. 1 = DDR3 SDRAM device in use. 0 = non-DDR3 SDRAM device in use. Present only in designs configured to support DDR3. **Programming Mode:** Static

UMCTL2 Operating Mode Status Register (STAT)

Register	Offset	R/W	Description	Reset Value
STAT	UMCTL2_BA+0x004	R	Operating Mode Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			selfref_cam_not_empty	Reserved			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		selfref_type		Reserved		operating_mode	

Bits	Description	
[31:13]	**Reserved**	Reserved.
[12]	**selfref_cam_not_empty**	**Self-Refresh with CAMs Not Empty** Set to 1 when self-refresh is entered but CAMs are not drained. Cleared after exiting self-refresh. **Programming Mode:** Static
[11:6]	**Reserved**	Reserved.
[5:4]	**selfref_type**	**Self-Refresh Type** Flags if self-refresh is entered, and if it is under automatic self-refresh control only or not. 00 = SDRAM is not in self-refresh. If retry is enabled by CRCPARCTL1.crc_parity_retry_enable, this also indicates that the SRE command is still in parity error window or retry is in-progress. 01 = SDRAM is in self-refresh, which is caused by PHY Master Request. 10 = SDRAM is in self-refresh, which was not caused solely under automatic self-refresh control. It could have been caused by Hardware Low Power Interface and/or Software (PWRCTL.selfref_sw). If retry is enabled, this ensures that the SRE command is executed correctly without parity error. 11 = SDRAM is in self-refresh, which was caused by Automatic self-refresh only. If retry is enabled, this ensures that the SRE command is executed correctly without parity error. **Programming Mode:** Static
[3:2]	**Reserved**	Reserved.
[1:0]	**operating_mode**	**Operating Mode** 00 = Init. 01 = Normal. 10 = Power-down. 11 = Self-refresh. **Programming Mode:** Static

UMCTL2 Mode Register Read/Write Control Register 0 (MRCTRL0)

Register	Offset	R/W	Description	Reset Value
MRCTRL0	UMCTL2_BA+0x010	R/W	Mode Register Read/Write Control Register 0	0x0000_0030

31	30	29	28	27	26	25	24
mr_wr	Reserved						
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
mr_addr				Reserved			
7	6	5	4	3	2	1	0
Reserved		mr_rank		Reserved			

Bits	Description	
[31]	**mr_wr**	**Mode Register Read or Write Operation Trigger** Setting this register bit to 1 triggers a mode register read or write operation. When the MR operation is complete, the uMCTL2 automatically clears this bit. The other fields of this register must be written in a separate APB transaction, before setting this mr_wr bit. It is recommended NOT to set this signal if in Init, Deep power-down, or MPSM operating modes. **Testable:** readOnly **Programming Mode:** Dynamic
[30:16]	**Reserved**	Reserved.
[15:12]	**mr_addr**	**Mode Register Address** Address of the mode register that is to be written to. 0000 = MR0. 0001 = MR1. 0010 = MR2. 0011 = MR3. 0100 = MR4. 0101 = MR5. 0110 = MR6. 0111 = MR7. **Programming Mode:** Dynamic
[11:6]	**Reserved**	Reserved.
[5:4]	**mr_rank**	**Mode Register Accessing Rank** Controls which rank is accessed by MRCTRL0.mr_wr. Normally, it is desired to access all ranks, so all bits must be set to 1. Examples (assuming uMCTL2 is configured for 2 ranks): 0x1 = Select rank 0 only. 0x2 = Select rank 1 only. 0x3 = Select ranks 0 and 1.

		Programming Mode: Dynamic
[3:0]	**Reserved**	Reserved.

UMCTL2 Mode Register Read/Write Control Register 1 (MRCTRL1)

Register	Offset	R/W	Description	Reset Value
MRCTRL1	UMCTL2_BA+0x014	R/W	Mode Register Read/Write Control Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
mr_data							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
mr_data							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**mr_data**	**Mode Register Write Data** Mode register write data for all DDR2/DDR3 modes. **Programming Mode:** Dynamic

UMCTL2 Mode Register Read/Write Status Register (MRSTAT)

Register	Offset	R/W	Description	Reset Value
MRSTAT	UMCTL2_BA+0x018	R	Mode Register Read/Write Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**mr_wr_busy**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**mr_wr_busy**	**Mode Register Write Busy** The SoC might initiate a MR write operation only if this signal is low. This signal goes: • High in the clock after the uMCTL2 accepts the MRW/MRR request • Low when the MRW/MRR command is issued to the SDRAM It is recommended not to perform MRW/MRR commands when 'MRSTAT.mr_wr_busy' is high. 0 = Indicates that the S oC can initiate a mode register write operation. 1 = Indicates that mode register write operation is in progress. **Programming Mode:** Dynamic

UMCTL2 Low Power Control Register (PWRCTL)

Register	Offset	R/W	Description	Reset Value
PWRCTL	UMCTL2_BA+0x030	R/W	Low Power Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
dis_cam_drain_selfref	**Reserved**	**selfref_sw**	**Reserved**	**en_dfi_dram_clk_disable**	**Reserved**	**powerdown_en**	**selfref_en**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**dis_cam_drain_selfref**	**Disable CAM Drain Before Entering Self-Refresh** Indicates whether skipping CAM draining is allowed when entering self-refresh. This register field cannot be modified while PWRCTL.selfref_sw == 1. 0 = CAMs must be empty before entering SR. 1 = CAMs are not emptied before entering SR (unsupported) Note, PWRCTL.dis_cam_drain_selfref=1 is unsupported in this release. PWRCTL.dis_cam_drain_selfref=0 is required. **Programming Mode:** Dynamic
[6]	**Reserved**	Reserved.
[5]	**selfref_sw**	**Self-Refresh Entry/Exit by Software** A value of 1 to this register causes system to move to self-refresh state immediately, as long as it is not in INIT or DPD/MPSM operating_mode. This is referred to as Software Entry/Exit to self-refresh. 1 = Software Entry to self-refresh. 0 = Software Exit from self-refresh. **Programming Mode:** Dynamic
[4]	**Reserved**	Reserved.
[3]	**en_dfi_dram_clk_disable**	**dfi_dram_clk_disable Enable Bit** Enables the assertion of dfi_dram_clk_disable whenever a clock is not required by the SDRAM. If set to 0, dfi_dram_clk_disable is never asserted. Assertion of dfi_dram_clk_disable is as follows: In DDR2/DDR3, can only be asserted in self-refresh. **Programming Mode:** Dynamic
[2]	**Reserved**	Reserved.
[1]	**powerdown_en**	**Power-Down Enable Bit**

		If it is true, then the uMCTL2 goes into power-down after a programmable number of cycles "maximum idle clocks before power down" (PWRTMG.powerdown_to_x32). This register bit may be re-programmed during the course of normal operation. **Programming Mode:** Dynamic
[0]	**selfref_en**	**Self-Refresh Enable Bit** If it is true, then the uMCTL2 puts the SDRAM into self-refresh after a programmable number of cycles "maximum idle clocks before self-refresh (PWRTMG.selfref_to_x32)". This register bit may be re-programmed during the course of normal operation. **Programming Mode:** Dynamic

UMCTL2 Low Power Timing Register (PWRTMG)

Register	Offset	R/W	Description	Reset Value
PWRTMG	UMCTL2_BA+0x034	R/W	Low Power Timing Register	0x0040_0010

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
selfref_to_x32							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			powerdown_to_x32				

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**selfref_to_x32**	**SDRAM Self-Refresh Enter After This Clock Number** After this many clocks of the DDRC command channel being idle the uMCTL2 automatically puts the SDRAM into self-refresh. The DDRC command channel is considered idle when there are no HIF commands outstanding. This must be enabled in the PWRCTL.selfref_en. FOR PERFORMANCE ONLY. Unit: Multiples of 32 DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 4
[15:5]	**Reserved**	Reserved.
[4:0]	**powerdown_to_x32**	**SDRAM Power-Down Enter After This Clock Number** After this many clocks of the DDRC command channel being idle the uMCTL2 automatically puts the SDRAM into power-down. The DDRC command channel is considered idle when there are no HIF commands outstanding. This must be enabled in the PWRCTL.powerdown_en. FOR PERFORMANCE ONLY. Unit: Multiples of 32 DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 4

UMCTL2 Hardware Low Power Control Register (HWLPCTL)

Register	Offset	R/W	Description	Reset Value
HWLPCTL	UMCTL2_BA+0x038	R/W	Hardware Low Power Control Register	0x0000_0003

31	30	29	28	27	26	25	24
Reserved				hw_lp_idle_x32			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
hw_lp_idle_x32							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						hw_lp_exit_idle_en	hw_lp_en

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:16]	**hw_lp_idle_x32**	**Hardware idle period** The cactive_ddrc output is driven low if the DDRC command channel is idle for hw_lp_idle * 32 cycles if not in INIT or DPD/MPSM operating_mode. The DDRC command channel is considered idle when there are no HIF commands outstanding. The hardware idle function is disabled when hw_lp_idle_x32=0. hw_lp_idle_x32=1 is an illegal value. FOR PERFORMANCE ONLY. Unit: Multiples of 32 DFI clock cycles. **Programming Mode:** Static
[15:2]	**Reserved**	Reserved.
[1]	**hw_lp_exit_idle_en**	**Hardware Exit Idle Enable Bit** When this bit is programmed to 1 the cactive_in_ddrc pin of the DDRC can be used to exit from the automatic clock stop, automatic power down or automatic self-refresh modes. Note, it does not cause exit of self-refresh that was caused by Hardware Low Power Interface and/or Software (PWRCTL.selfref_sw). **Programming Mode:** Static
[0]	**hw_lp_en**	**Hardware Low Power Interface Enable Bit** **Programming Mode:** Quasi-dynamic Group 2

UMCTL2 Refresh Control Register 0 (RFSHCTL0)

Register	Offset	R/W	Description	Reset Value
RFSHCTL0	UMCTL2_BA+0x050	R/W	Refresh Control Register 0	0x0021_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
refresh_margin				**Reserved**			**refresh_to_x1_x32**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
refresh_to_x1_x32				**Reserved**		**refresh_burst**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
refresh_burst				**Reserved**			

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:20]	**refresh_margin**	**Refresh Margin** Threshold value in number of DFI clock cycles before the critical refresh or page timer expires. A critical refresh is to be issued before this threshold is reached. It is recommended that this not be changed from the default value, currently shown as 0x2. It must always be less than internally used t_rfc_nom/32. Note that internally used t_rfc_nom is equal to RFSHTMG.t_rfc_nom_x1_x32 * 32 if RFSHTMG.t_rfc_nom_x1_sel=0. Unit: Multiples of 32 DFI clock cycles. **Programming Mode:** Dynamic = Refresh Related.
[19:17]	**Reserved**	Reserved.
[16:12]	**refresh_to_x1_x32**	**SDRAM Bus Idle Period** If the refresh timer (tRFCnom, also known as tREFI) has expired at least once, then a speculative refresh may be performed. A speculative refresh is a refresh performed at a time when refresh would be useful. When the SDRAM bus is idle for a period of time determined by this RFSHCTL0.refresh_to_x1_x32 and the refresh timer has expired at least once since the last refresh, then a speculative refresh is performed. Speculative refreshes continue successively until there are no refreshes pending or until new reads or writes are issued to the uMCTL2. FOR PERFORMANCE ONLY. Unit: DFI clock cycles or multiples of 32 DFI clock cycles, depending on RFSHTMG.t_rfc_nom_x1_sel. **Programming Mode:** Dynamic = Refresh Related.
[11:10]	**Reserved**	Reserved.
[9:4]	**refresh_burst**	**Refresh Burst Number** The programmed value + 1 is the number of refresh timeouts that is allowed to accumulate before traffic is blocked and the refreshes are forced to execute.

		Closing pages to perform a refresh is a one-time penalty that must be paid for each group of refreshes. Therefore, performing refreshes in a burst reduces the per-refresh penalty of these page closings. Higher numbers for RFSHCTL.refresh_burst slightly increase utilization; lower numbers decrease the worst-case latency associated with refreshes. 0 = Single refresh. 1 = Burst-of-2 refresh. 7 = Burst-of-8 refresh. For more information on burst refresh feature, see section 3.9 of DDR2 JEDEC specification - JESD79-2F.pdf. For DDR2/3, the refresh is always per-rank and not per-bank. The rank refresh can be accumulated over 8*tREFI cycles using the burst refresh feature. If using PHY-initiated updates, care must be taken in the setting of RFSHCTL0.refresh_burst, to ensure that tRFCmax is not violated due to a PHY-initiated update occurring shortly before a refresh burst is due. In this situation, the refresh burst is delayed until the PHY-initiated update is complete. **Programming Mode:** Dynamic - Refresh Related
[3:0]	**Reserved**	Reserved.

UMCTL2 Refresh Control Register 1 (RFSHCTL1)

Register	Offset	R/W	Description	Reset Value
RFSHCTL1	UMCTL2_BA+0x054	R/W	Refresh Control Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				**refresh_timer1_start_value_x32**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
refresh_timer1_start_value_x32							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**refresh_timer0_start_value_x32**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
refresh_timer0_start_value_x32							

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:16]	**refresh_timer1_start_value_x32**	**Refresh Timer Start Value for Rank 1** only present in multi-rank configurations This is useful in staggering the refreshes to multiple ranks to help traffic to proceed. FOR PERFORMANCE ONLY. Unit: Multiples of 32 DFI clock cycles. **Programming Mode:** Dynamic - Refresh Related
[15:12]	**Reserved**	Reserved.
[11:0]	**refresh_timer0_start_value_x32**	**Refresh Timer Start Value for Rank 0** only present in multi-rank configurations This is useful in staggering the refreshes to multiple ranks to help traffic to proceed. FOR PERFORMANCE ONLY. Unit: Multiples of 32 DFI clock cycles. **Programming Mode:** Dynamic - Refresh Related

UMCTL2 Refresh Control Register 3 (RFSHCTL3)

Register	Offset	R/W	Description	Reset Value
RFSHCTL3	UMCTL2_BA+0x060	R/W	Refresh Control Register 3	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**refresh_update_level**	**dis_auto_refresh**

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**refresh_update_level**	**Refresh Registers Updated Operation** Toggle this signal (either from 0 to 1 or from 1 to 0) to indicate that the refresh registers have been updated. refresh_update_level must not be toggled when the DDRC is in reset (core_ddrc_rstn = 0). The refresh registers are automatically updated when exiting reset. **Programming Mode:** Dynamic
[0]	**dis_auto_refresh**	**Disable Auto-Refresh Enable Bit** When '1', disable auto-refresh generated by the uMCTL2. When auto-refresh is disabled, the SoC must generate refreshes using the registers DBGCMD.rankn_refresh. When dis_auto_refresh transitions from 0 to 1, any pending refreshes are immediately scheduled by the uMCTL2. This register field is changeable on the fly. **Programming Mode:** Dynamic - Refresh Related

UMCTL2 Refresh Timing Register (RFSHTMG)

Register	Offset	R/W	Description	Reset Value
RFSHTMG	UMCTL2_BA+0x064	R/W	Refresh Timing Register	0x0062_008c

31	30	29	28	27	26	25	24
Reserved				t_rfc_nom_x1_x32			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
t_rfc_nom_x1_x32							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						t_rfc_min	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
t_rfc_min							

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:16]	**t_rfc_nom_x1_x32**	**Average Time Interval between Refreshes per Rank** (Specification: 7.8us for DDR2 and DDR3). When the controller is operating in 1:1 mode, set this register to RoundDown(tREFI/tCK) When the controller is operating in 1:2 mode, set this register to RoundDown(RoundDown(tREFI/tCK)/2) In both the previous cases, if RFSHTMG.t_rfc_nom_x1_sel = 0, divide the previous result by 32 and round down. **Note:** • RFSHTMG.t_rfc_nom_x1_x32 must be greater than 0x1 • If RFSHTMG.t_rfc_nom_x1_sel == 1, RFSHTMG.t_rfc_nom_x1_x32 must be greater than RFSHTMG.t_rfc_min. • If RFSHTMG.t_rfc_nom_x1_sel == 0, RFSHTMG.t_rfc_nom_x1_x32 * 32 must be greater than RFSHTMG.t_rfc_min. • In DDR2/DDR3: RFSHTMG.t_rfc_nom_x1_x32 must be less than or equal to 0xFFE Unit: DFI clock cycles or multiples of 32 DFI clock cycles, depending on RFSHTMG.t_rfc_nom_x1_sel. **Programming Mode:** Dynamic - Refresh Related
[15:10]	**Reserved**	Reserved.
[9:0]	**t_rfc_min**	**Minimum Time from Refresh to Refresh or Activate** tRFC (min) When the controller is operating in 1:1 mode, t_rfc_min must be set to RoundUp(tRFCmin/tCK). When the controller is operating in 1:2 mode, t_rfc_min must be set to RoundUp(RoundUp(tRFCmin/tCK)/2). Unit: DFI clock cycles. **Programming Mode:** Dynamic - Refresh Related

UMCTL2 CRC Parity Control Register0 (CRCPARCTL0)

Register	Offset	R/W	Description	Reset Value
CRCPARCTL0	UMCTL2_BA+0x0c0	R/W	CRC Parity Control Register0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					dfi_alert_err_cnt_clr	dfi_alert_err_int_clr	dfi_alert_err_int_en

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**dfi_alert_err_cnt_clr**	**DFI Alert Error Counter Clear Bit** Indicates the clear bit for DFI alert error counter. Asserting this bit clears the DFI alert error counter, CRCPARSTAT.dfi_alert_err_cnt. uMCTL2 automatically clears this bit. **Testable:** readOnly **Programming Mode:** Dynamic
[1]	**dfi_alert_err_int_clr**	**DFI Alert Error Interrupt Clear Bit** Interrupt clear bit for DFI alert error. If this bit is set, the alert error interrupt on CRCPARSTAT.dfi_alert_err_int is cleared. uMCTL2 automatically clears this bit. **Testable:** readOnly **Programming Mode:** Dynamic
[0]	**dfi_alert_err_int_en**	**DFI Alert Error Interrupt Enable Bit** If this bit is set, any parity/CRC error detected on the dfi_alert_n input results in an interrupt being set on CRCPARSTAT.dfi_alert_err_int. **Programming Mode:** Dynamic

UMCTL2 CRC Parity Status Register (CRCPARSTAT)

Register	Offset	R/W	Description	Reset Value
CRCPARSTAT	UMCTL2_BA+0x0cc	R	CRC Parity Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							**dfi_alert_err_int**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
dfi_alert_err_cnt							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
dfi_alert_err_cnt							

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**dfi_alert_err_int**	**DFI Alert Error Interrupt** If a parity/CRC error is detected on dfi_alert_n, and the interrupt is enabled by CRCPARCTL0.dfi_alert_err_int_en, this interrupt bit is set. It remains set until cleared by CRCPARCTL0.dfi_alert_err_int_clr. **Programming Mode:** Static
[15:0]	**dfi_alert_err_cnt**	**DFI Alert Error Count** If a parity/CRC error is detected on dfi_alert_n, this counter be incremented. This is independent of the setting of CRCPARCTL0.dfi_alert_err_int_en. It saturates at 0xFFFF, and can be cleared by asserting CRCPARCTL0.dfi_alert_err_cnt_clr. **Programming Mode:** Static

UMCTL2 SDRAM Initialization Register 0 (INIT0)

Register	Offset	R/W	Description	Reset Value
INIT0	UMCTL2_BA+0x0d0	R/W	SDRAM Initialization Register 0	0x0002_004e

31	30	29	28	27	26	25	24
skip_dram_init		Reserved				post_cke_x1024	
23	22	21	20	19	18	17	16
post_cke_x1024							
15	14	13	12	11	10	9	8
Reserved				pre_cke_x1024			
7	6	5	4	3	2	1	0
pre_cke_x1024							

Bits	Description	
[31:30]	**skip_dram_init**	**SDRAM Initialization Routine Skipped Enable Bit** If lower bit is enabled the SDRAM initialization routine is skipped. The upper bit decides what state the controller starts up in when reset is removed. 00 = SDRAM Initialization routine is run after power-up. 01 = SDRAM Initialization routine is skipped after power-up. The controller starts up in normal Mode 11 = SDRAM Initialization routine is skipped after power-up. The controller starts up in self-refresh Mode 10 = Reserved. **Programming Mode:** Quasi-dynamic Group 2
[29:26]	**Reserved**	Reserved.
[25:16]	**post_cke_x1024**	**Number of Cycles to Wait after Driving CKE High to Start the SDRAM Initialization** Indicates the number of cycles to wait after driving CKE high to start the SDRAM initialization sequence. DDR2 typically requires a 400 ns delay, requiring this value to be programmed to 2 at all clock speeds. When the controller is operating in 1:2 frequency ratio mode, program this to JEDEC spec value divided by 2, and round it up to the next integer value. Unit: Multiples of 1024 DFI clock cycles. **Programming Mode:** Static
[15:12]	**Reserved**	Reserved.
[11:0]	**pre_cke_x1024**	**Number of Cycles to Wait after Reset before Driving CKE High to Start SDRAM Initialization** Indicates the number of cycles to wait after reset before driving CKE high to start the SDRAM initialization sequence. DDR2 specifications typically require this to be programmed for a delay of >= 200 us. When the controller is operating in 1:2 frequency ratio mode, program this to JEDEC spec value divided by 2, and round it up to the next integer value. For DDR3 RDIMMs, this must include the time needed to satisfy tSTAB. Unit: Multiples of 1024 DFI clock cycles. **Programming Mode:** Static

UMCTL2 SDRAM Initialization Register 1 (INIT1)

Register	Offset	R/W	Description	Reset Value
INIT1	UMCTL2_BA+0x0d4	R/W	SDRAM Initialization Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							dram_rstn_x1024
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
dram_rstn_x1024							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				pre_ocd_x32			

Bits	Description	
[31:25]	**Reserved**	Reserved.
[24:16]	**dram_rstn_x1024**	**SDRAM Reset During Initialization sequence** Indicates the number of cycles to assert SDRAM reset signal during initialization sequence. This is only present for designs supporting DDR3 devices. For use with a Synopsys DDR PHY, this must be set to a minimum of 1. When the controller is operating in 1:2 frequency ratio mode, program this to JEDEC spec value divided by 2, and round it up to the next integer value. Unit: Multiples of 1024 DFI clock cycles. **Programming Mode:** Static
[15:4]	**Reserved**	Reserved.
[3:0]	**pre_ocd_x32**	**Wait Period before Driving OCD Complete Command to SDRAM** Indicates the wait period before driving the OCD complete command to SDRAM. There is no known specific requirement for this; it may be set to zero. Unit: Multiples of 32 DFI clock cycles. **Programming Mode:** Static

UMCTL2 SDRAM Initialization Register 3 (INIT3)

Register	Offset	R/W	Description	Reset Value
INIT3	UMCTL2_BA+0x0dc	R/W	SDRAM Initialization Register 3	0x0000_0510

31	30	29	28	27	26	25	24
mr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
mr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
emr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
emr							

Bits	Description	
[31:16]	**mr**	**Write Value to MR Register** DDR2:Indicates the value to write to MR register. Bit 8 is for DLL and the setting here is ignored. The uMCTL2 sets this bit appropriately. DDR3: Value loaded into MR0 register. **Programming Mode:** Quasi-dynamic Group 1, Group 4
[15:0]	**emr**	**Write Value to EMR Register** DDR2: Indicates the value to write to EMR register. Bits 9:7 are for OCD and the setting in this register is ignored. The uMCTL2 sets those bits appropriately. DDR3: Value to write to MR1 register Set bit 7 to 0. **Programming Mode:** Quasi-dynamic Group 4

UMCTL2 SDRAM Initialization Register 4 (INIT4)

Register	Offset	R/W	Description	Reset Value
INIT4	UMCTL2_BA+0x0e0	R/W	SDRAM Initialization Register 4	0x0000_0000

31	30	29	28	27	26	25	24
emr2							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
emr2							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
emr3							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
emr3							

Bits	Description	
[31:16]	**emr2**	**Write Value to EMR2 Register** DDR2: Indicates the value to write to EMR2 register. DDR3: Value to write to MR2 register. **Programming Mode:** Quasi-dynamic Group 4
[15:0]	**emr3**	**Write Value to EMR3 Register** DDR2: Indicates the value to write to EMR3 register. DDR3: Value to write to MR3 register. **Programming Mode:** Quasi-dynamic Group 2, Group 4

UMCTL2 SDRAM Initialization Register 5 (INIT5)

Register	Offset	R/W	Description	Reset Value
INIT5	UMCTL2_BA+0x0e4	R/W	SDRAM Initialization Register 5	0x0010_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
dev_zqinit_x32							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**dev_zqinit_x32**	**ZQ initial calibration, tZQINIT** Present only in designs configured to support DDR3. DDR3 typically requires 512 SDRAM clock cycles. When the controller is operating in 1:2 frequency ratio mode, program this to JEDEC spec value divided by 2, and round it up to the next integer value. Unit: Multiples of 32 DFI clock cycles. **Programming Mode:** Static
[15:0]	**Reserved**	Reserved.

UMCTL2 DIMM Control Register (DIMMCTL)

Register	Offset	R/W	Description	Reset Value
DIMMCTL	UMCTL2_BA+0x0f0	R/W	DIMM Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**dimm_addr_mirr_en**	**dimm_stagger_cs_en**

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**dimm_addr_mirr_en**	**DIMM Address Mirroring Enable Bit** Enables address mirroring (for multi-rank UDIMM implementations). Some UDIMMs implement address mirroring for odd ranks, which means that the following address, bank address and bank group bits are swapped: (A3, A4), (A5, A6), (A7, A8), (BA0, BA1). Setting this bit ensures that, for mode register accesses during the automatic initialization routine, these bits are swapped within the uMCTL2 to compensate for this UDIMM/RDIMM/LRDIMM swapping. **Note:** This has no effect on the address of any other memory accesses, or of software-driven mode register accesses. 1 = For odd ranks, implement address mirroring for MRS commands during initialization. 0 = Do not implement address mirroring. **Programming Mode:** Static
[0]	**dimm_stagger_cs_en**	**DIMM Staggering Enable Bit** Staggering enable for multi-rank accesses (for multi-rank UDIMM, RDIMM and LRDIMM implementations only). **Note:** Even if this bit is set it does not take care of software driven MR commands (through MRCTRL0/MRCTRL1), where software is responsible to send them to separate ranks as appropriate. 1 = DDR2/DDR3 Send all commands to even and odd ranks separately. 0 = Do not stagger accesses. **Programming Mode:** Static

UMCTL2 Rank Control Register (RANKCTL)

Register	Offset	R/W	Description	Reset Value
RANKCTL	UMCTL2_BA+0x0f4	R/W	Rank Control Register	0x0000_066f

31	30	29	28	27	26	25	24
Reserved					diff_rank_wr_gap_msb	Reserved	diff_rank_rd_gap_msb
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
max_rank_wr				diff_rank_wr_gap			
7	6	5	4	3	2	1	0
diff_rank_rd_gap				max_rank_rd			

Bits	Description	
[31:27]	**Reserved**	Reserved.
[26]	**diff_rank_wr_gap_msb**	**Different Rank Write Gap Extension Bit** Only present for multi-rank configurations. 1-bit extension to be used when RANKCTL.diff_rank_wr_gap field needs to be set to a value greater than 0xF. **Programming Mode:** Quasi-dynamic Group 2
[25]	**Reserved**	Reserved.
[24]	**diff_rank_rd_gap_msb**	**Different Rank Read Gap Extension Bit** Only present for multi-rank configurations. 1-bit extension to be used when RANKCTL.diff_rank_rd_gap field needs to be set to a value greater than 0xF. **Programming Mode:** Quasi-dynamic Group 2
[23:16]	**Reserved**	Reserved.
[15:12]	**max_rank_wr**	**Maximum Number of Writes that Can be Scheduled Consecutively to Same Rank** Only present for multi-rank configurations. Background: Writes to the same rank can be performed back-to-back. Writes to different ranks require additional gap dictated by the register RANKCTL.diff_rank_wr_gap. This is to avoid possible data bus contention as well as to give PHY enough time to switch the delay when changing ranks. The uMCTL2 arbitrates for bus access on a cycle-by-cycle basis; therefore after a write is scheduled, there are few clock cycles (determined by the value on RANKCTL.diff_rank_wr_gap register) in which only writes from the same rank are eligible to be scheduled. This prevents writes from other ranks from having fair access to the data bus. This parameter represents the maximum number of writes that can be scheduled consecutively to the same rank. After this number is reached, a delay equal to RANKCTL.diff_rank_wr_gap is inserted by the scheduler to allow all ranks a fair opportunity to be scheduled. Higher numbers increase bandwidth utilization, lower numbers increase fairness.

		This feature can be DISABLED by setting this register to 0. When set to 0, the controller stays on the same rank as long as commands are available for it. Minimum programmable value is 0 (feature disabled) and maximum programmable value is 0xF. For uPCTL2, this register field must be set to 0 (feature disabled) FOR PERFORMANCE ONLY. **Programming Mode:** Static
[11:8]	**diff_rank_wr_gap**	**Number of Clocks of Gap in Data Responses when Performing Consecutive Writes to Different Ranks** Only present for multi-rank configurations. Indicates the number of clocks of gap in data responses when performing consecutive writes to different ranks. This is used to switch the delays in the PHY to match the rank requirements. This value must consider both PHY requirement and ODT requirement. • PHY requirement - tphy_wrcsgap (For more information on value of tphy_wrcsgap, see PHY databook) If CRC feature is enabled, must be increased by 1. • ODT requirement - The value programmed in this register takes care of the ODT switch off timing requirement when switching ranks during writes. For other cases, diff_rank_wr_gap must be a minimum of ODTCFG.wr_odt_hold - BL/2 When the controller is operating in 1:1 mode, program this to the larger of PHY requirement or ODT requirement. If CRC is enabled, the value can be decreased by 2 because the programmed value is internally increased by 1 (it is always 2 in terms of DFI PHY clock cycle as CRC is supported only with 1:2 frequency ratio). When the controller is operating in 1:2 mode, program this to the larger value divided by two and round it up to the next integer. After PHY has completed training the value programmed may need to be increased. For more information, see relevant PHY documentation. If a value greater than 0xF is needed, the RANKCTL.diff_rank_wr_gap_msb field must be used as extension. For DFI 1:2 mode, a maximum value of {diff_rank_wr_gap_msb,diff_rank_wr_gap} < 'h11 is supported. For DFI 1:1 mode a maximum value of {diff_rank_wr_gap_msb,diff_rank_wr_gap} <= 'h1F is supported. Unit: DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 2
[7:4]	**diff_rank_rd_gap**	**Number of Clocks of Gap in Data Responses when Performing Consecutive Reads to Different Ranks** Only present for multi-rank configurations. Indicates the number of clocks of gap in data responses when performing consecutive reads to different ranks. This is used to switch the delays in the PHY to match the rank requirements. This value must consider both PHY requirement and ODT requirement. • PHY requirement:tphy_rdcsgap (For more information on value of tphy_rdcsgap, see PHY databook) • ODT requirement: The value programmed in this register takes care of the ODT switch off timing requirement when switching ranks during reads: diff_rank_rd_gap must be a minimum of ODTCFG.rd_odt_hold - BL/2 when the controller is operating in 1:1 mode, program this to the larger of PHY requirement or ODT requirement. When the controller is operating in 1:2 mode, program this to the larger value divided by two and round it up to the next integer. After PHY has completed training the value programmed may need to be increased. For more information, see relevant PHY documentation. If a value greater than 0xF is needed, the RANKCTL.diff_rank_rd_gap_msb field must be used as extension. For DFI 1:2 mode, a maximum value of

		{diff_rank_rd_gap_msb,diff_rank_rd_gap} < 'h11 is supported. For DFI 1:1 mode a maximum value of {diff_rank_rd_gap_msb,diff_rank_rd_gap} <= 'h1F is supported. Unit: DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 2
[3:0]	**max_rank_rd**	**Maximum Number of Reads that Can be Scheduled Consecutively to Same Rank** Only present for multi-rank configurations. Background: Reads to the same rank can be performed back-to-back. Reads to different ranks require additional gap dictated by the register RANKCTL.diff_rank_rd_gap. This is to avoid possible data bus contention as well as to give PHY enough time to switch the delay when changing ranks. The uMCTL2 arbitrates for bus access on a cycle-by-cycle basis; therefore after a read is scheduled, there are few clock cycles (determined by the value on RANKCTL.diff_rank_rd_gap register) in which only reads from the same rank are eligible to be scheduled. This prevents reads from other ranks from having fair access to the data bus. This parameter represents the maximum number of reads that can be scheduled consecutively to the same rank. After this number is reached, a delay equal to RANKCTL.diff_rank_rd_gap is inserted by the scheduler to allow all ranks a fair opportunity to be scheduled. Higher numbers increase bandwidth utilization, lower numbers increase fairness. This feature can be DISABLED by setting this register to 0. When set to 0, the controller stays on the same rank as long as commands are available for it. Minimum programmable value is 0 (feature disabled) and maximum programmable value is 0xF. For uPCTL2, this register field must be set to 0 (feature disabled) FOR PERFORMANCE ONLY. **Programming Mode:** Static

UMCTL2 SDRAM Timing Register 0 (DRAMTMG0)

Register	Offset	R/W	Description	Reset Value
DRAMTMG0	UMCTL2_BA+0x100	R/W	SDRAM Timing Register 0	0x0f10_1b0f

31	30	29	28	27	26	25	24
Reserved	wr2pre						
23	22	21	20	19	18	17	16
Reserved		t_faw					
15	14	13	12	11	10	9	8
Reserved	t_ras_max						
7	6	5	4	3	2	1	0
Reserved		t_ras_min					

Bits	Description	
[31]	**Reserved**	Reserved.
[30:24]	**wr2pre**	**Minimum Time Between Write and Precharge to Same Bank** Specifications: WL + BL/2 + tWR = approximately 8 cycles + 15 ns = 14 clocks @400 MHz and less for lower frequencies. where: • WL: Write latency • BL: Burst length. This must match the value programmed in the BL bit of the mode register to the SDRAM. BST (burst terminate) is not supported at present • tWR: Write recovery time. This comes directly from the SDRAM specification When the controller is operating in 1:2 frequency ratio mode, 1T mode, divide the previous value by 2. No rounding up. When the controller is operating in 1:2 frequency ratio mode, 2T mode, geardown mode, divide the previous value by 2 and round it up to the next integer value. Note that, depending on the PHY, if using LRDIMM, it may be necessary to adjust the value of this parameter to compensate for the extra cycle of latency through the LRDIMM. Unit: DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 1, Group 2, Group 4
[23:22]	**Reserved**	Reserved.
[21:16]	**t_faw**	**Four Active Window for 2 Kbytes Page Size** tFAW - valid only when 8 or more banks(or banks x bank groups) are present. In 8-bank design, at most 4 banks must be activated in a rolling window of tFAW cycles. When the controller is operating in 1:2 frequency ratio mode, program this to (tFAW/2) and round up to next integer value. In a 4-bank design, set this register to 0x1 independent of the 1:1/1:2 frequency mode. Unit: DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 2, Group 4
[15]	**Reserved**	Reserved.
[14:8]	**t_ras_max**	**Maximum Time Between Activate and Precharge to the Same Bank**

		This is the maximum time that a page can be kept open Minimum value of this register is 1. Zero is invalid. When the controller is operating in 1:1 frequency ratio mode, t_ras_max must be set to RoundDown(tRAS(max)/tCK/1024). When the controller is operating in 1:2 frequency ratio mode, t_ras_max must be set to RoundDown((RoundDown(tRAS(max)/tCK/1024)-1)/2). Unit: Multiples of 1024 DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 2, Group 4
[7:6]	**Reserved**	Reserved.
[5:0]	**t_ras_min**	**Minimum Time Between Activate and Precharge to the Same Bank** When the controller is operating in 1:1 frequency ratio mode, t_ras_min must be set to RoundUp(tRASmin/tCK) When the controller is operating in 1:2 frequency ratio mode, 1T mode, t_ras_min must be set to RoundDown(RoundUp(tRASmin/tCK)/2) When the controller is operating in 1:2 frequency ratio mode, 2T mode, geardown mode or LPDDR4 mode, t_ras min must be set to RoundUp(RoundUp(tRASmin/tCK)/2) Unit: DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 2, Group 4

UMCTL2 SDRAM Timing Register 1 (DRAMTMG1)

Register	Offset	R/W	Description	Reset Value
DRAMTMG1	UMCTL2_BA+0x104	R/W	SDRAM Timing Register 1	0x0008_0414

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			t_xp				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		rd2pre					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	t_rc						

Bits	Description	
[31:21]	**Reserved**	Reserved.
[20:16]	**t_xp**	**Minimum Time after Power-Down Exit to Any Operation** For DDR3, this must be programmed to tXPDLL if slow power down exit is selected in MR0[12]. When the controller is operating in 1:2 frequency ratio mode, program this to (tXP/2) and round it up to the next integer value. Units: DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 2, Group 4
[15:14]	**Reserved**	Reserved.
[13:8]	**rd2pre**	**Minimum Time from Read to Precharge of Same Bank** • DDR2 - tAL + BL/2 + max(RoundUp(tRTP/tCK), 2) - 2 • DDR3 - tAL + max (RoundUp(tRTP/tCK), 4) When the controller is operating in 1:2 mode, 1T mode, divide the previous value by 2. No rounding up. When the controller is operating in 1:2 mode, 2T mode, geardown mode, divide the previous value by 2 and round it up to the next integer value. Unit: DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 1, Group 2, Group 4
[7]	**Reserved**	Reserved.
[6:0]	**t_rc**	**Minimum Time Between Activates to Same Bank** When the controller is operating in 1:2 frequency ratio mode, program this to (tRC/2) and round up to next integer value. Unit: DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 2, Group 4

UMCTL2 SDRAM Timing Register 2 (DRAMTMG2)

Register	Offset	R/W	Description	Reset Value
DRAMTMG2	UMCTL2_BA+0x108	R/W	SDRAM Timing Register 2	0x0000_060d

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		rd2wr					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		wr2rd					

Bits	Description	
[31:14]	**Reserved**	Reserved.
[13:8]	**rd2wr**	**Minimum Time from Read Command to Write Command** This must include time for bus turnaround (both within ranks and between ranks) and all PHY and system requirements. After the PHY has completed training, the value programmed may need to be increased. Please see the relevant PHY databook for details of what should be included here. The following calculations are minimum values, and do not include the PHY/system requirements mentioned above: DDR2/3 : RL + BL/2 + 2 - WL Where: • WL: Write latency • BL: Burst length. This must match the value programmed in the BL bit of the mode register to the SDRAM • RL: Read latency = CAS latency. When the controller is operating in 1:2 frequency ratio mode, divide the value calculated using the previous equation by 2, and round it up to next integer. Unit: DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 1, Group 2, Group 4
[7:6]	**Reserved**	Reserved.
[5:0]	**wr2rd**	**Minimum Time from Write Command to Read Command** This must include time for bus turn-around and all PHY and system requirements. After the PHY has completed training, the value programmed may need to be increased. Please see the relevant PHY databook for details of what should be included here. The following calculations are minimum values, and do not include the PHY/system requirements mentioned above: DDR2/DDR3: CWL + BL/2 + tWTR Where: • CWL: CAS write latency • WL: Write latency

		• BL: Burst length. This must match the value programmed in the BL bit of the mode register to the SDRAM • tWTR: Internal write to read command delay. This comes directly from the SDRAM specification Add one extra cycle for LPDDR2/LPDDR3/LPDDR4 operation. When the controller is operating in 1:2 mode, divide the value calculated using the previous equation by 2, and round it up to next integer. If your configuration has RANKCTL1.wr2rd_dr, write to read bus turn-around between different physical ranks are controlled by RANKCTL1.wr2rd_dr. Unit: DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 1, Group 2, Group 4

UMCTL2 SDRAM Timing Register 3 (DRAMTMG3)

Register	Offset	R/W	Description	Reset Value
DRAMTMG3	UMCTL2_BA+0x10c	R/W	SDRAM Timing Register 3	0x0000_400c

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						t_mrd	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
t_mrd				Reserved		t_mod	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
t_mod							

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17:12]	**t_mrd**	**Number of Cycles to Wait after a Mode Register Write or Read** Depending on the connected SDRAM, tMRD represents: • DDR2: Time from MRS to any command • DDR3: Time from MRS to MRS command When the controller is operating in 1:2 frequency ratio mode, program this to (tMRD/2) and round it up to the next integer value. If CAL mode is enabled (DFITMG1.dfi_t_cmd_lat > 0), tCAL (=DFITMG1.dfi_cmd_lat) must be added to the previous calculations. Unit: DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 2, Group 4
[11:10]	**Reserved**	Reserved.
[9:0]	**t_mod**	**Number of Cycles Between Load Mode Command and Following Non-Load Mode Command** Used only in DDR3. Indicates the number of cycles between load mode command and following non-load mode command. If CAL mode is enabled (DFITMG1.dfi_t_cmd_lat > 0), tCAL (=DFITMG1.dfi_cmd_lat) must be added to the previous calculations. Set to tMOD if controller is operating in 1:1 frequency ratio mode, or tMOD/2 (rounded up to next integer) if controller is operating in 1:2 frequency ratio mode. Unit: DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 2, Group 4

UMCTL2 SDRAM Timing Register 4 (DRAMTMG4)

Register	Offset	R/W	Description	Reset Value
DRAMTMG4	UMCTL2_BA+0x110	R/W	SDRAM Timing Register 4	0x0504_0405

31	30	29	28	27	26	25	24
Reserved			t_rcd				
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				t_ccd			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				t_rrd			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			t_rp				

Bits	Description	
[31:29]	**Reserved**	Reserved.
[28:24]	**t_rcd**	**Minimum Time from Activate to Read or Write Command to Same Bank** When the controller is operating in 1:2 frequency ratio mode, program this to ((tRCD - tAL)/2) and round it up to the next integer value. Minimum value allowed for this register is 1, which implies minimum (tRCD - tAL) value to be 2 when the controller is operating in 1:2 frequency ratio mode. Unit: DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 1, Group 2, Group 4
[23:20]	**Reserved**	Reserved.
[19:16]	**t_ccd**	**Minimum Time Between Two Reads or Two Writes for Others** When the controller is operating in 1:2 frequency ratio mode, program this to (tCCD_L/2 or tCCD/2) and round it up to the next integer value. Unit: DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 2, Group 4
[15:12]	**Reserved**	Reserved.
[11:8]	**t_rrd**	**Minimum Time Between Activates from Bank "a" to Bank "b" for Others** When the controller is operating in 1:2 frequency ratio mode, program this to (tRRD/2) and round it up to the next integer value. Unit: DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 2, Group 4
[7:5]	**Reserved**	Reserved.
[4:0]	**t_rp**	**Minimum Time from Single-Bank Precharge to Activate of Same Bank** When the controller is operating in 1:1 frequency ratio mode, t_rp must be set to RoundUp(tRP/tCK). When the controller is operating in 1:2 frequency ratio mode, t_rp must be set to RoundDown(RoundUp(tRP/tCK)/2) + 1. Unit: DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 2, Group 4

UMCTL2 SDRAM Timing Register 5 (DRAMTMG5)

Register	Offset	R/W	Description	Reset Value
DRAMTMG5	UMCTL2_BA+0x114	R/W	SDRAM Timing Register 5	0x0505_0403

31	30	29	28	27	26	25	24
Reserved				t_cksrx			
23	22	21	20	19	18	17	16
Reserved	t_cksre						
15	14	13	12	11	10	9	8
Reserved		t_ckesr					
7	6	5	4	3	2	1	0
Reserved			t_cke				

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:24]	**t_cksrx**	**Time Before Self-Refresh Exit that CK is Maintained as a Valid Clock Before Issuing SRX** Specifies the clock stable time before SRX. Recommended settings: • DDR2 : 1 • DDR3 : tCKSRX When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer. Unit: DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 2, Group 4
[23]	**Reserved**	Reserved.
[22:16]	**t_cksre**	**Time After Self-Refresh Down Entry that CK is Maintained as a Valid Clock** Specifies the clock disable delay after SRE. Recommended settings: • DDR2 : 1 • DDR3 : Max (10 ns, 5 tCK) When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer. Unit: DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 2, Group 4
[15:14]	**Reserved**	Reserved.
[13:8]	**t_ckesr**	**Minimum CKE Low Width for Self-Refresh or Self-Refresh Power Down Entry to Exit Timing in Memory Clock Cycles** Recommended settings: • DDR2 : tCKE • DDR3 : tCKE + 1

		When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer. Unit: DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 2, Group 4
[7:5]	**Reserved**	Reserved.
[4:0]	**t_cke**	**Minimum Number of Cycles of CKE HIGH/LOW During Power-Down and Self-Refresh** • DDR2/DDR3: Set this to tCKE value When the controller is operating in 1:2 frequency ratio mode, program this to (value described above)/2 and round it up to the next integer value. Unit: DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 2, Group 4

UMCTL2 SDRAM Timing Register 8 (DRAMTMG8)

Register	Offset	R/W	Description	Reset Value
DRAMTMG8	UMCTL2_BA+0x120	R/W	SDRAM Timing Register 8	0x0000_4405

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	t_xs_dll_x32						
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	t_xs_x32						

Bits	Description	
[31:15]	**Reserved**	Reserved.
[14:8]	**t_xs_dll_x32**	**Exit Self-Refresh to Commands Requiring a Locked DLL** When the controller is operating in 1:2 frequency ratio mode, program this to the previous value divided by 2 and round up to next integer value. **Note:** Used only for DDR2 and DDR3 SDRAMs. Unit: Multiples of 32 DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 2, Group 4
[7]	**Reserved**	Reserved.
[6:0]	**t_xs_x32**	**Exit Self-Refresh to Commands Not Requiring a Locked DLL** When the controller is operating in 1:2 frequency ratio mode, program this to the previous value divided by 2 and round up to next integer value. **Note:** Used only for DDR2 and DDR3 SDRAMs. Unit: Multiples of 32 DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 2, Group 4

UMCTL2 SDRAM Timing Register 15 (DRAMTMG15)

Register	Offset	R/W	Description	Reset Value
DRAMTMG15	UMCTL2_BA+0x13c	R/W	SDRAM Timing Register 15	0x0000_0000

31	30	29	28	27	26	25	24
en_dfi_lp_t_stab	Reserved						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
t_stab_x32							

Bits	Description	
[31]	**en_dfi_lp_t_stab**	**Enable Bit for Using tSTAB When Exiting DFI LP** 1 = Enable using tSTAB when exiting DFI LP. This must be set when the PHY is stopping the clock during DFI LP to save maximum power. 0 = Disable using tSTAB when exiting DFI LP. **Programming Mode:** Quasi-dynamic Group 2, Group 4
[30:8]	**Reserved**	Reserved.
[7:0]	**t_stab_x32**	**Stabilization time** It is required in the following two cases for DDR3 RDIMM: ● When exiting power saving mode, if the clock is stopped, after re-enabling it the clock must be stable for a time specified by tSTAB ● After issuing control words that refers to clock timing (Specification: 6us for DDR3) When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer. Unit: Multiples of 32 DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 2, Group 4

UMCTL2 ZQ Control Register 0 (ZQCTL0)

Register	Offset	R/W	Description	Reset Value
ZQCTL0	UMCTL2_BA+0x180	R/W	ZQ Control Register 0	0x0200_0040

31	30	29	28	27	26	25	24
dis_auto_zq	dis_srx_zqcl	zq_resistor_shared	Reserved		t_zq_long_nop		
23	22	21	20	19	18	17	16
t_zq_long_nop							
15	14	13	12	11	10	9	8
Reserved						t_zq_short_nop	
7	6	5	4	3	2	1	0
t_zq_short_nop							

Bits	Description	
[31]	**dis_auto_zq**	**Disable Auto Generation of ZQCS/MPC Command** 1 = Disable uMCTL2 generation of ZQCS/MPC(ZQ calibration) command. Register DBGCMD.zq_calib_short can be used instead to issue ZQ calibration request from APB module 0 = Internally generate ZQCS/MPC(ZQ calibration) commands based on ZQCTL1.t_zq_short_interval_x1024. This is only present for designs supporting DDR3 devices. **Programming Mode:** Dynamic
[30]	**dis_srx_zqcl**	**Disable Issuing of ZQCL/MPC Command at Self-Refresh/SR-Power Down Exit** 1 = Disable issuing of ZQCL/MPC(ZQ calibration) command at self-refresh/SR-Power down exit. Only applicable when run in DDR3 mode 0 = Enable issuing of ZQCL/MPC(ZQ calibration) command at self-refresh/SR-Power down exit. Only applicable when run in DDR3 mode This is only present for designs supporting DDR3 devices. **Programming Mode:** Quasi-dynamic Group 2, Group 4
[29]	**zq_resistor_shared**	**ZQ Resistor Shared Between Ranks** 1 = Denotes that ZQ resistor is shared between ranks. Means ZQinit/ZQCL/ZQCS/MPC(ZQ calibration) commands are sent to one rank at a time with tZQinit/tZQCL/tZQCS/tZQCAL/tZQLAT timing met between commands so that commands to different ranks do not overlap 0 = ZQ resistor is not shared. This is only present for designs supporting DDR3 devices. **Programming Mode:** Static
[28:27]	**Reserved**	Reserved.
[26:16]	**t_zq_long_nop**	**Number of DFI Clock Cycles of NOP Required After a ZQCL/MPC Command is Issued to SDRAM** tZQoper for DDR3 When the controller is operating in 1:2 frequency ratio mode: DDR3: program this to tZQoper/2 and round it up to the next integer value. This is only present for designs supporting DDR3 devices. Unit: DFI clock cycles.

		Programming Mode: Static
[15:10]	**Reserved**	Reserved.
[9:0]	**t_zq_short_nop**	**Number of DFI Clock Cycles of NOP Required After a ZQCS/MPC Command is Issued to SDRAM** tZQCS for DDR3 When the controller is operating in 1:2 frequency ratio mode, program this to tZQCS/2 and round it up to the next integer value. This is only present for designs supporting DDR3 devices. Unit: DFI clock cycles. **Programming Mode:** Static

UMCTL2 ZQ Control Register 1 (ZQCTL1)

Register	Offset	R/W	Description	Reset Value
ZQCTL1	UMCTL2_BA+0x184	R/W	ZQ Control Register 1	0x0000_0100

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				t_zq_short_interval_x1024			
15	14	13	12	11	10	9	8
t_zq_short_interval_x1024							
7	6	5	4	3	2	1	0
t_zq_short_interval_x1024							

Bits	Description	
[31:20]	**Reserved**	Reserved.
[19:0]	**t_zq_short_interval_x1024**	**Average Interval to Wait Between Automatically Issuing ZQCS** Average Interval to Wait Between Automatically Issuing ZQCS (ZQ Calibration Short)/MPC(ZQ Calibration) Commands to DDR3 Devices. Meaningless, if ZQCTL0.dis_auto_zq=1. This is only present for designs supporting DDR3 devices. Unit: Multiples of 1024 DFI clock cycles. **Programming Mode:** Static

UMCTL2 DFI Timing Register 0 (DFITMG0)

Register	Offset	R/W	Description	Reset Value
DFITMG0	UMCTL2_BA+0x190	R/W	DFI Timing Register 0	0x0702_0002

31	30	29	28	27	26	25	24
Reserved			dfi_t_ctrl_delay				
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
dfi_rddata_use_dfi_phy_clk	dfi_t_rddata_en						
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
dfi_wrdata_use_dfi_phy_clk	Reserved	dfi_tphy_wrdata					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		dfi_tphy_wrlat					

Bits	Description	
[31:29]	**Reserved**	Reserved.
[28:24]	**dfi_t_ctrl_delay**	**Number of DFI Clock Cycles After an Assertion or De-assertion of the DFI Control Signals that the Control Signals at the PHY-DRAM Interface Reflect the Assertion or De-assertion** If the DFI clock and the memory clock are not phase-aligned, this timing parameter must be rounded up to the next integer value. Unit: DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 4
[23]	**dfi_rddata_use_dfi_phy_clk**	**dfi_rddata_en/dfi_rddata/dfi_rddata_valid is Generated Using HDR or SDR Values** Selects whether value in DFITMG0.dfi_t_rddata_en is in terms of HDR (DFI clock) or SDR (DFI PHY clock) cycles: • 0 in terms of HDR (DFI clock) cycles Refer to PHY specification for correct value. If using a Synopsys DWC DDR3/2 PHY, this field must be set to 0. **Programming Mode:** Static
[22:16]	**dfi_t_rddata_en**	**Time from the Assertion of a Read Command on the DFI Interface to the Assertion of the dfi_rddata_en Signal** Refer to PHY specification for correct value. This corresponds to the DFI parameter trddata_en. Unit: DFI clock cycles or DFI PHY clock cycles, depending on DFITMG0.dfi_rddata_use_dfi_phy_clk. **Programming Mode:** Quasi-dynamic Group 1, Group 4
[15]	**dfi_wrdata_use_dfi_phy_clk**	**dfi_wrdata_en/dfi_wrdata/dfi_wrdata_mask is Generated Using HDR or SDR Values** Selects whether value in DFITMG0.dfi_tphy_wrlat is in terms of HDR (DFI clock) or SDR (DFI PHY clock) cycles. Selects whether value in DFITMG0.dfi_tphy_wrdata is in terms of HDR (DFI clock) or SDR (DFI PHY clock) cycles. • 0 in terms of HDR (DFI clock) cycles

		Refer to PHY specification for correct value. If using a Synopsys DWC DDR3/2 PHY, this field must be set to 0. **Programming Mode:** Static
[14]	**Reserved**	Reserved.
[13:8]	**dfi_tphy_wrdata**	**Number of Clock Cycles Between When dfi_wrdata_en is Asserted to When the Associated Write Data is Driven on the dfi_wrdata Signal** This corresponds to the DFI timing parameter tphy_wrdata. For more information on correct value, see PHY specification. Note, maximum supported value is 8. Unit: DFI clock cycles or DFI PHY clock cycles, depending on DFITMG0.dfi_wrdata_use_dfi_phy_clk. **Programming Mode:** Quasi-dynamic Group 4
[7:6]	**Reserved**	Reserved.
[5:0]	**dfi_tphy_wrlat**	**Write latency** Number of clocks from the write command to write data enable (dfi_wrdata_en). This corresponds to the DFI timing parameter tphy_wrlat. Refer to PHY specification for correct value. Unit: DFI clock cycles or DFI PHY clock cycles, depending on DFITMG0.dfi_wrdata_use_dfi_phy_clk. **Programming Mode:** Quasi-dynamic Group 2, Group 4

UMCTL2 DFI Timing Register 1 (DFITMG1)

Register	Offset	R/W	Description	Reset Value
DFITMG1	UMCTL2_BA+0x194	R/W	DFI Timing Register 1	0x0000_0404

31	30	29	28	27	26	25	24
Reserved						dfi_t_parin_lat	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			dfi_t_wrdata_delay				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			dfi_t_dram_clk_disable				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			dfi_t_dram_clk_enable				

Bits	Description	
[31:26]	**Reserved**	Reserved.
[25:24]	**dfi_t_parin_lat**	**Number of DFI PHY Clock Cycles Between When the dfi_cs signal is Asserted and When the Associated dfi_parity_in Signal is Driven** Unit: DFI PHY clock cycles. **Programming Mode:** Quasi-dynamic Group 4
[23:21]	**Reserved**	Reserved.
[20:16]	**dfi_t_wrdata_delay**	**Number of DFI Clock Cycles Between When the dfi_wrdata_en Signal is Asserted and When the Corresponding Write Data Transfer is Completed on the DRAM Bus** This corresponds to the DFI timing parameter twrdata_delay. For more information on correct value, see PHY specification. For DFI 2.1 PHY, set to tphy_wrdata + (delay of DFI write data to the DRAM). Value to be programmed is in terms of DFI clocks, not PHY clocks. In FREQ_RATIO=2, divide PHY's value by 2 and round up to next integer. If using DFITMG0.dfi_wrdata_use_dfi_phy_clk=1, add 1 to the value. Unit: DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 4
[15:13]	**Reserved**	Reserved.
[12:8]	**dfi_t_dram_clk_disable**	**Number of DFI Clock Cycles from the Assertion of the dfi_dram_clk_disable Signal on the DFI Until the Clock to the DRAM Memory Devices, at the PHY-DRAM Boundary, Maintains a Low Value** If the DFI clock and the memory clock are not phase aligned, this timing parameter must be rounded up to the next integer value. Unit: DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 4
[7:5]	**Reserved**	Reserved.
[4:0]	**dfi_t_dram_clk_enable**	**Number of DFI Clock Cycles from the De-assertion of the dfi_dram_clk_disable Signal on the DFI Until the First Valid Rising Edge of the Clock to the DRAM Memory Devices, at the PHY-DRAM Boundary**

		If the DFI clock and the memory clock are not phase aligned, this timing parameter must be rounded up to the next integer value. Unit: DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 4

UMCTL2 DFI Low Power Configuration Register 0 (DFILPCFG0)

Register	Offset	R/W	Description	Reset Value
DFILPCFG0	UMCTL2_BA+0x198	R/W	DFI Low Power Configuration Register 0	0x0700_0000

31	30	29	28	27	26	25	24
Reserved			dfi_tlp_resp				
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
dfi_lp_wakeup_sr				Reserved			dfi_lp_en_sr
7	6	5	4	3	2	1	0
dfi_lp_wakeup_pd				Reserved			dfi_lp_en_pd

Bits	Description	
[31:29]	**Reserved**	Reserved.
[28:24]	**dfi_tlp_resp**	**Setting in DFI Clock Cycles for DFI's tlp_resp Time** Same value is used for both Power Down, self-refresh, Deep Power Down and Maximum Power Saving modes. For more information on recommended values, see PHY databook Unit: DFI clock cycles. **Programming Mode:** Static
[23:16]	**Reserved**	Reserved.
[15:12]	**dfi_lp_wakeup_sr**	**Value in DFI Clock Cycles to Drive on dfi_lp_wakeup Signal When Self-Refresh Mode is Entered** Determines the DFI's tlp_wakeup time: 0x0 = 16 cycles. 0x1 = 32 cycles. 0x2 = 64 cycles. 0x3 = 128 cycles. 0x4 = 256 cycles. 0x5 = 512 cycles. 0x6 = 1024 cycles. 0x7 = 2048 cycles. 0x8 = 4096 cycles. 0x9 = 8192 cycles. 0xA = 16384 cycles. 0xB = 32768 cycles. 0xC = 65536 cycles. 0xD = 131072 cycles. 0xE = 262144 cycles. 0xF = Unlimited. Unit: DFI clock cycles.

		Programming Mode: Static
[11:9]	**Reserved**	Reserved.
[8]	**dfi_lp_en_sr**	**Enable Bit for DFI Low Power Interface Handshaking During Self-Refresh Entry/Exit** 0 = Disabled. 1 = Enabled. **Programming Mode:** Static
[7:4]	**dfi_lp_wakeup_pd**	**Value in DFI Clock Cycles to Drive on dfi_lp_wakeup Signal When Power-Down Mode is Entered** Determines the DFI's tlp_wakeup time: 0x0 = 16 cycles. 0x1 = 32 cycles. 0x2 = 64 cycles. 0x3 = 128 cycles. 0x4 = 256 cycles. 0x5 = 512 cycles. 0x6 = 1024 cycles. 0x7 = 2048 cycles. 0x8 = 4096 cycles. 0x9 = 8192 cycles. 0xA = 16384 cycles. 0xB = 32768 cycles. 0xC = 65536 cycles. 0xD = 131072 cycles. 0xE = 262144 cycles. 0xF = Unlimited. Unit: DFI clock cycles. Programming Mode: Static
[3:1]	**Reserved**	Reserved.
[0]	**dfi_lp_en_pd**	**Enable Bit for DFI Low Power Interface Handshaking During Power Down Entry/Exit** 0 = Disabled. 1 = Enabled. **Programming Mode:** Static

UMCTL2 DFI Update Register 0 (DFIUPD0)

Register	Offset	R/W	Description	Reset Value
DFIUPD0	UMCTL2_BA+0x1a0	R/W	DFI Update Register 0	0x0040_0003

31	30	29	28	27	26	25	24
dis_auto_ctrlupd	dis_auto_ctrlupd_srx	ctrlupd_pre_srx	Reserved			dfi_t_ctrlup_max	
23	22	21	20	19	18	17	16
dfi_t_ctrlup_max							
15	14	13	12	11	10	9	8
Reserved						dfi_t_ctrlup_min	
7	6	5	4	3	2	1	0
dfi_t_ctrlup_min							

Bits	Description	
[31]	**dis_auto_ctrlupd**	**Disable Bit for Automatic dfi_ctrlupd_req Generation by uMCTL2** When '1', disable the automatic dfi_ctrlupd_req generation by the uMCTL2. The controller must issue the dfi_ctrlupd_req signal using register DBGCMD.ctrlupd. When '0', uMCTL2 issues dfi_ctrlupd_req periodically. **Programming Mode:** Quasi-dynamic Group 3
[30]	**dis_auto_ctrlupd_srx**	**Disable Bit for Automatic dfi_ctrlupd_req Generation by uMCTL2 at Self-Refresh Exit** When '1', disable the automatic dfi_ctrlupd_req generation by the uMCTL2 at self-refresh exit. When '0', uMCTL2 issues a dfi_ctrlupd_req before or after exiting self-refresh, depending on DFIUPD0.ctrlupd_pre_srx. **Programming Mode:** Static
[29]	**ctrlupd_pre_srx**	**dfi_ctrlupd_req Selection at SRX** 0 = Send ctrlupd after SRX. 1 = Send ctrlupd before SRX If DFIUPD0.dis_auto_ctrlupd_srx=1, this register has no impact, because no dfi_ctrlupd_req is issued when SRX. **Programming Mode:** Static
[28:26]	**Reserved**	Reserved.
[25:16]	**dfi_t_ctrlup_max**	**Maximum Number of DFI Clock Cycles that the dfi_ctrlupd_req Signal Can Assert** Lowest value to assign to this variable is 0x40. Unit: DFI clock cycles. **Programming Mode:** Static
[15:10]	**Reserved**	Reserved.
[9:0]	**dfi_t_ctrlup_min**	**Minimum Number of DFI Clock Cycles that the dfi_ctrlupd_req Signal must be Asserted The uMCTL2 expects the PHY to respond within this time If the PHY does not respond, the uMCTL2 de-asserts dfi_ctrlupd_req after dfi_t_ctrlup_min + 2 cycles Lowest value to assign to this variable is 0x1** Unit: DFI clock cycles. **Programming Mode:** Static

UMCTL2 DFI Update Register 1 (DFIUPD1)

Register	Offset	R/W	Description	Reset Value
DFIUPD1	UMCTL2_BA+0x1a4	R/W	DFI Update Register 1	0x0001_0001

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
dfi_t_ctrlupd_interval_min_x1024							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
dfi_t_ctrlupd_interval_max_x1024							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**dfi_t_ctrlupd_interval_min_x1024**	**Minimum Amount of Time Between uMCTL2 Initiated DFI Update Requests** Set this number higher to reduce the frequency of update requests, which can have a small impact on the latency of the first read request when the uMCTL2 is idle. The minimum allowed value for this field is 1. Unit: Multiples of 1024 DFI clock cycles. **Programming Mode:** Static
[15:8]	**Reserved**	Reserved.
[7:0]	**dfi_t_ctrlupd_interval_max_x1024**	**Maximum Amount of Time Between uMCTL2 Initiated DFI Update Requests** This timer resets with each update request; when the timer expires dfi_ctrlupd_req is sent and traffic is blocked until the dfi_ctrlupd_ackx is received. PHY can use this idle time to recalibrate the delay lines to the DLLs. The DFI controller update is also used to reset PHY FIFO pointers in case of data capture errors. Updates are required to maintain calibration over PVT, but frequent updates may impact performance. Minimum allowed value for this field is 1. **Note:** Value programmed for DFIUPD1.dfi_t_ctrlupd_interval_max_x1024 must be greater than DFIUPD1.dfi_t_ctrlupd_interval_min_x1024. Unit: Multiples of 1024 DFI clock cycles. **Programming Mode:** Static

UMCTL2 DFI Update Register 2 (DFIUPD2)

Register	Offset	R/W	Description	Reset Value
DFIUPD2	UMCTL2_BA+0x1a8	R/W	DFI Update Register 2	0x8000_0000

31	30	29	28	27	26	25	24
dfi_phyupd_en	Reserved						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31]	**dfi_phyupd_en**	**Enable Bit for Acknowledging PHY-Initiated Updates** 0 = Disabled. 1 = Enabled. **Programming Mode:** Static
[30:0]	**Reserved**	Reserved.

UMCTL2 DFI Miscellaneous Control Register (DFIMISC)

Register	Offset	R/W	Description	Reset Value
DFIMISC	UMCTL2_BA+0x1b0	R/W	DFI Miscellaneous Control Register	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved			dfi_frequency				
7	6	5	4	3	2	1	0
Reserved		dfi_init_start	ctl_idle_en	Reserved			dfi_init_complete_en

Bits	Description	
[31:13]	**Reserved**	Reserved.
[12:8]	**dfi_frequency**	**Operating Frequency of System** The number of supported frequencies and the mapping of signal values to clock frequencies are defined by the PHY. **Programming Mode:** Quasi-dynamic Group 1
[7:6]	**Reserved**	Reserved.
[5]	**dfi_init_start**	**PHY Initialization Start Request Signal** When asserted it triggers the PHY init start request. **Programming Mode:** Quasi-dynamic Group 3
[4]	**ctl_idle_en**	**Enable Bit for ctl_idle Signal** It is non-DFI related pin specific to certain Synopsys PHYs. For more information on ctl_idle functionality, see signal description of ctl_idle signal. **Programming Mode:** Static
[3:1]	**Reserved**	Reserved.
[0]	**dfi_init_complete_en**	**PHY Initialization Complete Enable Bit** When asserted the dfi_init_complete signal can be used to trigger SDRAM initialisation **Programming Mode:** Quasi-dynamic Group 3

UMCTL2 DFI Status Register (DFISTAT)

Register	Offset	R/W	Description	Reset Value
DFISTAT	UMCTL2_BA+0x1bc	R	DFI Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						dfi_lp_ack	dfi_init_complete

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**dfi_lp_ack**	**Value of the dfi_lp_ack to the controller** **Programming Mode:** Dynamic
[0]	**dfi_init_complete**	**Status flag when the DFI initialization has been completed** The DFI INIT triggered by dfi_init_start signal and then the dfi_init_complete flag is polled to know when the initialization is done. **Programming Mode:** Dynamic

UMCTL2 DFI PHY Master (DFIPHYMSTR)

Register	Offset	R/W	Description	Reset Value
DFIPHYMSTR	UMCTL2_BA+0x1c4	R/W	DFI PHY Master	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**dfi_phymstr_en**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**dfi_phymstr_en**	**PHY Master Interface Enable Bit** 0 = Disabled. 1 = Enabled. **Programming Mode:** Static

UMCTL2 Address Map Register 0 (ADDRMAP0)

Register	Offset	R/W	Description	Reset Value
ADDRMAP0	UMCTL2_BA+0x200	R/W	Address Map Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			addrmap_cs_bit0				

Bits	Description	
[31:5]	**Reserved**	Reserved.
[4:0]	**addrmap_cs_bit0**	**HIF Address Bits for Rank Address Bit 0** Valid Range: 0 to 29, and 31 Internal Base: 6 The selected HIF address bit is determined by adding the internal base to the value of this field. If unused, set to 31 and then rank address bit 0 is set to 0. **Programming Mode:** Static

UMCTL2 Address Map Register 1 (ADDRMAP1)

Register	Offset	R/W	Description	Reset Value
ADDRMAP1	UMCTL2_BA+0x204	R/W	Address Map Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved		addrmap_bank_b2					
15	14	13	12	11	10	9	8
Reserved		addrmap_bank_b1					
7	6	5	4	3	2	1	0
Reserved		addrmap_bank_b0					

Bits	Description	
[31:22]	**Reserved**	Reserved.
[21:16]	**addrmap_bank_b2**	**HIF Address Bits for Bank Address Bit 2** Valid Range: 0 to 31, and 63 Internal Base: 4 The selected HIF address bit is determined by adding the internal base to the value of this field. If unused, set to 63 and then bank address bit 2 is set to 0. **Programming Mode:** Static
[15:14]	**Reserved**	Reserved.
[13:8]	**addrmap_bank_b1**	**HIF Address Bits for Bank Address Bit 1** Valid Range: 0 to 32, and 63 Internal Base: 3 The selected HIF address bit for each of the bank address bits is determined by adding the internal base to the value of this field. If unused, set to 63 and then bank address bit 1 is set to 0. **Programming Mode:** Static
[7:6]	**Reserved**	Reserved.
[5:0]	**addrmap_bank_b0**	**HIF Address Bits for Bank Address Bit 0** Valid Range: 0 to 32, and 63 Internal Base: 2 The selected HIF address bit for each of the bank address bits is determined by adding the internal base to the value of this field. If unused, set to 63 and then bank address bit 0 is set to 0. **Programming Mode:** Static

UMCTL2 Address Map Register 2 (ADDRMAP2)

Register	Offset	R/W	Description	Reset Value
ADDRMAP2	UMCTL2_BA+0x208	R/W	Address Map Register 2	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				addrmap_col_b5			
23	22	21	20	19	18	17	16
Reserved				addrmap_col_b4			
15	14	13	12	11	10	9	8
Reserved			addrmap_col_b3				
7	6	5	4	3	2	1	0
Reserved				addrmap_col_b2			

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:24]	**addrmap_col_b5**	**HIF Address Bit for Column Address Bit for Three Bus Width Modes** • Full bus width mode - Selects the HIF address bit used as column address bit 5 • Half bus width mode - Selects the HIF address bit used as column address bit 6 Valid Range: 0 to 7, and 15 Internal Base: 5 The selected HIF address bit is determined by adding the internal base to the value of this field. If unused, set to 15 and then this column address bit is set to 0. **Programming Mode:** Static
[23:20]	**Reserved**	Reserved.
[19:16]	**addrmap_col_b4**	**HIF Address Bit for Column Address Bit for Three Bus Width Modes** • Full bus width mode - Selects the HIF address bit used as column address bit 4 • Half bus width mode - Selects the HIF address bit used as column address bit 5 Valid Range: 0 to 7, and 15 Internal Base: 4 The selected HIF address bit is determined by adding the internal base to the value of this field. If unused, set to 15 and then this column address bit is set to 0. **Programming Mode:** Static
[15:13]	**Reserved**	Reserved.
[12:8]	**addrmap_col_b3**	**HIF Address Bit for Column Address Bit for Three Bus Width Modes** • Full bus width mode - Selects the HIF address bit used as column address bit 3 • Half bus width mode - Selects the HIF address bit used as column address bit 4 Valid Range: 0 to 7. Internal Base: 3 The selected HIF address bit is determined by adding the internal base to the value of this field. **Programming Mode:** Static

[7:4]	**Reserved**	Reserved.
[3:0]	**addrmap_col_b2**	**HIF Address Bit for Column Address Bit for Three Bus Width Modes** ● Full bus width mode - Selects the HIF address bit used as column address bit 2 ● Half bus width mode - Selects the HIF address bit used as column address bit 3 Valid Range: 0 to 7 Internal Base: 2 The selected HIF address bit is determined by adding the internal base to the value of this field. In our chip, we configure hardware parameter “MEMC_BURST_LENGTH” to equal to 8 and Full Bus Width (MSTR.data_bus_width==00), it is recommended to program this to 0 so that HIF[2] maps to column address bit 2. **Programming Mode:** Static

UMCTL2 Address Map Register 3 (ADDRMAP3)

Register	Offset	R/W	Description	Reset Value
ADDRMAP3	UMCTL2_BA+0x20c	R/W	Address Map Register 3	0x0000_0000

31	30	29	28	27	26	25	24
Reserved			addrmap_col_b9				
23	22	21	20	19	18	17	16
Reserved			addrmap_col_b8				
15	14	13	12	11	10	9	8
Reserved			addrmap_col_b7				
7	6	5	4	3	2	1	0
Reserved			addrmap_col_b6				

Bits	Description	
[31:29]	**Reserved**	Reserved.
[28:24]	**addrmap_col_b9**	**HIF Address Bit for Column Address Bit for Three Bus Width Modes** • Full bus width mode - Selects the HIF address bit used as column address bit 9 • Half bus width mode - Selects the HIF address bit used as column address bit 11 Valid Range: 0 to 7. Internal Base: 9 The selected HIF address bit is determined by adding the internal base to the value of this field. **Note:** Per JEDEC DDR2/3 specification, column address bit 10 is reserved for indicating auto-precharge, and hence no source address bit can be mapped to column address bit 10. If unused, set to 31 and then this column address bit is set to 0. **Programming Mode:** Static
[23:21]	**Reserved**	Reserved.
[20:16]	**addrmap_col_b8**	**HIF Address Bit for Column Address Bit for Three Bus Width Modes** • Full bus width mode - Selects the HIF address bit used as column address bit 8 • Half bus width mode - Selects the HIF address bit used as column address bit 9 Valid Range: 0 to 7, and 31. Internal Base: 8 The selected HIF address bit is determined by adding the internal base to the value of this field. **Note:** Per JEDEC DDR2/3 specification, column address bit 10 is reserved for indicating auto-precharge, and hence no source address bit can be mapped to column address bit 10. If unused, set to 31 and then this column address bit is set to 0. **Programming Mode:** Static
[15:13]	**Reserved**	Reserved.
[12:8]	**addrmap_col_b7**	**HIF Address Bit for Column Address Bit for Three Bus Width Modes** • Full bus width mode - Selects the HIF address bit used as column address bit 7 • Half bus width mode - Selects the HIF address bit used as column address bit 8

		Valid Range: 0 to 7, and 31. Internal Base: 7 The selected HIF address bit is determined by adding the internal base to the value of this field. If unused, set to 31 and then this column address bit is set to 0. **Programming Mode:** Static
[7:5]	**Reserved**	Reserved.
[4:0]	**addrmap_col_b6**	**HIF Address Bit for Column Address Bit for Three Bus Width Modes** ● Full bus width mode - Selects the HIF address bit used as column address bit 6. ● Half bus width mode - Selects the HIF address bit used as column address bit 7. Valid Range: 0 to 7, and 31. Internal Base: 6 The selected HIF address bit is determined by adding the internal base to the value of this field. If unused, set to 31 and then this column address bit is set to 0. **Programming Mode:** Static

UMCTL2 Address Map Register 4 (ADDRMAP4)

Register	Offset	R/W	Description	Reset Value
ADDRMAP4	UMCTL2_BA+0x210	R/W	Address Map Register 4	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			**addrmap_col_b11**				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**addrmap_col_b10**				

Bits	Description	
[31:13]	**Reserved**	Reserved.
[12:8]	**addrmap_col_b11**	**HIF Address Bit for Column Address Bit for Three Bus Width Modes** • Full bus width mode - Selects the HIF address bit used as column address bit 13 • Half bus width mode - UNUSED. See later in this description for value you need to set to make it unused Valid Range: 0 to 7, and 31. Internal Base: 11 The selected HIF address bit is determined by adding the internal base to the value of this field. **Note:** Per JEDEC DDR2/3 specification, column address bit 10 is reserved for indicating auto-precharge, and hence no source address bit can be mapped to column address bit 10. If unused, set to 31 and then this column address bit is set to 0. **Programming Mode:** Static
[7:5]	**Reserved**	Reserved.
[4:0]	**addrmap_col_b10**	**HIF Address Bit for Column Address Bit for Three Bus Width Modes** • Full bus width mode: Selects the HIF address bit used as column address bit 11 • Half bus width mode: Selects the HIF address bit used as column address bit 13 Valid Range: 0 to 7, and 31. Internal Base: 10 The selected HIF address bit is determined by adding the internal base to the value of this field. **Note:** Per JEDEC DDR2/3 specification, column address bit 10 is reserved for indicating auto-precharge, and hence no source address bit can be mapped to column address bit 10. If unused, set to 31 and then this column address bit is set to 0. **Programming Mode:** Static

UMCTL2 Address Map Register 5 (ADDRMAP5)

Register	Offset	R/W	Description	Reset Value
ADDRMAP5	UMCTL2_BA+0x214	R/W	Address Map Register 5	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				addrmap_row_b11			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				addrmap_row_b2_10			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				addrmap_row_b1			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				addrmap_row_b0			

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:24]	**addrmap_row_b11**	**HIF Address Bits for Row Address Bit 11** Valid Range: 0 to 11, and 15 Internal Base: 17 The selected HIF address bit is determined by adding the internal base to the value of this field. If unused, set to 15 and then row address bit 11 is set to 0. **Programming Mode:** Static
[23:20]	**Reserved**	Reserved.
[19:16]	**addrmap_row_b2_10**	**HIF Address Bits for Row Address Bits 2 to 10** Valid Range: 0 to 11, and 15 Internal Base: 8 (for row address bit 2), 9 (for row address bit 3), 10 (for row address bit 4) and so on, increasing to 16 (for row address bit 10) The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. When set to 15, the values of row address bits 2 to 10 are defined by registers ADDRMAP9, ADDRMAP10, ADDRMAP11. **Programming Mode:** Static
[15:12]	**Reserved**	Reserved.
[11:8]	**addrmap_row_b1**	**HIF Address Bits for Row Address Bit 1** Valid Range: 0 to 11 Internal Base: 7 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. **Programming Mode:** Static
[7:4]	**Reserved**	Reserved.
[3:0]	**addrmap_row_b0**	**HIF Address Bits for Row Address Bit 0** Valid Range: 0 to 11

		Internal Base: 6 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. **Programming Mode:** Static

UMCTL2 Address Map Register 6 (ADDRMAP6)

Register	Offset	R/W	Description	Reset Value
ADDRMAP6	UMCTL2_BA+0x218	R/W	Address Map Register 6	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				**addrmap_row_b15**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				**addrmap_row_b14**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**addrmap_row_b13**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**addrmap_row_b12**			

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:24]	**addrmap_row_b15**	**HIF Address Bit for Row Address Bit 15** Valid Range: 0 to 11, and 15 Internal Base: 21 The selected HIF address bit is determined by adding the internal base to the value of this field. If unused, set to 15 and then row address bit 15 is set to 0. **Programming Mode:** Static
[23:20]	**Reserved**	Reserved.
[19:16]	**addrmap_row_b14**	**HIF Address Bit for Row Address Bit 14** Valid Range: 0 to 11, and 15 Internal Base: 20 The selected HIF address bit is determined by adding the internal base to the value of this field. If unused, set to 15 and then row address bit 14 is set to 0. **Programming Mode:** Static
[15:12]	**Reserved**	Reserved.
[11:8]	**addrmap_row_b13**	**HIF Address Bit for Row Address Bit 13** Valid Range: 0 to 11, and 15 Internal Base: 19 The selected HIF address bit is determined by adding the internal base to the value of this field. If unused, set to 15 and then row address bit 13 is set to 0. **Programming Mode:** Static
[7:4]	**Reserved**	Reserved.
[3:0]	**addrmap_row_b12**	**HIF Address Bit for Row Address Bit 12** Valid Range: 0 to 11, and 15 Internal Base: 18 The selected HIF address bit is determined by adding the internal base to the value of this field.

		If unused, set to 15 and then row address bit 12 is set to 0. **Programming Mode:** Static

UMCTL2 Address Map Register 9 (ADDRMAP9)

Register	Offset	R/W	Description	Reset Value
ADDRMAP9	UMCTL2_BA+0x224	R/W	Address Map Register 9	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				addrmap_row_b5			
23	22	21	20	19	18	17	16
Reserved				addrmap_row_b4			
15	14	13	12	11	10	9	8
Reserved				addrmap_row_b3			
7	6	5	4	3	2	1	0
Reserved				addrmap_row_b2			

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:24]	**addrmap_row_b5**	**HIF Address Bits for Row Address Bit 5** Valid Range: 0 to 11 Internal Base: 11 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15. **Programming Mode:** Static
[23:20]	**Reserved**	Reserved.
[19:16]	**addrmap_row_b4**	**HIF Address Bits for Row Address Bit 4** Valid Range: 0 to 11 Internal Base: 10 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15. **Programming Mode:** Static
[15:12]	**Reserved**	Reserved.
[11:8]	**addrmap_row_b3**	**HIF Address Bits for Row Address Bit 3** Valid Range: 0 to 11 Internal Base: 9 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15. **Programming Mode:** Static
[7:4]	**Reserved**	Reserved.
[3:0]	**addrmap_row_b2**	**HIF Address Bits for Row Address Bit 2** Valid Range: 0 to 11 Internal Base: 8

		The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15. **Programming Mode:** Static

UMCTL2 Address Map Register 10 (ADDRMAP10)

Register	Offset	R/W	Description	Reset Value
ADDRMAP10	UMCTL2_BA+0x228	R/W	Address Map Register 10	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				**addrmap_row_b9**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				**addrmap_row_b8**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**addrmap_row_b7**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**addrmap_row_b6**			

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:24]	**addrmap_row_b9**	**HIF Address Bits for Row Address Bit 9** Valid Range: 0 to 11 Internal Base: 15 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15. **Programming Mode:** Static
[23:20]	**Reserved**	Reserved.
[19:16]	**addrmap_row_b8**	**HIF Address Bits for Row Address Bit 8** Valid Range: 0 to 11 Internal Base: 14 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15. **Programming Mode:** Static
[15:12]	**Reserved**	Reserved.
[11:8]	**addrmap_row_b7**	**HIF Address Bits for Row Address Bit 7** Valid Range: 0 to 11 Internal Base: 13 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15. **Programming Mode:** Static
[7:4]	**Reserved**	Reserved.
[3:0]	**addrmap_row_b6**	**HIF Address Bits for Row Address Bit 6** Valid Range: 0 to 11 Internal Base: 12

		The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15. **Programming Mode:** Static

UMCTL2 Address Map Register 11 (ADDRMAP11)

Register	Offset	R/W	Description	Reset Value
ADDRMAP11	UMCTL2_BA+0x22c	R/W	Address Map Register 11	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**addrmap_row_b10**			

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3:0]	**addrmap_row_b10**	**HIF Address Bits for Row Address Bit 10** Valid Range: 0 to 11 Internal Base: 16 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15. **Programming Mode:** Static

UMCTL2 ODT Configuration Register (ODTCFG)

Register	Offset	R/W	Description	Reset Value
ODTCFG	UMCTL2_BA+0x240	R/W	ODT Configuration Register	0x0400_0400

31	30	29	28	27	26	25	24
Reserved				wr_odt_hold			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			wr_odt_delay				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				rd_odt_hold			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	rd_odt_delay					Reserved	

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:24]	**wr_odt_hold**	**DFI PHY Clock Cycles to Hold ODT for a Write Command** The minimum supported value is 2. Recommended values: DDR2: • BL8 - 0x5 (DDR2-400/533/667), 0x6 (DDR2-800), 0x7 (DDR2-1066) • BL4 - 0x3 (DDR2-400/533/667), 0x4 (DDR2-800), 0x5 (DDR2-1066) DDR3: • BL8 - 0x6 Unit: DFI PHY clock cycles. **Programming Mode:** Quasi-dynamic Group 1, Group 4
[23:21]	**Reserved**	Reserved.
[20:16]	**wr_odt_delay**	**Delay in DFI PHY Clock Cycles from Issuing a Write Command to Setting ODT Values Associated with that Command** ODT setting must remain constant for the entire time that DQS is driven by the uMCTL2. Recommended values: DDR2: • CWL + AL - 3 (DDR2-400/533/667), • CWL + AL - 4 (DDR2-800), • CWL + AL - 5 (DDR2-1066) If (CWL + AL - 3 < 0), uMCTL2 does not support ODT for write operation. DDR3: • 0x0 Unit: DFI PHY clock cycles. **Programming Mode:** Quasi-dynamic Group 1, Group 4
[15:12]	**Reserved**	Reserved.
[11:8]	**rd_odt_hold**	**DFI PHY Clock Cycles to Hold ODT for a Read Command**

		The minimum supported value is 2. Recommended values: DDR2: • BL8 - 0x6 (not DDR2-1066), 0x7 (DDR2-1066) • BL4 - 0x4 (not DDR2-1066), 0x5 (DDR2-1066) DDR3: • BL8 - 0x6 Unit: DFI PHY clock cycles. **Programming Mode:** Quasi-dynamic Group 1, Group 4
[7]	**Reserved**	Reserved.
[6:2]	**rd_odt_delay**	**Delay in DFI PHY Clock Cycles from Issuing a Read Command to Setting ODT Values Associated with that Command** ODT setting must remain constant for the entire time that DQS is driven by the uMCTL2. Recommended values: DDR2: • CL + AL - 4 (not DDR2-1066), • CL + AL - 5 (DDR2-1066) If (CL + AL - 4 < 0), uMCTL2 does not support ODT for read operation. DDR3: • CL - CWL Unit: DFI PHY clock cycles. **Programming Mode:** Quasi-dynamic Group 1, Group 4
[1:0]	**Reserved**	Reserved.

UMCTL2 ODT/Rank Map Register (ODTMAP)

Register	Offset	R/W	Description	Reset Value
ODTMAP	UMCTL2_BA+0x244	R/W	ODT/Rank Map Register	0x0000_2211

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		rank1_rd_odt		Reserved		rank1_wr_odt	
7	6	5	4	3	2	1	0
Reserved		rank0_rd_odt		Reserved		rank0_wr_odt	

Bits	Description	
[31:14]	**Reserved**	Reserved.
[13:12]	**rank1_rd_odt**	**Remote ODTs Must be Turned on During a Read from Rank 1** Each rank has a remote ODT (in the SDRAM) which can be turned on by setting the appropriate bit here. Rank 0 is controlled by the LSB; rank 1 is controlled by bit next to the LSB, and so on. For each rank, set its bit to 1 to enable its ODT. Present only in configurations that have 2 or more ranks. **Programming Mode:** Static
[11:10]	**Reserved**	Reserved.
[9:8]	**rank1_wr_odt**	**Remote ODTs Must be Turned on During a Write to Rank 1** Each rank has a remote ODT (in the SDRAM) which can be turned on by setting the appropriate bit here. Rank 0 is controlled by the LSB; rank 1 is controlled by bit next to the LSB, and so on. For each rank, set its bit to 1 to enable its ODT. Present only in configurations that have 2 or more ranks. **Programming Mode:** Static
[7:6]	**Reserved**	Reserved.
[5:4]	**rank0_rd_odt**	**Remote ODTs Must be Turned on During a Read from Rank 0** Each rank has a remote ODT (in the SDRAM) which can be turned on by setting the appropriate bit here. Rank 0 is controlled by the LSB; rank 1 is controlled by bit next to the LSB, and so on. For each rank, set its bit to 1 to enable its ODT. **Programming Mode:** Static
[3:2]	**Reserved**	Reserved.
[1:0]	**rank0_wr_odt**	**Remote ODTs Must be Turned on During a Write to Rank 0** Each rank has a remote ODT (in the SDRAM) which can be turned on by setting the appropriate bit here.

		Rank 0 is controlled by the LSB; rank 1 is controlled by bit next to the LSB, and so on. For each rank, set its bit to 1 to enable its ODT. **Programming Mode:** Static

UMCTL2 Scheduler Control Register (SCHED)

Register	Offset	R/W	Description	Reset Value
SCHED	UMCTL2_BA+0x250	R/W	Scheduler Control Register	0x0000_1004

31	30	29	28	27	26	25	24
Reserved	**rdwr_idle_gap**						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
go2critical_hysteresis							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			**lpr_num_entries**				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**Pageclose**	**prefer_write**	**Reserved**

Bits	Description	
[31]	**Reserved**	Reserved.
[30:24]	**rdwr_idle_gap**	**Switch Clock Cycles Between Different Transaction Store** When the preferred transaction store is empty for these many clock cycles, switch to the alternate transaction store if it is non-empty. The read transaction store (both high and low priority) is the default preferred transaction store and the write transaction store is the alternative store. When prefer write over read is set this is reversed. 0x0 is a legal value for this register. When set to 0x0, the transaction store switching happens immediately when the switching conditions become true. FOR PERFORMANCE ONLY. Unit: DFI clock cycles. **Programming Mode:** Static
[23:16]	**go2critical_hysteresis**	**UNUSED** **Programming Mode:** Static
[15:13]	**Reserved**	Reserved.
[12:8]	**lpr_num_entries**	**Number of Entries in the Low Priority Transaction Store** It is this value + 1. (MEMC_NO_OF_ENTRY (hardware configures to 32) - (SCHED.lpr_num_entries + 1)) is the number of entries available for the high priority transaction store. Setting this to maximum value allocates all entries to low priority transaction store. Setting this to 0 allocates 1 entry to low priority transaction store and the rest to high priority transaction store. **Programming Mode:** Static
[7:3]	**Reserved**	Reserved.
[2]	**Pageclose**	**Page Close Bit** If true, bank is kept open only while there are page hit transactions available in the CAM to that bank. The last read or write command in the CAM with a bank and page hit is executed with auto-precharge if SCHED1.pageclose_timer=0.

		Even if this register set to 1 and SCHED1.pageclose_timer is set to 0, explicit precharge (and not auto-precharge) may be issued in some cases where there is a mode switch between Write and Read or between LPR and HPR. The Read and Write commands that are executed as part of the ECC scrub requests are also executed without auto-precharge. If false, the bank remains open until there is a need to close it (to open a different page, or for page timeout or refresh timeout) - also known as open page policy. The open page policy can be overridden by setting the per-command-autopre bit on the HIF interface (hif_cmd_autopre). The pageclose feature provids a midway between Open and Close page policies. FOR PERFORMANCE ONLY. **Programming Mode:** Quasi-dynamic Group 3
[1]	**prefer_write**	**Setting Bit for Bank Selector Prefers Writes over Reads** If set, then the bank selector prefers writes over reads. FOR DEBUG ONLY. **Programming Mode:** Static
[0]	**Reserved**	Reserved.

UMCTL2 Scheduler Control Register 1 (SCHED1)

Register	Offset	R/W	Description	Reset Value
SCHED1	UMCTL2_BA+0x254	R/W	Scheduler Control Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
pageclose_timer							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**pageclose_timer**	**Page Close Timer** This field works in conjunction with SCHED.pageclose. It only has meaning if SCHED.pageclose equals to 1. If SCHED.pageclose==1 and pageclose_timer==0, then an auto-precharge may be scheduled for last read or write command in the CAM with a bank and page hit. Note, sometimes an explicit precharge is scheduled instead of the auto-precharge. For more information, see SCHED.pageclose. If SCHED.pageclose==1 and pageclose_timer>0, then an auto-precharge is not scheduled for last read or write command in the CAM with a bank and page hit. Instead, a timer is started, with pageclose_timer as the initial value. There is a timer on a per bank basis. The timer decrements unless the next read or write in the CAM to a bank is a page hit. It gets reset to pageclose_timer value if the next read or write in the CAM to a bank is a page hit. Once the timer has reached zero, an explicit precharge is attempted to be scheduled. Unit: DFI clock cycles. **Programming Mode:** Static

UMCTL2 High Priority Read CAM Register 1 (PERFHPR1)

Register	Offset	R/W	Description	Reset Value
PERFHPR1	UMCTL2_BA+0x25c	R/W	High Priority Read CAM Register 1	0x0f00_0001

31	30	29	28	27	26	25	24
hpr_xact_run_length							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
hpr_max_starve							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
hpr_max_starve							

Bits	Description	
[31:24]	**hpr_xact_run_length**	**Number of Transactions that are Serviced once the HPR Queue Goes Critical** Indicates the number of transactions that are serviced once the HPR queue goes critical is the smaller of: (a) This number (b) Number of transactions available Unit: Transaction. FOR PERFORMANCE ONLY. **Programming Mode:** Quasi-dynamic Group 3
[23:16]	**Reserved**	Reserved.
[15:0]	**hpr_max_starve**	**Number of DFI Clocks that the HPR Queue can be Starved Before It Goes Critical** Indicates the number of DFI clocks that the HPR queue can be starved before it goes critical. The minimum valid functional value for this register is 0x1. Programming it to 0x0 disables the starvation functionality. During normal operation, this function must not be disabled as it causes excessive latencies. FOR PERFORMANCE ONLY. Unit: DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 3

UMCTL2 Low Priority Read CAM Register 1 (PERFLPR1)

Register	Offset	R/W	Description	Reset Value
PERFLPR1	UMCTL2_BA+0x264	R/W	Low Priority Read CAM Register 1	0x0f00_007f

31	30	29	28	27	26	25	24
lpr_xact_run_length							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
lpr_max_starve							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
lpr_max_starve							

Bits	Description	
[31:24]	**lpr_xact_run_length**	**Number of Transactions that are Serviced Once the LPR Queue Goes Critical** Indicates the number of transactions that are serviced once the LPR queue goes critical is the smaller of: (a) This number (b) Number of transactions available Unit: Transaction. FOR PERFORMANCE ONLY. **Programming Mode:** Quasi-dynamic Group 3
[23:16]	**Reserved**	Reserved.
[15:0]	**lpr_max_starve**	**Number of DFI Clocks that the LPR Queue can be Starved Before It Goes Critical** Indicates the number of DFI clocks that the LPR queue can be starved before it goes critical. The minimum valid functional value for this register is 0x1. Programming it to 0x0 disables the starvation functionality. During normal operation, this function must not be disabled as it causes excessive latencies. FOR PERFORMANCE ONLY. Unit: DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 3

UMCTL2 Write CAM Register 1 (PERFWR1)

Register	Offset	R/W	Description	Reset Value
PERFWR1	UMCTL2_BA+0x26c	R/W	Write CAM Register 1	0x0f00_007f

31	30	29	28	27	26	25	24
w_xact_run_length							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
w_max_starve							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
w_max_starve							

Bits	Description	
[31:24]	**w_xact_run_length**	**Number of Transactions that are Serviced Once the WR Queue Goes Critical** Indicates the number of transactions that are serviced once the WR queue goes critical is the smaller of: (a) This number (b) Number of transactions available Unit: Transaction. FOR PERFORMANCE ONLY. **Programming Mode:** Quasi-dynamic Group 3
[23:16]	**Reserved**	Reserved.
[15:0]	**w_max_starve**	**Number of DFI Clocks that the WR Queue can be Starved Before It Goes Critical** Indicates the number of DFI clocks that the WR queue can be starved before it goes critical. The minimum valid functional value for this register is 0x1. Programming it to 0x0 disables the starvation functionality. During normal operation, this function must not be disabled as it causes excessive latencies. FOR PERFORMANCE ONLY. Unit: DFI clock cycles. **Programming Mode:** Quasi-dynamic Group 3

UMCTL2 Debug Register 0 (DBG0)

Register	Offset	R/W	Description	Reset Value
DBG0	UMCTL2_BA+0x300	R/W	Debug Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
dis_max_rank_wr_opt	**dis_max_rank_rd_opt**	**Reserved**	**dis_collision_page_opt**	**Reserved**	**dis_act_bypass**	**dis_rd_bypass**	**dis_wc**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**dis_max_rank_wr_opt**	**Disable Bit to Optimize max_rank_wr and max_logical_rank_wr** This register is for debug purpose only. For normal operation, this register must be set to 0. **Programming Mode:** Static
[6]	**dis_max_rank_rd_opt**	**Disable Bit to Optimize max_rank_rd and max_logical_rank_rd** This register is for debug purpose only. For normal operation, this register must be set to 0. **Programming Mode:** Static
[5]	**Reserved**	Reserved.
[4]	**dis_collision_page_opt**	**Disable Bit for Collision Page Option** When this is set to '0', auto-precharge is disabled for the flushed command in a collision case. Collision cases are write followed by read to same address, read followed by write to same address, or write followed by write to same address with DBG0.dis_wc bit = 1 (where same address comparisons exclude the two address bits representing critical word). FOR DEBUG ONLY. **Programming Mode:** Static
[3]	**Reserved**	Reserved.
[2]	**dis_act_bypass**	**Disable Bit for Bypass Path for High Priority Read Activates** Only present in designs supporting activate bypass. When 1, disable bypass path for high priority read activates. FOR DEBUG ONLY. **Programming Mode:** Static
[1]	**dis_rd_bypass**	**Disable Bit for Bypass Path for High Priority Read Page Hits** Only present in designs supporting read bypass. When 1, disable bypass path for high priority read page hits. FOR DEBUG ONLY.

		Programming Mode: Static
[0]	**dis_wc**	**Disable Bit for Write Combine** When 1, disable write combine. FOR DEBUG ONLY. **Programming Mode:** Static

UMCTL2 Debug Register 1 (DBG1)

Register	Offset	R/W	Description	Reset Value
DBG1	UMCTL2_BA+0x304	R/W	Debug Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**dis_hif**	**dis_dq**

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**dis_hif**	**Disable Bit for HIF Command** When 1, uMCTL2 asserts the HIF command signal hif_cmd_stall. uMCTL2 ignores the hif_cmd_valid and all other associated request signals. This bit is intended to be switched on-the-fly. **Programming Mode:** Dynamic
[0]	**dis_dq**	**Disable Bit for De-queue Transactions** When 1, uMCTL2 does not de-queue any transactions from the CAM. Bypass is also disabled. All transactions are queued in the CAM. No reads or writes are issued to SDRAM as long as this is asserted. This bit may be used to prevent reads or writes being issued by the uMCTL2, which makes it safe to modify certain register fields associated with reads and writes. After setting this bit, it is strongly recommended to poll DBGCAM.wr_data_pipeline_empty and DBGCAM.rd_data_pipeline_empty, before making changes to any registers which affect reads and writes. This ensures that the relevant logic in the DDRC is idle. This bit is intended to be switched on-the-fly. **Programming Mode:** Dynamic

UMCTL2 CAM Debug Register (DBGCAM)

Register	Offset	R/W	Description	Reset Value
DBGCAM	UMCTL2_BA+0x308	R	CAM Debug Register	0x3600_0000

31	30	29	28	27	26	25	24
Reserved		wr_data_pipeline_empty	rd_data_pipeline_empty	Reserved	dbg_wr_q_empty	dbg_rd_q_empty	dbg_stall
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		dbg_w_q_depth					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		dbg_lpr_q_depth					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		dbg_hpr_q_depth					

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29]	**wr_data_pipeline_empty**	**Write Data Pipeline Empty** This bit indicates that the write data pipeline on the DFI interface is empty. This register is intended to be polled at least twice after setting DBG1.dis_dq, to ensure that all remaining commands/data have completed. **Programming Mode:** Dynamic
[28]	**rd_data_pipeline_empty**	**Read Data Pipeline Empty** This bit indicates that the read data pipeline on the DFI interface is empty. This register is intended to be polled at least twice after setting DBG1.dis_dq, to ensure that all remaining commands/data have completed. **Programming Mode:** Dynamic
[27]	**Reserved**	Reserved.
[26]	**dbg_wr_q_empty**	**Write Command Queues and Write Data Buffers Empty** When 1, all the Write command queues and Write data buffers inside DDRC are empty. This register is to be used for debug purpose. An example use-case scenario: When the controller enters self-refresh using the Low-Power entry sequence, controller is expected to have executed all the commands in its queues and the write and read data drained. Hence this register must be 1 at that time. FOR DEBUG ONLY **Programming Mode:** Dynamic
[25]	**dbg_rd_q_empty**	**Read Command Queues and Read Data Buffers Empty** When 1, all the Read command queues and Read data buffers inside DDRC are empty. This register is to be used for debug purpose. An example use-case scenario: When the controller enters self-refresh using the Low-Power entry sequence, controller is expected to have executed all the commands in its queues and the write and read data drained. Hence this register must be 1 at that time. FOR DEBUG ONLY **Programming Mode:** Dynamic

[24]	**dbg_stall**	**Command Queues and Data Buffers Stall** FOR DEBUG ONLY. **Programming Mode:** Dynamic
[23:22]	**Reserved**	Reserved.
[21:16]	**dbg_w_q_depth**	**Write Queue Depth** The last entry of WR queue is reserved for ECC SCRUB operation. This entry is not included in the calculation of the queue depth. FOR DEBUG ONLY **Programming Mode:** Dynamic
[15:14]	**Reserved**	Reserved.
[13:8]	**dbg_lpr_q_depth**	**Low Priority Read Queue Depth** The last entry of LPR queue is reserved for ECC SCRUB operation. This entry is not included in the calculation of the queue depth. FOR DEBUG ONLY **Programming Mode:** Dynamic
[7:6]	**Reserved**	Reserved.
[5:0]	**dbg_hpr_q_depth**	**High Priority Read Queue Depth** FOR DEBUG ONLY. **Programming Mode:** Dynamic

UMCTL2 Command Debug Register (DBGCMD)

Register	Offset	R/W	Description	Reset Value
DBGCMD	UMCTL2_BA+0x30c	R/W	Command Debug Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		ctrlupd	zq_calib_short	Reserved		rank1_refresh	rank0_refresh

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**ctrlupd**	**Signal for uMCTL2 to Issue dfi_ctrlupd_req to PHY** Setting this register bit to 1 indicates to the uMCTL2 to issue a dfi_ctrlupd_req to the PHY. When this request is stored in the uMCTL2, the bit is automatically cleared. This operation must only be performed when DFIUPD0.dis_auto_ctrlupd=1. **Testable:** readOnly **Programming Mode:** Dynamic
[4]	**zq_calib_short**	**Signal for uMCTL2 to Issue ZQCS to SDRAM** Setting this register bit to 1 indicates to the uMCTL2 to issue a ZQCS (ZQ calibration short)/MPC(ZQ calibration) command to the SDRAM. When this request is stored in the uMCTL2, the bit is automatically cleared. This operation can be performed only when ZQCTL0.dis_auto_zq=1. It is recommended NOT to set this register bit if in Init, in self-refresh or Deep power-down operating modes or Maximum Power Saving Mode. For Deep power down and Maximum Power Saving Mode, it is not scheduled, although DBGSTAT.zq_calib_short_busy is de-asserted. **Testable:** readOnly **Programming Mode:** Dynamic
[3:2]	**Reserved**	Reserved.
[1]	**rank1_refresh**	**Signal for uMCTL2 to Issue Refresh to Rank 1** Setting this register bit to 1 indicates to the uMCTL2 to issue a refresh to rank 1. Writing to this bit causes DBGSTAT.rank1_refresh_busy to be set. When DBGSTAT.rank1_refresh_busy is cleared, the command has been stored in uMCTL2. For 3DS configuration, refresh is sent to rank index 1. This operation can be performed only when RFSHCTL3.dis_auto_refresh=1. It is recommended NOT to set this register bit if in Init or Deep power-down operating modes or Maximum Power Saving Mode. **Testable:** readOnly

		Programming Mode: Dynamic
[0]	**rank0_refresh**	**Signal for uMCTL2 to Issue Refresh to Rank 0** Setting this register bit to 1 indicates to the uMCTL2 to issue a refresh to rank 0. Writing to this bit causes DBGSTAT.rank0_refresh_busy to be set. When DBGSTAT.rank0_refresh_busy is cleared, the command has been stored in uMCTL2. For 3DS configuration, refresh is sent to rank index 0. This operation can be performed only when RFSHCTL3.dis_auto_refresh=1. It is recommended NOT to set this register bit if in Init or Deep power-down operating modes or Maximum Power Saving Mode. **Testable:** readOnly **Programming Mode:** Dynamic

UMCTL2 Status Debug Register (DBGSTAT)

Register	Offset	R/W	Description	Reset Value
DBGSTAT	UMCTL2_BA+0x310	R	Status Debug Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		ctrlupd_busy	zq_calib_short_busy	Reserved		rank1_refresh_busy	rank0_refresh_busy

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**ctrlupd_busy**	**ctrlupd Operation Busy State** SoC might initiate a ctrlupd operation only if this signal is low. This signal goes high in the clock after the uMCTL2 accepts the ctrlupd request. It goes low when the ctrlupd operation is initiated in the uMCTL2. It is recommended not to perform ctrlupd operations when this signal is high. 0 = Indicates that the SoC can initiate a ctrlupd operation. 1 = Indicates that ctrlupd operation has not been initiated yet in the uMCTL2. **Programming Mode:** Dynamic
[4]	**zq_calib_short_busy**	**ZQCS Operation Busy State** SoC might initiate a ZQCS (ZQ calibration short) operation only if this signal is low. This signal goes high in the clock after the uMCTL2 accepts the ZQCS request. It goes low when the ZQCS operation is initiated in the uMCTL2. It is recommended not to perform ZQCS operations when this signal is high. 0 = Indicates that the SoC can initiate a ZQCS operation. 1 = Indicates that ZQCS operation has not been initiated yet in the uMCTL2. **Programming Mode:** Dynamic
[3:2]	**Reserved**	Reserved.
[1]	**rank1_refresh_busy**	**Rank1 Refresh Operation Busy** SoC might initiate a rank1_refresh operation (refresh operation to rank 1) only if this signal is low. This signal goes high in the clock after DBGCMD.rank1_refresh is set to one. It goes low when the rank1_refresh operation is stored in the uMCTL2. It is recommended not to perform rank1_refresh operations when this signal is high. 0 = Indicates that the SoC can initiate a rank1_refresh operation. 1 = Indicates that rank1_refresh operation has not been stored yet in the uMCTL2. **Programming Mode:** Dynamic
[0]	**rank0_refresh_busy**	**Rank0 Refresh Operation Busy**

		SoC might initiate a rank0_refresh operation (refresh operation to rank 0) only if this signal is low. This signal goes high in the clock after DBGCMD.rank0_refresh is set to one. It goes low when the rank0_refresh operation is stored in the uMCTL2. It is recommended not to perform rank0_refresh operations when this signal is high. 0 = Indicates that the SoC can initiate a rank0_refresh operation. 1 = Indicates that rank0_refresh operation has not been stored yet in the uMCTL2. **Programming Mode:** Dynamic

UMCTL2 Software Register Programming Control Enable (SWCTL)

Register	Offset	R/W	Description	Reset Value
SWCTL	UMCTL2_BA+0x320	R/W	Software Register Programming Control Enable	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**sw_done**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**sw_done**	**Quasi-Dynamic Register Programming Enable Bit** Enables quasi-dynamic register programming outside reset. Program this register to 0 to enable quasi-dynamic programming. Set back register to 1 once programming is done. **Programming Mode:** Dynamic

UMCTL2 Software Register Programming Control Status (SWSTAT)

Register	Offset	R/W	Description	Reset Value
SWSTAT	UMCTL2_BA+0x324	R	Software Register Programming Control Status	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							sw_done_ack

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**sw_done_ack**	**Register Programming Done ACK** This register is the echo of SWCTL.sw_done. Wait for sw_done value 1 to propagate to sw_done_ack at the end of the programming sequence to ensure that the correct registers values are propagated to the destination clock domains. **Testable:** untestable **Programming Mode:** Static

UMCTL2 Static Registers Write Enable (SWCTLSTATIC)

Register	Offset	R/W	Description	Reset Value
SWCTLSTATIC	UMCTL2_BA+0x328	R/W	Static Registers Write Enable	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**sw_static_unlock**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**sw_static_unlock**	**Static Register Programming Unlock Bit** Enables static register programming outside reset. Program this register to 1 to enable static register programming. Set register back to 0 once programming is done. **Programming Mode:** Dynamic

UMCTL2 AXI Poison Configuration Register (POISONCFG)

Register	Offset	R/W	Description	Reset Value
POISONCFG	UMCTL2_BA+0x36c	R/W	AXI Poison Configuration Register. Common for all AXI ports.	0x0011_0011

31	30	29	28	27	26	25	24
Reserved							rd_poison_intr_clr
23	22	21	20	19	18	17	16
Reserved			rd_poison_intr_en	Reserved			rd_poison_slverr_en
15	14	13	12	11	10	9	8
Reserved							wr_poison_intr_clr
7	6	5	4	3	2	1	0
Reserved			wr_poison_intr_en	Reserved			wr_poison_slverr_en

Bits	Description	
[31:25]	**Reserved**	Reserved.
[24]	**rd_poison_intr_clr**	**Read Transaction Poisoning Interrupt Clear Bit** Allow 2/3 clock cycles for correct value to propagate to controller logic and clear the interrupts. uMCTL2 automatically clears this bit. **Testable:** readOnly **Programming Mode:** Dynamic
[23:21]	**Reserved**	Reserved.
[20]	**rd_poison_intr_en**	**Read Transaction Poisoning Interrupt Enable Bit** If set to 1, enables interrupts for read transaction poisoning. **Programming Mode:** Dynamic
[19:17]	**Reserved**	Reserved.
[16]	**rd_poison_slverr_en**	**Read Transaction Poisoning SLVERR Response Enable Bit** If set to 1, enables SLVERR response for read transaction poisoning. **Programming Mode:** Dynamic
[15:9]	**Reserved**	Reserved.
[8]	**wr_poison_intr_clr**	**Write Transaction Poisoning Interrupt Clear Bit** Interrupt clear for write transaction poisoning. Allow 2/3 clock cycles for correct value to propagate to controller logic and clear the interrupts. uMCTL2 automatically clears this bit. **Testable:** readOnly **Programming Mode:** Dynamic
[7:5]	**Reserved**	Reserved.
[4]	**wr_poison_intr_en**	**Write Transaction Poisoning Interrupt Enable Bit** If set to 1, enables interrupts for write transaction poisoning. **Programming Mode:** Dynamic

[3:1]	**Reserved**	Reserved.
[0]	**wr_poison_slverr_en**	**Write Transaction Poisoning SLVERR Response Enable Bit** If set to 1, enables SLVERR response for write transaction poisoning. **Programming Mode:** Dynamic

UMCTL2 AXI Poison Status Register (POISONSTAT)

Register	Offset	R/W	Description	Reset Value
POISONSTAT	UMCTL2_BA+0x370	R	AXI Poison Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
rd_poison_intr_7	rd_poison_intr_6	rd_poison_intr_5	rd_poison_intr_4	rd_poison_intr_3	rd_poison_intr_2	rd_poison_intr_1	rd_poison_intr_0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
wr_poison_intr_7	wr_poison_intr_6	wr_poison_intr_5	wr_poison_intr_4	wr_poison_intr_3	wr_poison_intr_2	wr_poison_intr_1	wr_poison_intr_0

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23]	**rd_poison_intr_7**	**Read Transaction Poisoning Error Interrupt for Port 7** This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's read address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register rd_poison_intr_clr, then value propagated to APB clock. **Programming Mode:** Dynamic
[22]	**rd_poison_intr_6**	**Read Transaction Poisoning Error Interrupt for Port 6** This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's read address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register rd_poison_intr_clr, then value propagated to APB clock. **Programming Mode:** Dynamic
[21]	**rd_poison_intr_5**	**Read Transaction Poisoning Error Interrupt for Port 5** This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's read address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register rd_poison_intr_clr, then value propagated to APB clock. **Programming Mode:** Dynamic
[20]	**rd_poison_intr_4**	**Read Transaction Poisoning Error Interrupt for Port 4** This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's read address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register rd_poison_intr_clr, then value propagated to APB clock. **Programming Mode:** Dynamic
[19]	**rd_poison_intr_3**	**Read Transaction Poisoning Error Interrupt for Port 3** This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's read address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register rd_poison_intr_clr, then value propagated to APB clock.

		Programming Mode: Dynamic
[18]	**rd_poison_intr_2**	**Read Transaction Poisoning Error Interrupt for Port 2** This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's read address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register rd_poison_intr_clr, then value propagated to APB clock. **Programming Mode:** Dynamic
[17]	**rd_poison_intr_1**	**Read Transaction Poisoning Error Interrupt for Port 1** This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's read address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register rd_poison_intr_clr, then value propagated to APB clock. **Programming Mode:** Dynamic
[16]	**rd_poison_intr_0**	**Read Transaction Poisoning Error Interrupt for Port 0** This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's read address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register rd_poison_intr_clr, then value propagated to APB clock. **Programming Mode:** Dynamic
[15:8]	**Reserved**	Reserved.
[7]	**wr_poison_intr_7**	**Write Transaction Poisoning Error Interrupt for Port 7** This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's write address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register wr_poison_intr_clr, then value propagated to APB clock. **Programming Mode:** Dynamic
[6]	**wr_poison_intr_6**	**Write Transaction Poisoning Error Interrupt for Port 6** This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's write address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register wr_poison_intr_clr, then value propagated to APB clock. **Programming Mode:** Dynamic
[5]	**wr_poison_intr_5**	**Write Transaction Poisoning Error Interrupt for Port 5** This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's write address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register wr_poison_intr_clr, then value propagated to APB clock. **Programming Mode:** Dynamic
[4]	**wr_poison_intr_4**	**Write Transaction Poisoning Error Interrupt for Port 4** This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's write address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register wr_poison_intr_clr, then value propagated to APB clock. **Programming Mode:** Dynamic
[3]	**wr_poison_intr_3**	**Write Transaction Poisoning Error Interrupt for Port 3** This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's write address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register wr_poison_intr_clr, then value propagated to APB clock. **Programming Mode:** Dynamic

[2]	**wr_poison_intr_2**	**Write Transaction Poisoning Error Interrupt for Port 2** This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's write address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register wr_poison_intr_clr, then value propagated to APB clock. **Programming Mode:** Dynamic
[1]	**wr_poison_intr_1**	**Write Transaction Poisoning Error Interrupt for Port 1** This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's write address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register wr_poison_intr_clr, then value propagated to APB clock. **Programming Mode:** Dynamic
[0]	**wr_poison_intr_0**	**Write Transaction Poisoning Error Interrupt for Port 0** This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's write address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register wr_poison_intr_clr, then value propagated to APB clock. **Programming Mode:** Dynamic

UMCTL2 Port Status Register (PSTAT)

Register	Offset	R/W	Description	Reset Value
PSTAT	UMCTL2_BA+0x3fc	R	Port Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
wr_port_busy_7	wr_port_busy_6	wr_port_busy_5	wr_port_busy_4	wr_port_busy_3	wr_port_busy_2	wr_port_busy_1	wr_port_busy_0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
rd_port_busy_7	rd_port_busy_6	rd_port_busy_5	rd_port_busy_4	rd_port_busy_3	rd_port_busy_2	rd_port_busy_1	rd_port_busy_0

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23]	**wr_port_busy_7**	**Outstanding Writes for AXI Port 7** Indicates if there are outstanding writes for AXI port 7. **Programming Mode:** Dynamic
[22]	**wr_port_busy_6**	**Outstanding Writes for AXI Port 6** Indicates if there are outstanding writes for AXI port 6. **Programming Mode:** Dynamic
[21]	**wr_port_busy_5**	**Outstanding Writes for AXI Port 5** Indicates if there are outstanding writes for AXI port 5. **Programming Mode:** Dynamic
[20]	**wr_port_busy_4**	**Outstanding Writes for AXI Port 4** Indicates if there are outstanding writes for AXI port 4. **Programming Mode:** Dynamic
[19]	**wr_port_busy_3**	**Outstanding Writes for AXI Port 3** Indicates if there are outstanding writes for AXI port 3. **Programming Mode:** Dynamic
[18]	**wr_port_busy_2**	**Outstanding Writes for AXI Port 2** Indicates if there are outstanding writes for AXI port 2. **Programming Mode:** Dynamic
[17]	**wr_port_busy_1**	**Outstanding Writes for AXI Port 1** Indicates if there are outstanding writes for AXI port 1. **Programming Mode:** Dynamic
[16]	**wr_port_busy_0**	**Outstanding Writes for AXI Port 0** Indicates if there are outstanding writes for AXI port 0. **Programming Mode:** Dynamic

[15:8]	**Reserved**	Reserved.
[7]	**rd_port_busy_7**	**Outstanding Reads for AXI Port 7** Indicates if there are outstanding reads for AXI port 7. **Programming Mode:** Dynamic
[6]	**rd_port_busy_6**	**Outstanding Reads for AXI Port 6** Indicates if there are outstanding reads for AXI port 6. **Programming Mode:** Dynamic
[5]	**rd_port_busy_5**	**Outstanding Reads for AXI Port 5** Indicates if there are outstanding reads for AXI port 5. **Programming Mode:** Dynamic
[4]	**rd_port_busy_4**	**Outstanding Reads for AXI Port 4** Indicates if there are outstanding reads for AXI port 4. **Programming Mode:** Dynamic
[3]	**rd_port_busy_3**	**Outstanding Reads for AXI Port 3** Indicates if there are outstanding reads for AXI port 3. **Programming Mode:** Dynamic
[2]	**rd_port_busy_2**	**Outstanding Reads for AXI Port 2** Indicates if there are outstanding reads for AXI port 2. **Programming Mode:** Dynamic
[1]	**rd_port_busy_1**	**Outstanding Reads for AXI Port 1** Indicates if there are outstanding reads for AXI port 1. **Programming Mode:** Dynamic
[0]	**rd_port_busy_0**	**Outstanding Reads for AXI Port 0** Indicates if there are outstanding reads for AXI port 0. **Programming Mode:** Dynamic

UMCTL2 Port Common Configuration Register (PCCFG)

Register	Offset	R/W	Description	Reset Value
PCCFG	UMCTL2_BA+0x400	R/W	Port Common Configuration Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							bl_exp_mode
7	6	5	4	3	2	1	0
Reserved			pagematch_limit	Reserved			go2critical_en

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8]	**bl_exp_mode**	**Burst Length Expansion Mode** By default, (that is, bl_exp_mode==0) XPI expands every AXI burst into multiple HIF commands, using the memory burst length as a unit. If set to 1, then XPI uses half of the memory burst length as a unit. This applies to both reads and writes. When MSTR.data_bus_width==00, setting bl_exp_mode to 1 has no effect. Functionality is also not supported if Data Channel Interleave is enabled. **Programming Mode:** Static
[7:5]	**Reserved**	Reserved.
[4]	**pagematch_limit**	**Page Match Four Limit** If set to 1, limits the number of consecutive same page DDRC transactions that can be granted by the Port Arbiter to four when Page Match feature is enabled. If set to 0, there is no limit imposed on number of consecutive same page DDRC transactions. **Programming Mode:** Static
[3:1]	**Reserved**	Reserved.
[0]	**go2critical_en**	**go2critical Function Enable Bit** If set to 1 (enabled), sets co_gs_go2critical_wr and co_gs_go2critical_lpr/co_gs_go2critical_hpr signals going to DDRC based on urgent input (awurgent, arurgent) coming from AXI master. If set to 0 (disabled), co_gs_go2critical_wr and co_gs_go2critical_lpr/co_gs_go2critical_hpr signals at DDRC are driven to 1b'0. For uPCTL2, this register field must be set to 0. **Programming Mode:** Static

UMCTL2 Port 0 Configuration Read Register (PCFGR_0)

Register	Offset	R/W	Description	Reset Value
PCFGR_0	UMCTL2_BA+0x404	R/W	Port 0 Configuration Read Register	0x0000_4000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							**rdwr_ordered_en**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**rd_port_pagematch_en**	**rd_port_urgent_en**	**rd_port_aging_en**	**Reserved**		**rd_port_priority**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
rd_port_priority							

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**rdwr_ordered_en**	**Read/Writes Ordered Enable Bit** If set to 1, preserves the ordering between read transaction and write transaction issued to the same address, on a given port. In other words, the controller ensures that all same address read and write commands from the application port interface are transported to the DFI interface in the order of acceptance. This feature is useful in cases where software coherency is desired for masters issuing back-to-back read/write transactions without waiting for write/read responses. Note that this register has an effect only if necessary logic is instantiated through the UMCTL2_RDWR_ORDERED_n parameter. **Programming Mode:** Static
[15]	**Reserved**	Reserved.
[14]	**rd_port_pagematch_en**	**Read Page Match Enable Bit** If set to 1, enables the Page Match feature. If enabled, once a requesting port is granted, the port is continued to be granted if the following immediate commands are to the same memory page (same rank, same bank and same row). See also related PCCFG.pagematch_limit register. **Programming Mode:** Static
[13]	**rd_port_urgent_en**	**AXI Urgent Sideband Signal Enable Bit** If set to 1, enables the AXI urgent sideband signal (arurgent). When enabled and arurgent is asserted by the master, that port becomes the highest priority and co_gs_go2critical_lpr/co_gs_go2critical_hpr signal to DDRC is asserted if enabled in PCCFG.go2critical_en register. Note that arurgent signal can be asserted anytime and as long as required which is independent of address handshaking (it is not associated with any particular command). **Programming Mode:** Static
[12]	**rd_port_aging_en**	**Read Channel of Port Aging Function Enable Bit** If set to 1, enables aging function for the read channel of the port. **Programming Mode:** Static

[11:10]	**Reserved**	Reserved.
[9:0]	**rd_port_priority**	**Read Channel of Port Priority** Determines the initial load value of read aging counters. These counters are parallel loaded after reset, or after each grant to the corresponding port. The aging counters down-count every clock cycle where the port is requesting but not granted. The higher significant 5-bits of the read aging counter sets the priority of the read channel of a given port. Port's priority increases as the higher significant 5-bits of the counter starts to decrease. When the aging counter becomes 0, the corresponding port channel has the highest priority level (timeout condition - Priority0). For multi-port configurations, the aging counters cannot be used to set port priorities when external dynamic priority inputs (arqos) are enabled (timeout is still applicable). For single port configurations, the aging counters are only used when they timeout (become 0) to force read-write direction switching. In this case, external dynamic priority input, arqos (for reads only) can still be used to set the DDRC read priority (2 priority levels: low priority read - LPR, high priority read - HPR) on a command by command basis. **Note:** The two LSBs of this register field are tied internally to 2'b00. **Programming Mode:** Static

UMCTL2 Port 0 Configuration Write Register (PCFGW_0)

Register	Offset	R/W	Description	Reset Value
PCFGW_0	UMCTL2_BA+0x408	R/W	Port 0 Configuration Write Register	0x0000_4000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**wr_port_page match_en**	**wr_port_urgen t_en**	**wr_port_aging _en**	**Reserved**		**wr_port_priority**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
wr_port_priority							

Bits	Description	
[31:15]	**Reserved**	Reserved.
[14]	**wr_port_pagematch_en**	**Write Page Match Enable Bit** If set to 1, enables the Page Match feature. If enabled, once a requesting port is granted, the port is continued to be granted if the following immediate commands are to the same memory page (same rank, same bank and same row). See also related PCCFG.pagematch_limit register. **Programming Mode:** Static
[13]	**wr_port_urgent_en**	**AXI Urgent Sideband Signal Enable Bit** If set to 1, enables the AXI urgent sideband signal (awurgent). When enabled and awurgent is asserted by the master, that port becomes the highest priority and co_gs_go2critical_wr signal to DDRC is asserted if enabled in PCCFG.go2critical_en register. Note that awurgent signal can be asserted anytime and as long as required which is independent of address handshaking (it is not associated with any particular command). **Programming Mode:** Static
[12]	**wr_port_aging_en**	**Write Channel of Port Aging Function Enable Bit** If set to 1, enables aging function for the write channel of the port. **Programming Mode:** Static
[11:10]	**Reserved**	Reserved.
[9:0]	**wr_port_priority**	**Write Channel of Port Priority** Determines the initial load value of write aging counters. These counters are parallel loaded after reset, or after each grant to the corresponding port. The aging counters down-count every clock cycle where the port is requesting but not granted. The higher significant 5-bits of the write aging counter sets the initial priority of the write channel of a given port. Port's priority increases as the higher significant 5-bits of the counter starts to decrease. When the aging counter becomes 0, the corresponding port channel has the highest priority level.

		For multi-port configurations, the aging counters cannot be used to set port priorities when external dynamic priority inputs (awqos) are enabled (timeout is still applicable). For single port configurations, the aging counters are only used when they timeout (become 0) to force read-write direction switching. **Note:** The two LSBs of this register field are tied internally to 2'b00. **Programming Mode:** Static

UMCTL2 Port 0 Control Register (PCTRL_0)

Register	Offset	R/W	Description	Reset Value
PCTRL_0	UMCTL2_BA+0x490	R/W	Port 0 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**port_en**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**port_en**	**AXI Port 0 Enable Bit** **Programming Mode:** Dynamic

UMCTL2 Port 0 Read QoS Configuration Register 0 (PCFGQOS0_0)

Register	Offset	R/W	Description	Reset Value
PCFGQOS0_0	UMCTL2_BA+0x494	R/W	Port 0 Read QoS Configuration Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		**rqos_map_region1**		**Reserved**		**rqos_map_region0**	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**rqos_map_level1**			

Bits	Description	
[31:22]	**Reserved**	Reserved.
[21:20]	**rqos_map_region1**	**Traffic Class of Region 1** This bit field indicates the traffic class of region 1. Valid values are: 0 = LPR. 1 = VPR. 2 = HPR. For dual address queue configurations, region1 maps to the blue address queue. In this case, valid values are 0 = LPR. 1 = VPR only. When VPR support is disabled (UMCTL2_VPR_EN = 0) and traffic class of region 1 is set to 1 (VPR), VPR traffic is aliased to LPR traffic. **Programming Mode:** Quasi-dynamic Group 3
[19:18]	**Reserved**	Reserved.
[17:16]	**rqos_map_region0**	**Traffic Class of Region 0** This bit field indicates the traffic class of region 0. Valid values are: 0 = LPR. 1 = VPR. 2 = HPR. For dual address queue configurations, region 0 maps to the blue address queue. In this case, valid values are: 0: LPR and 1: VPR only. When VPR support is disabled (UMCTL2_VPR_EN = 0) and traffic class of region0 is set to 1 (VPR), VPR traffic is aliased to LPR traffic. **Programming Mode:** Quasi-dynamic Group 3
[15:4]	**Reserved**	Reserved.

[3:0]	**rqos_map_level1**	**End of Region0 Mapping** Separation level1 indicating the end of region0 mapping; start of region0 is 0. Possible values for level1 are 0 to 13 (for dual RAQ) or 0 to 14 (for single RAQ) which corresponds to arqos. Note that for PA, arqos values are used directly as port priorities, where the higher the value corresponds to higher port priority. All of the map_level* registers must be set to distinct values. **Programming Mode:** Quasi-dynamic Group 3

UMCTL2 Port 1 Configuration Read Register (PCFGR_1)

Register	Offset	R/W	Description	Reset Value
PCFGR_1	UMCTL2_BA+0x4b4	R/W	Port 1 Configuration Read Register	0x0000_4000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							rdwr_ordered_en
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	rd_port_pagematch_en	rd_port_urgent_en	rd_port_aging_en	Reserved		rd_port_priority	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
rd_port_priority							

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**rdwr_ordered_en**	**Read/Writes Ordered Enable Bit** If set to 1, preserves the ordering between read transaction and write transaction issued to the same address, on a given port. In other words, the controller ensures that all same address read and write commands from the application port interface are transported to the DFI interface in the order of acceptance. This feature is useful in cases where software coherency is desired for masters issuing back-to-back read/write transactions without waiting for write/read responses. **Programming Mode:** Static
[15]	**Reserved**	Reserved.
[14]	**rd_port_pagematch_en**	**Read Page Match Enable Bit** If set to 1, enables the Page Match feature. If enabled, once a requesting port is granted, the port is continued to be granted if the following immediate commands are to the same memory page (same rank, same bank and same row). See also related PCCFG.pagematch_limit register. **Programming Mode:** Static
[13]	**rd_port_urgent_en**	**AXI Urgent Sideband Signal Enable Bit** If set to 1, enables the AXI urgent sideband signal (arurgent). When enabled and arurgent is asserted by the master, that port becomes the highest priority and co_gs_go2critical_lpr/co_gs_go2critical_hpr signal to DDRC is asserted if enabled in PCCFG.go2critical_en register. Note that arurgent signal can be asserted anytime and as long as required which is independent of address handshaking (it is not associated with any particular command). **Programming Mode:** Static
[12]	**rd_port_aging_en**	**Read Channel of Port Aging Function Enable Bit** If set to 1, enables aging function for the read channel of the port. **Programming Mode:** Static
[11:10]	**Reserved**	Reserved.
[9:0]	**rd_port_priority**	**Read Channel of Port Priority**

		Determines the initial load value of read aging counters. These counters are parallel loaded after reset, or after each grant to the corresponding port. The aging counters down-count every clock cycle where the port is requesting but not granted. The higher significant 5-bits of the read aging counter sets the priority of the read channel of a given port. Port's priority increases as the higher significant 5-bits of the counter starts to decrease. When the aging counter becomes 0, the corresponding port channel has the highest priority level (timeout condition - Priority0). For multi-port configurations, the aging counters cannot be used to set port priorities when external dynamic priority inputs (arqos) are enabled (timeout is still applicable). For single port configurations, the aging counters are only used when they timeout (become 0) to force read-write direction switching. In this case, external dynamic priority input, arqos (for reads only) can still be used to set the DDRC read priority (2 priority levels: low priority read - LPR, high priority read - HPR) on a command by command basis. **Note:** The two LSBs of this register field are tied internally to 2'b00. **Programming Mode:** Static

UMCTL2 Port 1 Configuration Write Register (PCFGW_1)

Register	Offset	R/W	Description	Reset Value
PCFGW_1	UMCTL2_BA+0x4b8	R/W	Port 1 Configuration Write Register	0x0000_4000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	wr_port_page match_en	wr_port_urgen t_en	wr_port_aging _en	Reserved		wr_port_priority	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
wr_port_priority							

Bits	Description	
[31:15]	**Reserved**	Reserved.
[14]	**wr_port_pagematch_en**	**Write Page Match Enable Bit** If set to 1, enables the Page Match feature. If enabled, once a requesting port is granted, the port is continued to be granted if the following immediate commands are to the same memory page (same rank, same bank and same row). See also related PCCFG.pagematch_limit register. **Programming Mode:** Static
[13]	**wr_port_urgent_en**	**AXI Urgent Sideband Signal Enable Bit** If set to 1, enables the AXI urgent sideband signal (awurgent). When enabled and awurgent is asserted by the master, that port becomes the highest priority and co_gs_go2critical_wr signal to DDRC is asserted if enabled in PCCFG.go2critical_en register. Note that awurgent signal can be asserted anytime and as long as required which is independent of address handshaking (it is not associated with any particular command). **Programming Mode:** Static
[12]	**wr_port_aging_en**	**Write Channel of Port Aging Function Enable Bit** If set to 1, enables aging function for the write channel of the port. **Programming Mode:** Static
[11:10]	**Reserved**	Reserved.
[9:0]	**wr_port_priority**	**Write Channel of Port Priority** Determines the initial load value of write aging counters. These counters are parallel loaded after reset, or after each grant to the corresponding port. The aging counters down-count every clock cycle where the port is requesting but not granted. The higher significant 5-bits of the write aging counter sets the initial priority of the write channel of a given port. Port's priority increases as the higher significant 5-bits of the counter starts to decrease. When the aging counter becomes 0, the corresponding port channel has the highest priority level.

		For multi-port configurations, the aging counters cannot be used to set port priorities when external dynamic priority inputs (awqos) are enabled (timeout is still applicable). For single port configurations, the aging counters are only used when they timeout (become 0) to force read-write direction switching. **Note:** The two LSBs of this register field are tied internally to 2'b00. **Programming Mode:** Static

UMCTL2 Port 1 Control Register (PCTRL_1)

Register	Offset	R/W	Description	Reset Value
PCTRL_1	UMCTL2_BA+0x540	R/W	Port 1 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**port_en**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**port_en**	**AXI Port 1 Enable Bit** **Programming Mode:** Dynamic

UMCTL2 Port 1 Read QoS Configuration Register 0 (PCFGQOS0_1)

Register	Offset	R/W	Description	Reset Value
PCFGQOS0_1	UMCTL2_BA+0x544	R/W	Port 1 Read QoS Configuration Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved		rqos_map_region1		Reserved		rqos_map_region0	
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				rqos_map_level1			

Bits	Description	
[31:22]	**Reserved**	Reserved.
[21:20]	**rqos_map_region1**	**Traffic Class of Region 1** This bit field indicates the traffic class of region 1. Valid values are: • 0 = LPR. • 1 = VPR. • 2 = HPR. For dual address queue configurations, region1 maps to the blue address queue. In this case, valid values are 0 = LPR. 1 = VPR only. When VPR support is disabled (UMCTL2_VPR_EN = 0) and traffic class of region 1 is set to 1 (VPR), VPR traffic is aliased to LPR traffic. **Programming Mode:** Quasi-dynamic Group 3
[19:18]	**Reserved**	Reserved.
[17:16]	**rqos_map_region0**	**Traffic Class of Region 0** This bit field indicates the traffic class of region 0. Valid values are: 0 = LPR. 1 = VPR. 2 = HPR. For dual address queue configurations, region 0 maps to the blue address queue. In this case, valid values are: 0: LPR and 1: VPR only. When VPR support is disabled (UMCTL2_VPR_EN = 0) and traffic class of region0 is set to 1 (VPR), VPR traffic is aliased to LPR traffic. **Programming Mode:** Quasi-dynamic Group 3

[15:4]	**Reserved**	Reserved.
[3:0]	**rqos_map_level1**	**End of Region0 Mapping** Separation level1 indicating the end of region0 mapping; start of region0 is 0. Possible values for level1 are 0 to 13 (for dual RAQ) or 0 to 14 (for single RAQ) which corresponds to arqos. Note that for PA, arqos values are used directly as port priorities, where the higher the value corresponds to higher port priority. All of the map_level* registers must be set to distinct values. **Programming Mode:** Quasi-dynamic Group 3

UMCTL2 Port 2 Configuration Read Register (PCFGR_2)

Register	Offset	R/W	Description	Reset Value
PCFGR_2	UMCTL2_BA+0x564	R/W	Port 2 Configuration Read Register	0x0000_4000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							rdwr_ordered_en
15	14	13	12	11	10	9	8
Reserved	rd_port_pagematch_en	rd_port_urgent_en	rd_port_aging_en	Reserved		rd_port_priority	
7	6	5	4	3	2	1	0
rd_port_priority							

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**rdwr_ordered_en**	**Read/Writes Ordered Enable Bit** If set to 1, preserves the ordering between read transaction and write transaction issued to the same address, on a given port. In other words, the controller ensures that all same address read and write commands from the application port interface are transported to the DFI interface in the order of acceptance. This feature is useful in cases where software coherency is desired for masters issuing back-to-back read/write transactions without waiting for write/read responses. **Programming Mode:** Static
[15]	**Reserved**	Reserved.
[14]	**rd_port_pagematch_en**	**Read Page Match Enable Bit** If set to 1, enables the Page Match feature. If enabled, once a requesting port is granted, the port is continued to be granted if the following immediate commands are to the same memory page (same rank, same bank and same row). See also related PCCFG.pagematch_limit register. **Programming Mode:** Static
[13]	**rd_port_urgent_en**	**AXI Urgent Sideband Signal Enable Bit** If set to 1, enables the AXI urgent sideband signal (arurgent). When enabled and arurgent is asserted by the master, that port becomes the highest priority and co_gs_go2critical_lpr/co_gs_go2critical_hpr signal to DDRC is asserted if enabled in PCCFG.go2critical_en register. Note that arurgent signal can be asserted anytime and as long as required which is independent of address handshaking (it is not associated with any particular command). **Programming Mode:** Static
[12]	**rd_port_aging_en**	**Read Channel of Port Aging Function Enable Bit** If set to 1, enables aging function for the read channel of the port. **Programming Mode:** Static
[11:10]	**Reserved**	Reserved.
[9:0]	**rd_port_priority**	**Read Channel of Port Priority**

		Determines the initial load value of read aging counters. These counters are parallel loaded after reset, or after each grant to the corresponding port. The aging counters down-count every clock cycle where the port is requesting but not granted. The higher significant 5-bits of the read aging counter sets the priority of the read channel of a given port. Port's priority increases as the higher significant 5-bits of the counter starts to decrease. When the aging counter becomes 0, the corresponding port channel has the highest priority level (timeout condition - Priority0). For multi-port configurations, the aging counters cannot be used to set port priorities when external dynamic priority inputs (arqos) are enabled (timeout is still applicable). For single port configurations, the aging counters are only used when they timeout (become 0) to force read-write direction switching. In this case, external dynamic priority input, arqos (for reads only) can still be used to set the DDRC read priority (2 priority levels: low priority read - LPR, high priority read - HPR) on a command by command basis. **Note:** The two LSBs of this register field are tied internally to 2'b00. **Programming Mode:** Static

UMCTL2 Port 2 Configuration Write Register (PCFGW_2)

Register	Offset	R/W	Description	Reset Value
PCFGW_2	UMCTL2_BA+0x568	R/W	Port 2 Configuration Write Register	0x0000_4000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**wr_port_page match_en**	**wr_port_urgen t_en**	**wr_port_aging _en**	**Reserved**		**wr_port_priority**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
wr_port_priority							

Bits	Description	
[31:15]	**Reserved**	Reserved.
[14]	**wr_port_pagematch_en**	**Write Page Match Enable Bit** If set to 1, enables the Page Match feature. If enabled, once a requesting port is granted, the port is continued to be granted if the following immediate commands are to the same memory page (same rank, same bank and same row). See also related PCCFG.pagematch_limit register. **Programming Mode:** Static
[13]	**wr_port_urgent_en**	**AXI Urgent Sideband Signal Enable Bit** If set to 1, enables the AXI urgent sideband signal (awurgent). When enabled and awurgent is asserted by the master, that port becomes the highest priority and co_gs_go2critical_wr signal to DDRC is asserted if enabled in PCCFG.go2critical_en register. Note that awurgent signal can be asserted anytime and as long as required which is independent of address handshaking (it is not associated with any particular command). **Programming Mode:** Static
[12]	**wr_port_aging_en**	**Write Channel of Port Aging Function Enable Bit** If set to 1, enables aging function for the write channel of the port. **Programming Mode:** Static
[11:10]	**Reserved**	Reserved.
[9:0]	**wr_port_priority**	**Write Channel of Port Priority** Determines the initial load value of write aging counters. These counters are parallel loaded after reset, or after each grant to the corresponding port. The aging counters down-count every clock cycle where the port is requesting but not granted. The higher significant 5-bits of the write aging counter sets the initial priority of the write channel of a given port. Port's priority increases as the higher significant 5-bits of the counter starts to decrease. When the aging counter becomes 0, the corresponding port channel has the highest priority level.

		For multi-port configurations, the aging counters cannot be used to set port priorities when external dynamic priority inputs (awqos) are enabled (timeout is still applicable). For single port configurations, the aging counters are only used when they timeout (become 0) to force read-write direction switching. **Note:** The two LSBs of this register field are tied internally to 2'b00. **Programming Mode:** Static

UMCTL2 Port 2 Control Register (PCTRL_2)

Register	Offset	R/W	Description	Reset Value
PCTRL_2	UMCTL2_BA+0x5f0	R/W	Port 2 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**port_en**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**port_en**	**AXI Port 2 Enable Bit** **Programming Mode:** Dynamic

UMCTL2 Port 2 Read QoS Configuration Register 0 (PCFGQOS0_2)

Register	Offset	R/W	Description	Reset Value
PCFGQOS0_2	UMCTL2_BA+0x5f4	R/W	Port 2 Read QoS Configuration Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved		rqos_map_region1		Reserved		rqos_map_region0	
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				rqos_map_level1			

Bits	Description	
[31:22]	**Reserved**	Reserved.
[21:20]	**rqos_map_region1**	**Traffic Class of Region 1** This bit field indicates the traffic class of region 1. Valid values are: 0 = LPR. 1 = VPR. 2 = HPR. For dual address queue configurations, region1 maps to the blue address queue. In this case, valid values are 0 = LPR. 1 = VPR only. When VPR support is disabled (UMCTL2_VPR_EN = 0) and traffic class of region 1 is set to 1 (VPR), VPR traffic is aliased to LPR traffic. **Programming Mode:** Quasi-dynamic Group 3
[19:18]	**Reserved**	Reserved.
[17:16]	**rqos_map_region0**	**Traffic Class of Region 0** This bit field indicates the traffic class of region 0. Valid values are: 0 = LPR. 1 = VPR. 2 = HPR. For dual address queue configurations, region 0 maps to the blue address queue. In this case, valid values are: 0: LPR and 1: VPR only. When VPR support is disabled (UMCTL2_VPR_EN = 0) and traffic class of region0 is set to 1 (VPR), VPR traffic is aliased to LPR traffic. **Programming Mode:** Quasi-dynamic Group 3
[15:4]	**Reserved**	Reserved.

[3:0]	**rqos_map_level1**	**End of Region0 Mapping** Separation level1 indicating the end of region0 mapping; start of region0 is 0. Possible values for level1 are 0 to 13 (for dual RAQ) or 0 to 14 (for single RAQ) which corresponds to arqos. Note that for PA, arqos values are used directly as port priorities, where the higher the value corresponds to higher port priority. All of the map_level* registers must be set to distinct values. **Programming Mode:** Quasi-dynamic Group 3

UMCTL2 Port 3 Configuration Read Register (PCFGR_3)

Register	Offset	R/W	Description	Reset Value
PCFGR_3	UMCTL2_BA+0x614	R/W	Port 3 Configuration Read Register	0x0000_4000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							rdwr_ordered_en
15	14	13	12	11	10	9	8
Reserved	rd_port_pagematch_en	rd_port_urgent_en	rd_port_aging_en	Reserved		rd_port_priority	
7	6	5	4	3	2	1	0
rd_port_priority							

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**rdwr_ordered_en**	**Read/Writes Ordered Enable Bit** If set to 1, preserves the ordering between read transaction and write transaction issued to the same address, on a given port. In other words, the controller ensures that all same address read and write commands from the application port interface are transported to the DFI interface in the order of acceptance. This feature is useful in cases where software coherency is desired for masters issuing back-to-back read/write transactions without waiting for write/read responses. **Programming Mode:** Static
[15]	**Reserved**	Reserved.
[14]	**rd_port_pagematch_en**	**Read Page Match Enable Bit** If set to 1, enables the Page Match feature. If enabled, once a requesting port is granted, the port is continued to be granted if the following immediate commands are to the same memory page (same rank, same bank and same row). See also related PCCFG.pagematch_limit register. **Programming Mode:** Static
[13]	**rd_port_urgent_en**	**AXI Urgent Sideband Signal Enable Bit** If set to 1, enables the AXI urgent sideband signal (arurgent). When enabled and arurgent is asserted by the master, that port becomes the highest priority and co_gs_go2critical_lpr/co_gs_go2critical_hpr signal to DDRC is asserted if enabled in PCCFG.go2critical_en register. Note that arurgent signal can be asserted anytime and as long as required which is independent of address handshaking (it is not associated with any particular command). **Programming Mode:** Static
[12]	**rd_port_aging_en**	**Read Channel of Port Aging Function Enable Bit** If set to 1, enables aging function for the read channel of the port. **Programming Mode:** Static
[11:10]	**Reserved**	Reserved.
[9:0]	**rd_port_priority**	**Read Channel of Port Priority**

		Determines the initial load value of read aging counters. These counters are parallel loaded after reset, or after each grant to the corresponding port. The aging counters down-count every clock cycle where the port is requesting but not granted. The higher significant 5-bits of the read aging counter sets the priority of the read channel of a given port. Port's priority increases as the higher significant 5-bits of the counter starts to decrease. When the aging counter becomes 0, the corresponding port channel has the highest priority level (timeout condition - Priority0). For multi-port configurations, the aging counters cannot be used to set port priorities when external dynamic priority inputs (arqos) are enabled (timeout is still applicable). For single port configurations, the aging counters are only used when they timeout (become 0) to force read-write direction switching. In this case, external dynamic priority input, arqos (for reads only) can still be used to set the DDRC read priority (2 priority levels: low priority read - LPR, high priority read - HPR) on a command by command basis. **Note:** The two LSBs of this register field are tied internally to 2'b00. **Programming Mode:** Static

UMCTL2 Port 3 Configuration Write Register (PCFGW_3)

Register	Offset	R/W	Description	Reset Value
PCFGW_3	UMCTL2_BA+0x618	R/W	Port 3 Configuration Write Register	0x0000_4000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**wr_port_page match_en**	**wr_port_urgen t_en**	**wr_port_aging _en**	**Reserved**		**wr_port_priority**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
wr_port_priority							

Bits	Description	
[31:15]	**Reserved**	Reserved.
[14]	**wr_port_pagematch_en**	**Write Page Match Enable Bit** If set to 1, enables the Page Match feature. If enabled, once a requesting port is granted, the port is continued to be granted if the following immediate commands are to the same memory page (same rank, same bank and same row). See also related PCCFG.pagematch_limit register. **Programming Mode:** Static
[13]	**wr_port_urgent_en**	**AXI Urgent Sideband Signal Enable Bit** If set to 1, enables the AXI urgent sideband signal (awurgent). When enabled and awurgent is asserted by the master, that port becomes the highest priority and co_gs_go2critical_wr signal to DDRC is asserted if enabled in PCCFG.go2critical_en register. Note that awurgent signal can be asserted anytime and as long as required which is independent of address handshaking (it is not associated with any particular command). **Programming Mode:** Static
[12]	**wr_port_aging_en**	**Write Channel of Port Aging Function Enable Bit** If set to 1, enables aging function for the write channel of the port. **Programming Mode:** Static
[11:10]	**Reserved**	Reserved.
[9:0]	**wr_port_priority**	**Write Channel of Port Priority** Determines the initial load value of write aging counters. These counters are parallel loaded after reset, or after each grant to the corresponding port. The aging counters down-count every clock cycle where the port is requesting but not granted. The higher significant 5-bits of the write aging counter sets the initial priority of the write channel of a given port. Port's priority increases as the higher significant 5-bits of the counter starts to decrease. When the aging counter becomes 0, the corresponding port channel has the highest priority level.

		For multi-port configurations, the aging counters cannot be used to set port priorities when external dynamic priority inputs (awqos) are enabled (timeout is still applicable). For single port configurations, the aging counters are only used when they timeout (become 0) to force read-write direction switching. **Note:** The two LSBs of this register field are tied internally to 2'b00. **Programming Mode:** Static

UMCTL2 Port 3 Control Register (PCTRL_3)

Register	Offset	R/W	Description	Reset Value
PCTRL_3	UMCTL2_BA+0x6a0	R/W	Port 3 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**port_en**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**port_en**	**AXI Port 3 Enable Bit** **Programming Mode:** Dynamic

UMCTL2 Port 3 Read QoS Configuration Register 0 (PCFGQOS0_3)

Register	Offset	R/W	Description	Reset Value
PCFGQOS0_3	UMCTL2_BA+0x6a4	R/W	Port 3 Read QoS Configuration Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		rqos_map_region1		Reserved		rqos_map_region0	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				rqos_map_level1			

Bits	Description	
[31:22]	**Reserved**	Reserved.
[21:20]	**rqos_map_region1**	**Traffic Class of Region 1** This bit field indicates the traffic class of region 1. Valid values are: 0 = LPR. 1 = VPR. 2 = HPR. For dual address queue configurations, region1 maps to the blue address queue. In this case, valid values are 0 = LPR. 1 = VPR only. When VPR support is disabled (UMCTL2_VPR_EN = 0) and traffic class of region 1 is set to 1 (VPR), VPR traffic is aliased to LPR traffic. **Programming Mode:** Quasi-dynamic Group 3
[19:18]	**Reserved**	Reserved.
[17:16]	**rqos_map_region0**	**Traffic Class of Region 0** This bit field indicates the traffic class of region 0. Valid values are: 0 = LPR. 1 = VPR. 2 = HPR. For dual address queue configurations, region 0 maps to the blue address queue. In this case, valid values are: 0: LPR and 1: VPR only. When VPR support is disabled (UMCTL2_VPR_EN = 0) and traffic class of region0 is set to 1 (VPR), VPR traffic is aliased to LPR traffic. **Programming Mode:** Quasi-dynamic Group 3
[15:4]	**Reserved**	Reserved.

[3:0]	**rqos_map_level1**	**End of Region0 Mapping** Separation level1 indicating the end of region0 mapping; start of region0 is 0. Possible values for level1 are 0 to 13 (for dual RAQ) or 0 to 14 (for single RAQ) which corresponds to arqos. Note that for PA, arqos values are used directly as port priorities, where the higher the value corresponds to higher port priority. All of the map_level* registers must be set to distinct values. **Programming Mode:** Quasi-dynamic Group 3

UMCTL2 Port 4 Configuration Read Register (PCFGR_4)

Register	Offset	R/W	Description	Reset Value
PCFGR_4	UMCTL2_BA+0x6c4	R/W	Port 4 Configuration Read Register	0x0000_4000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							**rdwr_ordered_en**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**rd_port_pagematch_en**	**rd_port_urgent_en**	**rd_port_aging_en**	**Reserved**		**rd_port_priority**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
rd_port_priority							

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**rdwr_ordered_en**	**Read/Writes Ordered Enable Bit** If set to 1, preserves the ordering between read transaction and write transaction issued to the same address, on a given port. In other words, the controller ensures that all same address read and write commands from the application port interface are transported to the DFI interface in the order of acceptance. This feature is useful in cases where software coherency is desired for masters issuing back-to-back read/write transactions without waiting for write/read responses. **Programming Mode:** Static
[15]	**Reserved**	Reserved.
[14]	**rd_port_pagematch_en**	**Read Page Match Enable Bit** If set to 1, enables the Page Match feature. If enabled, once a requesting port is granted, the port is continued to be granted if the following immediate commands are to the same memory page (same rank, same bank and same row). See also related PCCFG.pagematch_limit register. **Programming Mode:** Static
[13]	**rd_port_urgent_en**	**AXI Urgent Sideband Signal Enable Bit** If set to 1, enables the AXI urgent sideband signal (arurgent). When enabled and arurgent is asserted by the master, that port becomes the highest priority and co_gs_go2critical_lpr/co_gs_go2critical_hpr signal to DDRC is asserted if enabled in PCCFG.go2critical_en register. Note that arurgent signal can be asserted anytime and as long as required which is independent of address handshaking (it is not associated with any particular command). **Programming Mode:** Static
[12]	**rd_port_aging_en**	**Read Channel of Port Aging Function Enable Bit** If set to 1, enables aging function for the read channel of the port. **Programming Mode:** Static
[11:10]	**Reserved**	Reserved.
[9:0]	**rd_port_priority**	**Read Channel of Port Priority**

		Determines the initial load value of read aging counters. These counters are parallel loaded after reset, or after each grant to the corresponding port. The aging counters down-count every clock cycle where the port is requesting but not granted. The higher significant 5-bits of the read aging counter sets the priority of the read channel of a given port. Port's priority increases as the higher significant 5-bits of the counter starts to decrease. When the aging counter becomes 0, the corresponding port channel has the highest priority level (timeout condition - Priority0). For multi-port configurations, the aging counters cannot be used to set port priorities when external dynamic priority inputs (arqos) are enabled (timeout is still applicable). For single port configurations, the aging counters are only used when they timeout (become 0) to force read-write direction switching. In this case, external dynamic priority input, arqos (for reads only) can still be used to set the DDRC read priority (2 priority levels: low priority read - LPR, high priority read - HPR) on a command by command basis. **Note:** The two LSBs of this register field are tied internally to 2'b00. **Programming Mode:** Static

UMCTL2 Port 4 Configuration Write Register (PCFGW_4)

Register	Offset	R/W	Description	Reset Value
PCFGW_4	UMCTL2_BA+0x6c8	R/W	Port 4 Configuration Write Register	0x0000_4000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	wr_port_page match_en	wr_port_urgen t_en	wr_port_aging _en	Reserved		wr_port_priority	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
wr_port_priority							

Bits	Description	
[31:15]	**Reserved**	Reserved.
[14]	**wr_port_pagematch_en**	**Write Page Match Enable Bit** If set to 1, enables the Page Match feature. If enabled, once a requesting port is granted, the port is continued to be granted if the following immediate commands are to the same memory page (same rank, same bank and same row). See also related PCCFG.pagematch_limit register. **Programming Mode:** Static
[13]	**wr_port_urgent_en**	**AXI Urgent Sideband Signal Enable Bit** If set to 1, enables the AXI urgent sideband signal (awurgent). When enabled and awurgent is asserted by the master, that port becomes the highest priority and co_gs_go2critical_wr signal to DDRC is asserted if enabled in PCCFG.go2critical_en register. Note that awurgent signal can be asserted anytime and as long as required which is independent of address handshaking (it is not associated with any particular command). **Programming Mode:** Static
[12]	**wr_port_aging_en**	**Write Channel of Port Aging Function Enable Bit** If set to 1, enables aging function for the write channel of the port. **Programming Mode:** Static
[11:10]	**Reserved**	Reserved.
[9:0]	**wr_port_priority**	**Write Channel of Port Priority** Determines the initial load value of write aging counters. These counters are parallel loaded after reset, or after each grant to the corresponding port. The aging counters down-count every clock cycle where the port is requesting but not granted. The higher significant 5-bits of the write aging counter sets the initial priority of the write channel of a given port. Port's priority increases as the higher significant 5-bits of the counter starts to decrease. When the aging counter becomes 0, the corresponding port channel has the highest priority level.

		For multi-port configurations, the aging counters cannot be used to set port priorities when external dynamic priority inputs (awqos) are enabled (timeout is still applicable). For single port configurations, the aging counters are only used when they timeout (become 0) to force read-write direction switching. **Note:** The two LSBs of this register field are tied internally to 2'b00. **Programming Mode:** Static

UMCTL2 Port 4 Control Register (PCTRL_4)

Register	Offset	R/W	Description	Reset Value
PCTRL_4	UMCTL2_BA+0x750	R/W	Port 4 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**port_en**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**port_en**	**AXI Port 4 Enable Bit** **Programming Mode:** Dynamic

UMCTL2 Port 4 Read QoS Configuration Register 0 (PCFGQOS0_4)

Register	Offset	R/W	Description	Reset Value
PCFGQOS0_4	UMCTL2_BA+0x754	R/W	Port 4 Read QoS Configuration Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved		rqos_map_region1		Reserved		rqos_map_region0	
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				rqos_map_level1			

Bits	Description	
[31:22]	**Reserved**	Reserved.
[21:20]	**rqos_map_region1**	**Traffic Class of Region 1** This bit field indicates the traffic class of region 1. Valid values are: 0 = LPR. 1 = VPR. 2 = HPR. For dual address queue configurations, region1 maps to the blue address queue. In this case, valid values are 0 = LPR. 1 = VPR only. When VPR support is disabled (UMCTL2_VPR_EN = 0) and traffic class of region 1 is set to 1 (VPR), VPR traffic is aliased to LPR traffic. **Programming Mode:** Quasi-dynamic Group 3
[19:18]	**Reserved**	Reserved.
[17:16]	**rqos_map_region0**	**Traffic Class of Region 0** This bit field indicates the traffic class of region 0. Valid values are: 0 = LPR. 1 = VPR. 2 = HPR. For dual address queue configurations, region 0 maps to the blue address queue. In this case, valid values are: 0: LPR and 1: VPR only. When VPR support is disabled (UMCTL2_VPR_EN = 0) and traffic class of region0 is set to 1 (VPR), VPR traffic is aliased to LPR traffic. **Programming Mode:** Quasi-dynamic Group 3
[15:4]	**Reserved**	Reserved.

[3:0]	**rqos_map_level1**	**End of Region0 Mapping** Separation level1 indicating the end of region0 mapping; start of region0 is 0. Possible values for level1 are 0 to 13 (for dual RAQ) or 0 to 14 (for single RAQ) which corresponds to arqos. Note that for PA, arqos values are used directly as port priorities, where the higher the value corresponds to higher port priority. All of the map_level* registers must be set to distinct values. **Programming Mode:** Quasi-dynamic Group 3

UMCTL2 Port 5 Configuration Read Register (PCFGR_5)

Register	Offset	R/W	Description	Reset Value
PCFGR_5	UMCTL2_BA+0x774	R/W	Port 5 Configuration Read Register	0x0000_4000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							**rdwr_ordered_en**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**rd_port_pagematch_en**	**rd_port_urgent_en**	**rd_port_aging_en**	**Reserved**		**rd_port_priority**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
rd_port_priority							

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**rdwr_ordered_en**	**Read/Writes Ordered Enable Bit** If set to 1, preserves the ordering between read transaction and write transaction issued to the same address, on a given port. In other words, the controller ensures that all same address read and write commands from the application port interface are transported to the DFI interface in the order of acceptance. This feature is useful in cases where software coherency is desired for masters issuing back-to-back read/write transactions without waiting for write/read responses. **Programming Mode:** Static
[15]	**Reserved**	Reserved.
[14]	**rd_port_pagematch_en**	**Read Page Match Enable Bit** If set to 1, enables the Page Match feature. If enabled, once a requesting port is granted, the port is continued to be granted if the following immediate commands are to the same memory page (same rank, same bank and same row). See also related PCCFG.pagematch_limit register. **Programming Mode:** Static
[13]	**rd_port_urgent_en**	**AXI Urgent Sideband Signal Enable Bit** If set to 1, enables the AXI urgent sideband signal (arurgent). When enabled and arurgent is asserted by the master, that port becomes the highest priority and co_gs_go2critical_lpr/co_gs_go2critical_hpr signal to DDRC is asserted if enabled in PCCFG.go2critical_en register. Note that arurgent signal can be asserted anytime and as long as required which is independent of address handshaking (it is not associated with any particular command). **Programming Mode:** Static
[12]	**rd_port_aging_en**	**Read Channel of Port Aging Function Enable Bit** If set to 1, enables aging function for the read channel of the port. **Programming Mode:** Static
[11:10]	**Reserved**	Reserved.
[9:0]	**rd_port_priority**	**Read Channel of Port Priority**

		Determines the initial load value of read aging counters. These counters are parallel loaded after reset, or after each grant to the corresponding port. The aging counters down-count every clock cycle where the port is requesting but not granted. The higher significant 5-bits of the read aging counter sets the priority of the read channel of a given port. Port's priority increases as the higher significant 5-bits of the counter starts to decrease. When the aging counter becomes 0, the corresponding port channel has the highest priority level (timeout condition - Priority0). For multi-port configurations, the aging counters cannot be used to set port priorities when external dynamic priority inputs (arqos) are enabled (timeout is still applicable). For single port configurations, the aging counters are only used when they timeout (become 0) to force read-write direction switching. In this case, external dynamic priority input, arqos (for reads only) can still be used to set the DDRC read priority (2 priority levels: low priority read - LPR, high priority read - HPR) on a command by command basis. **Note:** The two LSBs of this register field are tied internally to 2'b00. **Programming Mode:** Static

UMCTL2 Port 5 Configuration Write Register (PCFGW_5)

Register	Offset	R/W	Description	Reset Value
PCFGW_5	UMCTL2_BA+0x778	R/W	Port 5 Configuration Write Register	0x0000_4000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**wr_port_page match_en**	**wr_port_urgen t_en**	**wr_port_aging _en**	**Reserved**		**wr_port_priority**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
wr_port_priority							

Bits	Description	
[31:15]	**Reserved**	Reserved.
[14]	**wr_port_pagematch_en**	**Write Page Match Enable Bit** If set to 1, enables the Page Match feature. If enabled, once a requesting port is granted, the port is continued to be granted if the following immediate commands are to the same memory page (same rank, same bank and same row). See also related PCCFG.pagematch_limit register. **Programming Mode:** Static
[13]	**wr_port_urgent_en**	**AXI Urgent Sideband Signal Enable Bit** If set to 1, enables the AXI urgent sideband signal (awurgent). When enabled and awurgent is asserted by the master, that port becomes the highest priority and co_gs_go2critical_wr signal to DDRC is asserted if enabled in PCCFG.go2critical_en register. Note that awurgent signal can be asserted anytime and as long as required which is independent of address handshaking (it is not associated with any particular command). **Programming Mode:** Static
[12]	**wr_port_aging_en**	**Write Channel of Port Aging Function Enable Bit** If set to 1, enables aging function for the write channel of the port. **Programming Mode:** Static
[11:10]	**Reserved**	Reserved.
[9:0]	**wr_port_priority**	**Write Channel of Port Priority** Determines the initial load value of write aging counters. These counters are parallel loaded after reset, or after each grant to the corresponding port. The aging counters down-count every clock cycle where the port is requesting but not granted. The higher significant 5-bits of the write aging counter sets the initial priority of the write channel of a given port. Port's priority increases as the higher significant 5-bits of the counter starts to decrease. When the aging counter becomes 0, the corresponding port channel has the highest priority level.

		For multi-port configurations, the aging counters cannot be used to set port priorities when external dynamic priority inputs (awqos) are enabled (timeout is still applicable). For single port configurations, the aging counters are only used when they timeout (become 0) to force read-write direction switching. **Note:** The two LSBs of this register field are tied internally to 2'b00. **Programming Mode:** Static

UMCTL2 Port 5 Control Register (PCTRL_5)

Register	Offset	R/W	Description	Reset Value
PCTRL_5	UMCTL2_BA+0x800	R/W	Port 5 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							port_en

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**port_en**	**AXI Port 5 Enable Bit** **Programming Mode:** Dynamic

UMCTL2 Port 5 Read QoS Configuration Register 0 (PCFGQOS0_5)

Register	Offset	R/W	Description	Reset Value
PCFGQOS0_5	UMCTL2_BA+0x804	R/W	Port 5 Read QoS Configuration Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved		rqos_map_region1		Reserved		rqos_map_region0	
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				rqos_map_level1			

Bits	Description	
[31:22]	**Reserved**	Reserved.
[21:20]	**rqos_map_region1**	**Traffic Class of Region 1** This bit field indicates the traffic class of region 1. Valid values are: 0 = LPR. 1 = VPR. 2 = HPR. For dual address queue configurations, region1 maps to the blue address queue. In this case, valid values are 0 = LPR. 1 = VPR only. When VPR support is disabled (UMCTL2_VPR_EN = 0) and traffic class of region 1 is set to 1 (VPR), VPR traffic is aliased to LPR traffic. **Programming Mode:** Quasi-dynamic Group 3
[19:18]	**Reserved**	Reserved.
[17:16]	**rqos_map_region0**	**Traffic Class of Region 0** This bit field indicates the traffic class of region 0. Valid values are: 0 = LPR. 1 = VPR. 2 = HPR. For dual address queue configurations, region 0 maps to the blue address queue. In this case, valid values are: 0: LPR and 1: VPR only. When VPR support is disabled (UMCTL2_VPR_EN = 0) and traffic class of region0 is set to 1 (VPR), VPR traffic is aliased to LPR traffic. **Programming Mode:** Quasi-dynamic Group 3
[15:4]	**Reserved**	Reserved.

[3:0]	**rqos_map_level1**	**End of Region0 Mapping** Separation level1 indicating the end of region0 mapping; start of region0 is 0. Possible values for level1 are 0 to 13 (for dual RAQ) or 0 to 14 (for single RAQ) which corresponds to arqos. Note that for PA, arqos values are used directly as port priorities, where the higher the value corresponds to higher port priority. All of the map_level* registers must be set to distinct values. **Programming Mode:** Quasi-dynamic Group 3

UMCTL2 Port 6 Configuration Read Register (PCFGR_6)

Register	Offset	R/W	Description	Reset Value
PCFGR_6	UMCTL2_BA+0x824	R/W	Port 6 Configuration Read Register	0x0000_4000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							**rdwr_ordered_en**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**rd_port_pagematch_en**	**rd_port_urgent_en**	**rd_port_aging_en**	**Reserved**		**rd_port_priority**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
rd_port_priority							

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**rdwr_ordered_en**	**Read/Writes Ordered Enable Bit** If set to 1, preserves the ordering between read transaction and write transaction issued to the same address, on a given port. In other words, the controller ensures that all same address read and write commands from the application port interface are transported to the DFI interface in the order of acceptance. This feature is useful in cases where software coherency is desired for masters issuing back-to-back read/write transactions without waiting for write/read responses. **Programming Mode:** Static
[15]	**Reserved**	Reserved.
[14]	**rd_port_pagematch_en**	**Read Page Match Enable Bit** If set to 1, enables the Page Match feature. If enabled, once a requesting port is granted, the port is continued to be granted if the following immediate commands are to the same memory page (same rank, same bank and same row). See also related PCCFG.pagematch_limit register. **Programming Mode:** Static
[13]	**rd_port_urgent_en**	**AXI Urgent Sideband Signal Enable Bit** If set to 1, enables the AXI urgent sideband signal (arurgent). When enabled and arurgent is asserted by the master, that port becomes the highest priority and co_gs_go2critical_lpr/co_gs_go2critical_hpr signal to DDRC is asserted if enabled in PCCFG.go2critical_en register. Note that arurgent signal can be asserted anytime and as long as required which is independent of address handshaking (it is not associated with any particular command). **Programming Mode:** Static
[12]	**rd_port_aging_en**	**Read Channel of Port Aging Function Enable Bit** If set to 1, enables aging function for the read channel of the port. **Programming Mode:** Static
[11:10]	**Reserved**	Reserved.
[9:0]	**rd_port_priority**	**Read Channel of Port Priority**

		Determines the initial load value of read aging counters. These counters are parallel loaded after reset, or after each grant to the corresponding port. The aging counters down-count every clock cycle where the port is requesting but not granted. The higher significant 5-bits of the read aging counter sets the priority of the read channel of a given port. Port's priority increases as the higher significant 5-bits of the counter starts to decrease. When the aging counter becomes 0, the corresponding port channel has the highest priority level (timeout condition - Priority0). For multi-port configurations, the aging counters cannot be used to set port priorities when external dynamic priority inputs (arqos) are enabled (timeout is still applicable). For single port configurations, the aging counters are only used when they timeout (become 0) to force read-write direction switching. In this case, external dynamic priority input, arqos (for reads only) can still be used to set the DDRC read priority (2 priority levels: low priority read - LPR, high priority read - HPR) on a command by command basis. **Note:** The two LSBs of this register field are tied internally to 2'b00. **Programming Mode:** Static

UMCTL2 Port 6 Configuration Write Register (PCFGW_6)

Register	Offset	R/W	Description	Reset Value
PCFGW_6	UMCTL2_BA+0x828	R/W	Port 6 Configuration Write Register	0x0000_4000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**wr_port_page match_en**	**wr_port_urgen t_en**	**wr_port_aging _en**	**Reserved**		**wr_port_priority**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
wr_port_priority							

Bits	Description	
[31:15]	**Reserved**	Reserved.
[14]	**wr_port_pagematch_en**	**Write Page Match Enable Bit** If set to 1, enables the Page Match feature. If enabled, once a requesting port is granted, the port is continued to be granted if the following immediate commands are to the same memory page (same rank, same bank and same row). See also related PCCFG.pagematch_limit register. **Programming Mode:** Static
[13]	**wr_port_urgent_en**	**AXI Urgent Sideband Signal Enable Bit** If set to 1, enables the AXI urgent sideband signal (awurgent). When enabled and awurgent is asserted by the master, that port becomes the highest priority and co_gs_go2critical_wr signal to DDRC is asserted if enabled in PCCFG.go2critical_en register. Note that awurgent signal can be asserted anytime and as long as required which is independent of address handshaking (it is not associated with any particular command). **Programming Mode:** Static
[12]	**wr_port_aging_en**	**Write Channel of Port Aging Function Enable Bit** If set to 1, enables aging function for the write channel of the port. **Programming Mode:** Static
[11:10]	**Reserved**	Reserved.
[9:0]	**wr_port_priority**	**Write Channel of Port Priority** Determines the initial load value of write aging counters. These counters are parallel loaded after reset, or after each grant to the corresponding port. The aging counters down-count every clock cycle where the port is requesting but not granted. The higher significant 5-bits of the write aging counter sets the initial priority of the write channel of a given port. Port's priority increases as the higher significant 5-bits of the counter starts to decrease. When the aging counter becomes 0, the corresponding port channel has the highest priority level.

		For multi-port configurations, the aging counters cannot be used to set port priorities when external dynamic priority inputs (awqos) are enabled (timeout is still applicable). For single port configurations, the aging counters are only used when they timeout (become 0) to force read-write direction switching. **Note:** The two LSBs of this register field are tied internally to 2'b00. **Programming Mode:** Static

UMCTL2 Port 6 Control Register (PCTRL_6)

Register	Offset	R/W	Description	Reset Value
PCTRL_6	UMCTL2_BA+0x8b0	R/W	Port 6 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**port_en**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**port_en**	**AXI Port 6 Enable Bit** **Programming Mode:** Dynamic

UMCTL2 Port 6 Read QoS Configuration Register 0 (PCFGQOS0_6)

Register	Offset	R/W	Description	Reset Value
PCFGQOS0_6	UMCTL2_BA+0x8b4	R/W	Port 6 Read QoS Configuration Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		rqos_map_region1		Reserved		rqos_map_region0	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				rqos_map_level1			

Bits	Description	
[31:22]	**Reserved**	Reserved.
[21:20]	**rqos_map_region1**	**Traffic Class of Region 1** This bit field indicates the traffic class of region 1. Valid values are: 0 = LPR. 1 = VPR. 2 = HPR. For dual address queue configurations, region1 maps to the blue address queue. In this case, valid values are 0 = LPR. 1 = VPR only. When VPR support is disabled (UMCTL2_VPR_EN = 0) and traffic class of region 1 is set to 1 (VPR), VPR traffic is aliased to LPR traffic. **Programming Mode:** Quasi-dynamic Group 3
[19:18]	**Reserved**	Reserved.
[17:16]	**rqos_map_region0**	**Traffic Class of Region 0** This bit field indicates the traffic class of region 0. Valid values are: 0 = LPR. 1 = VPR. 2 = HPR. For dual address queue configurations, region 0 maps to the blue address queue. In this case, valid values are: 0: LPR and 1: VPR only. When VPR support is disabled (UMCTL2_VPR_EN = 0) and traffic class of region0 is set to 1 (VPR), VPR traffic is aliased to LPR traffic. **Programming Mode:** Quasi-dynamic Group 3
[15:4]	**Reserved**	Reserved.

[3:0]	**rqos_map_level1**	**End of Region0 Mapping** Separation level1 indicating the end of region0 mapping; start of region0 is 0. Possible values for level1 are 0 to 13 (for dual RAQ) or 0 to 14 (for single RAQ) which corresponds to arqos. Note that for PA, arqos values are used directly as port priorities, where the higher the value corresponds to higher port priority. All of the map_level* registers must be set to distinct values. **Programming Mode:** Quasi-dynamic Group 3

UMCTL2 SAR Base Address Register n (SARBASE0)

Register	Offset	R/W	Description	Reset Value
SARBASE0	UMCTL2_BA+0xf04	R/W	SAR Base Address Register n	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						base_addr	

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1:0]	**base_addr**	**Base Address for Address Region n** Base address for address region n specified as awaddr[UMCTL2_A_ADDRW-1:x] and araddr[UMCTL2_A_ADDRW-1:x], where x is determined by the minimum block size parameter UMCTL2_SARMINSIZE: (x=log2(block size)). **Programming Mode:** Static

UMCTL2 SAR Size Register n (SARSIZE0)

Register	Offset	R/W	Description	Reset Value
SARSIZE0	UMCTL2_BA+0xf08	R/W	SAR Size Register n	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
nblocks							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**nblocks**	**Number of Blocks for Address Region n** This register determines the total size of the region in multiples of minimum block size as specified by the hardware parameter UMCTL2_SARMINSIZE. The register value is encoded as number of blocks = nblocks + 1. For example, if register is programmed to 0, region can have 1 block. **Programming Mode:** Static

UMCTL2 UMCTL2 Version Number Register (UMCTL2_VER_NUMBER)

Register	Offset	R/W	Description	Reset Value
UMCTL2_VER_NUMBER	UMCTL2_BA+0xff0	R	UMCTL2 Version Number Register	0x3338_302a

31	30	29	28	27	26	25	24
ver_number							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ver_number							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ver_number							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ver_number							

Bits	Description	
[31:0]	**ver_number**	**Device Version Number Value** This is in ASCII format, with each byte corresponding to a character of the version number **Programming Mode:** Static

UMCTL2 UMCTL2 Version Type Register (UMCTL2_VER_TYPE)

Register	Offset	R/W	Description	Reset Value
UMCTL2_VER_TYPE	UMCTL2_BA+0xff4	R	UMCTL2 Version Type Register	0x6761_2a2a

31	30	29	28	27	26	25	24
ver_type							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ver_type							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ver_type							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ver_type							

Bits	Description	
[31:0]	**ver_type**	**Device Version Type Value** This is in ASCII format, with each byte corresponding to a character of the version type **Programming Mode:** Static

6.12 DDR 3/2 PHY Controller (DDR32PHY)

6.12.1 Overview

Portions Copyright (c) Synopsys, Inc. All rights reserved. Used with permission.

The DDR3/2 PHY is a DDR PHY controller and specific hardware configuration of Synopsys DDR3/2 PHY v1.50a.

The DDR3/2 PHY Utility Block (PUB) is a specific hardware configuration of Synopsys DDR3/2 PHY Utility Block (PUB) v3.18.

The DWC DDR3/2 PHY core, Synopsys' Double Data Rate 3 (DDR3) physical layer, provides an interface between the DDR3/DDR2 memory controller and external DDR3/DDR2 SDRAM devices.

The DWC DDR3/2 PHY is a complete mixed-signal IP solution designed to provide DDR 3/2 SDRAM connectivity. The DWC DDR3/2 PHY supports a range of DDR3 SDRAM speeds, from DDR3-667 through DDR3-1066, with backward compatibility provided for DDR2-667 through DDR2-1066 devices. Targeted toward supporting x16 DDR3 SDRAM components, DWC DDR3/2 PHY supports interfaces widths of 16 bits wide.

The DWC PHY Utility Block (PUB) provides control features to ease the customer implementation of digitally controlled features of the PHY such as initialization, DQS gate training, delay line calibration and VT compensation, write leveling, and programmable configuration controls. The PUB has built-in self-test features to provide support for production testing of the DWC PHY. It also provides a DFI interface to the PHY.

The PUB includes configuration registers that are accessible via a configuration port. The configuration port is an APB interface. A complete memory interface and control solution is achieved when the PUB is combined with Synopsys DWC DDR3/2 PHY and with either the Synopsys DesignWare Cores SDRAM Enhanced Universal DDR Memory Controller (uMCTL2).

6.12.2 Features

- Supports DDR3, DDR3L, and DDR2 operation
- Compatible with JEDEC standard DDR2/DDR3/DDR3L SDRAMs
- Supports a range of DDR3/DDR3L SDRAM speeds, from DDR3-667/DDR3L-667 through DDR3-1066/DDR3L-1066
- Supports a range of DDR2 SDRAM speeds, from DDR2-667 through DDR2-1066
- Maximum controller clock frequency of 266 MHz resulting in maximum SDRAM data rate of 1066Mbps
- Includes embedded PLL and DDLs necessary to meet timing specifications
- DDR3 PHY-Controller interface runs at 1/4 the memory baud rate, simplifying core logic timing constraints
- Write leveling delay line (WLDL) to compensate address and control versus data delays of up to 1 clock cycle or 2500ps
- Write and read bit delay lines (BDLs) compensate per-bit delay skew of up to 600ps at fast PVT; delay resolution approximately 15ps under typical conditions
- PHY Utility Block (PUB), a design that includes PHY control features, such as write leveling and data eye training, and provides support for production testing of the DWC DDR3/2 PHY
- SDRAM DLL off mode is not supported
- Data path width in 16-bits

- Support up to 2 memory ranks
- Support HDR mode only
- Single data channel configurations
- Complete PHY initialization, training and control
- Automatic DQS gate training
- Delay line calibrations and VT compensation
- Automatic write leveling
- Automatic read and write data bit de-skew
- Automatic DQ/DQS eye training
- PHY control and configuration registers
- APB interfaces to configuration registers
- DFI interface 2.1

6.12.2.1 Compatible Standards

Table 6.12-1 is compatible standards with the DWC DDR3/2 PHY design.

JESD79-2E	JEDEC DDR2 SDRAM Specification
JESD79-3C	JEDEC DDR3 SDRAM Specification
DFI v2.1	DDR PHY Interface (DFI) Specification

Table 6.12-1 Compatible Standards

6.12.3 Block Diagram

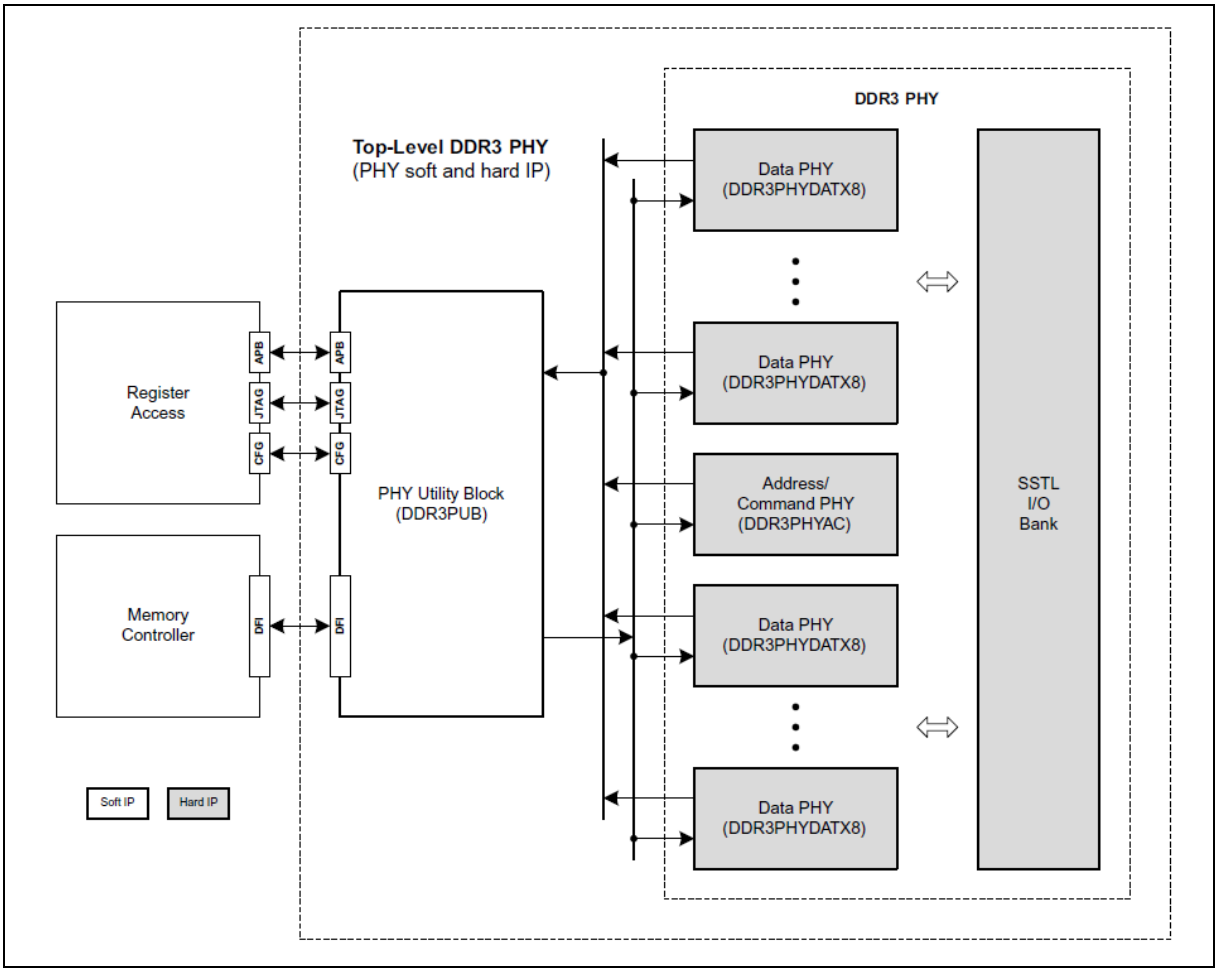


Figure 6.12-1 DDR 3/2 PHY Block Diagram

In Figure 6.12-1, the full DWC DDR3/2 PHY IP solution consists of multiple components:

- DDR3PHYAC: PHY component for the address/command lane (hereafter known as AC).
- DDR3PHYDATX8: PHY component for the data lane (hereafter known as DATX8).
- SSTL I/O library: the Stub Series Terminated Logic (SSTL) I/O library include PVT-compensated on-die termination (ODT) and output impedance.
- PHY Utility Block (PUB): a block that provides configuration, control, and other utility functions for all three PHY hard components.

6.12.4 DDR3/2 PHY SDRAM Interface

The DDR3/2 PHY SDRAM interfaces are listed in Table 6.12-2.

Pin Name	DDR3/3L (Non-MCP)	DDR3L (MCP)	DDR2 (Non-MCP)	DDR2 (MCP)	Description
MD8 to MD15	Yes	Yes	Yes	Yes	Byte lane 1 DQ
MDQS1_P	Yes	Yes	Yes	Yes	Byte lane 1 DQS
MDQS1_N	Yes	Yes	Yes	Yes	Byte lane 1 DQS#

MDMask1	Yes	Yes	Yes	Yes	Byte lane 1 DM
MD0 to MD7	Yes	Yes	Yes	Yes	Byte lane 0 DQ
MDQS0_P	Yes	Yes	Yes	Yes	Byte lane 0 DQS
MDQS0_N	Yes	Yes	Yes	Yes	Byte lane 0 DQS#
MDMask0	Yes	Yes	Yes	Yes	Byte lane 0 DM
MA0 to MA15	Yes	Yes	Yes	Yes	Row/Column address/ CA pins
MBA0 to MBA2	Yes	Yes	Yes	Yes	Bank address
MCKE0	Yes	Yes	Yes	Yes	CKE
MCKE1	Yes	No	Yes	No	CKE
MCK_P	Yes	Yes	Yes	Yes	CK
MCK_N	Yes	Yes	Yes	Yes	CK#
MRASn	Yes	Yes	Yes	Yes	RAS#
MCASn	Yes	Yes	Yes	Yes	CAS#
MWEn	Yes	Yes	Yes	Yes	WE#
MCS0n	Yes	Yes	Yes	Yes	CS#
MCS1n	Yes	No	Yes	No	CS#
MODT0	Yes	Yes	Yes	Yes	ODT
MODT1	Yes	No	Yes	No	ODT
MRESETn	Yes	Yes	No	No	RESET#
VREF_CA	Yes	Yes	Yes	Yes	Connect to (VDDQ/2)
VREF_DQ	Yes	Yes	Yes	Yes	Connect to (VDDQ/2)
MZQ_DDR	Yes	Yes	Yes	Yes	Connect to external RZQ =240 Ohm +/-1%

Table 6.12-2 DDR3/2 PHY I/O list

6.12.5 Basic Configuration

6.12.5.1 DDR PHY Basic Configuration

- Clock Source Configuration:
  - Set DDR PLL control registers in CLK_PLL2CTL0, CLK_PLL2CTL1, and CLK_PLL2CTL2.
- Reset Configuration:
  - Reset APB interface of DDR PUB in DDRPUBRST (SYS_IPRST0[30]).

6.12.6 Functional Description

6.12.6.1 Initialization

Figure 6.12-2 shows a high-level illustration of the initialization sequence of the PHY. A detailed sequence description and timing diagram is described in sections that follow. The initialization sequence has two phases. The first phase (shown in solid lines in Figure 6.12-2) happens automatically at reset and is as follows:

1. Before and during configuration reset (if cfg_rst_n is asserted), the PHY is un-initialized and remains in this state until the reset is de-asserted.
2. At reset de-assertion, the PHY moves into the PLL initialization (lock) phase. This phase may be bypassed at any time by writing a '1' to the PLL initialization bypass register bit (PIR[LOCKBYP]).
3. In parallel to PLL initialization, the impedance calibration phase also starts at reset de-assertion. This phase can also be bypassed by writing a '1' to the impedance calibration bypass register bit (PIR[ZCALBYP].
4. Once the PLL is initialized, the delay calibration phase starts. This phase can also be bypassed by writing a '1' to the delay calibration bypass register bit (PIR[DCALBYP].
5. Since the clocks to the PHY are not stable until after the PLL is locked and the delay lines are calibrated, the initialization phase waits a few programmable clock cycles in the PHY initialization reset phase before de-asserting the reset to the PHY.
6. Once the PLL initialization, impedance calibration and delay calibration phases are completed it means the PHY is initialized.

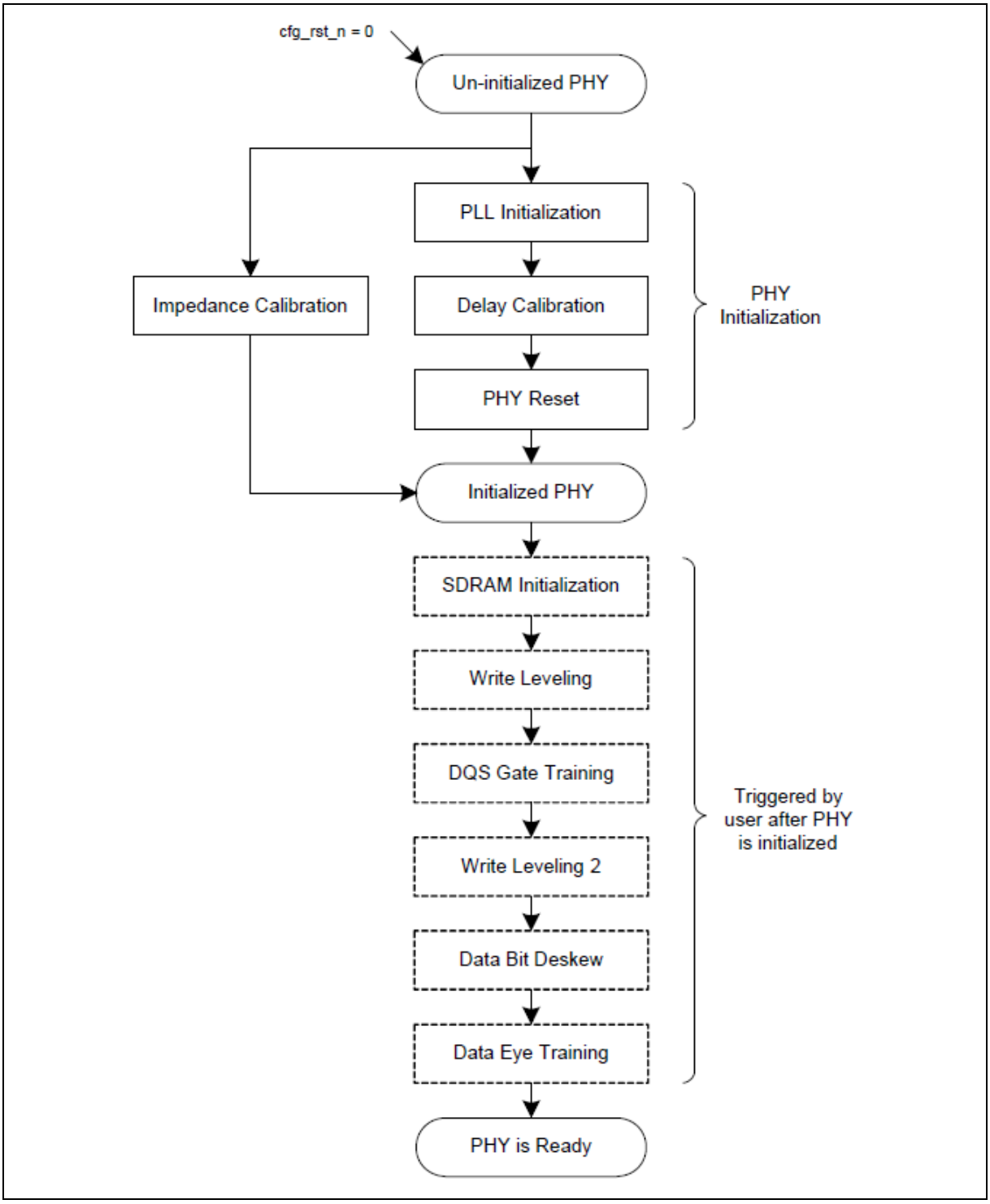


Figure 6.12-2 PHY Initialization Flow

6.12.6.2 SDRAM Initialization

Prior to normal operation, DDR SDRAMs must be initialized. The PUB has a built-in SDRAM initialization routine that may be triggered by software or memory controller by writing to the PHY Initialization

Register (PIR). The initialization routine built into the PUB is generic and does not require any knowledge of the type or configuration of external SDRAMs to be properly executed. The routine is designed with the relevant JEDEC specifications for the fastest and slowest SDRAMs supported by the PUB to result in a universal initialization sequence. This generic sequence is applicable to DDR3 and DDR2 SDRAMs. It is recommended to use the built-in PUB routine to initialize the SDRAM.

The following sections describe the built in SDRAM initialization routines for different types of SDRAMs. The necessary configurations of the SDRAM must first be programmed into the PUB before triggering SDRAM initialization. SDRAM initialization must be triggered only after the PHY has been initialized. All main SDRAM initialization timing parameters described below are programmable in controller clock cycles in PTR1 and PTR2 registers.

DDR3 Initialization Sequence:

The initialization steps for DDR3 SDRAMs are as follows:

1. Optionally maintain RESET# low for a minimum of either 200 us (power-up initialization) or 100ns (power-on initialization). The PUB drives RESET# low from the beginning of reset assertion and therefore this step may be skipped when DRAM initialization is triggered if enough time has expired to satisfy the RESET# low time.
2. After RESET# is de-asserted, wait a minimum of 500 us with CKE low.
3. Apply NOP and drive CKE high.
4. Wait a minimum of tXPR.
5. Issue a load Mode Register 2 (MR2) command.
6. Issue a load Mode Register 3 (MR3) command.
7. Issue a load Mode Register 1 (MR1) command (to set parameters and enable DLL).
8. Issue a load Mode Register 0 (MR0) command to set parameters and reset DLL.
9. Issue ZQ calibration command.
10. Wait 512 SDRAM clock cycles for the DLL to lock (tDLLK) and ZQ calibration (tZQinit) to finish. This wait time is relative to Step 8, i.e. relative to when the DLL reset command was issued onto the SDRAM command bus.

DDR2 Initialization Sequence:

The following are the initialization steps for DDR2 SDRAMs:

1. Wait a minimum of 200 us with CKE driven low.
2. Apply NOP and set CKE high.
3. Wait a minimum of 400 ns.
4. Issue a PRECHARGE ALL command.
5. Issue a load Extended Mode Register 2 (EMR2) command.
6. Issue a load Extended Mode Register 3 (EMR3) command.
7. Issue a load Extended Mode Register (EMR) command to enable DLL.
8. Issue a load Mode Register (MR) command to reset DLL.
9. Issue PRECHARGE ALL command.
10. Issue two REFRESH commands.
11. Issue a load Mode Register (MR) command to program device operating parameters without resetting the DLL.
12. Wait 200 SDRAM clock cycles for the DLL to lock. This wait time is relative to Step 8, i.e.

relative to when the DLL reset command was issued onto the SDRAM command bus.

13. Execute EMR OCD Default command.
14. Execute EMR OCD Exit command.

6.12.6.3 Initialization Trigger and Bypass

All initialization steps shown in Figure 6.12-2 can be triggered using the PHY Initialization Register (PIR). Writing a '1' to PIR[INIT] register bit will start initialization, with the routines to be run being selected by the corresponding PIR register bits. If multiple routines are selected, they are run in the order shown in Figure 6.12-2. This is also the order of the select bits in PIR register. The completion of the routines is indicated in the PHY General Status Register 0 (PGSR0) with the corresponding done status bits. This is described in “PHY General Status Registers 0-1 (PGSR0-1)”. The PGSR[IDONE] bit indicates the overall completion of the initialization sequence. An initialization done status register bit is cleared (reset to '0') when the corresponding routine is re-triggered.

The de-assertion of reset will automatically trigger the PUB to perform PLL initialization (locking), delay line calibration and impedance calibration. Once the PLL has locked, delay lines have been calibrated and impedance calibration has completed, the SDRAM initialization, write leveling, DQS gating, bit deskew and eye training may be triggered or performed by software or memory controller.

Since the PUB allows the selection of individual routines to be run when initialization is triggered using PIR register, only those routines that automatically trigger on reset de-assertion have individual bypass capability. This means that PLL locking, delay calibration and/or impedance calibration may be bypassed any time by writing a '1' to the corresponding bypass register bit in the PIR register. Once a routine is bypassed, it is internally registered as completed and the corresponding done status register bit is set in the PGSR0 register. It is up to the user to re-trigger or perform the bypassed routine at a later time before the PHY can be used. The PIR[INITBYP] register bit provides the option to bypass the whole initialization sequence.

6.12.6.4 Data Training

After the PHY and SDRAM have been successfully initialized, the PHY may be trained for optimum perating timing margins. This includes training for: write leveling, the training of the DQS gating during reads, bit deskew and the training of the read data eye.

6.12.6.5 DFI Interface

The DFI Interface block is used to connect the memory controller (MC) to the DDR PHY hard IP blocks (DDR3PHYAC, DDR3PHYDATX8, and SSTL I/Os). DFI requests from the MC enter the DFI interface and are converted to corresponding low-level PHY signals and timing.

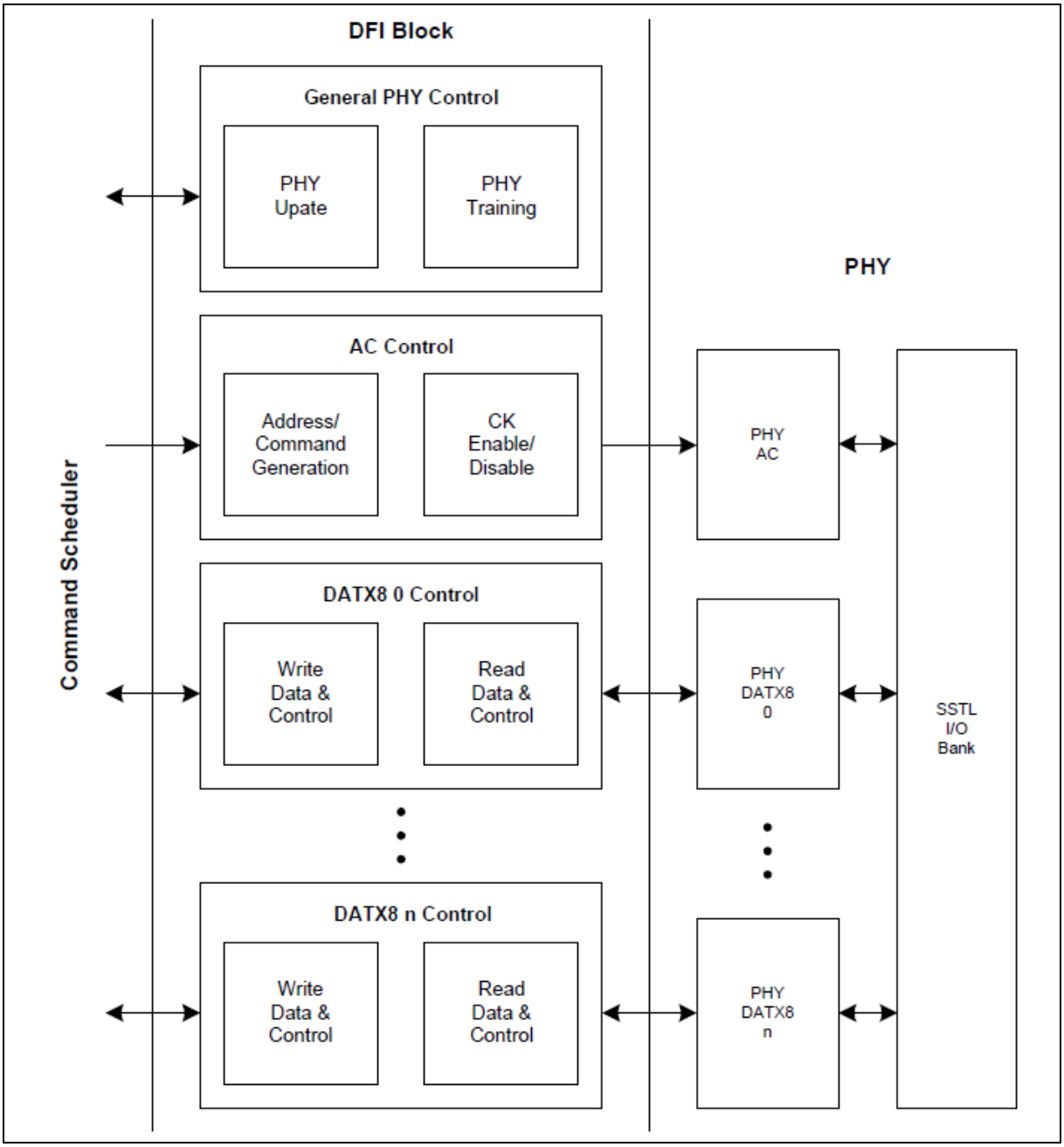


Figure 6.12-3 DFI Block Diagram

6.12.6.6 Configuration Port

The configuration port provides access to PUB registers with APB interface. The configuration clock (pclk) is asynchronous to the controller clock (ctl_clk), allowing it to operate at any frequency up to the frequency of the controller clock, and at any phase with respect to the controller clock. Before changing any PUB configuration, the user must ensure that all queued up commands that were issued through the controller have finished executing.

An APB interface can be selected to be the main access protocol for PUB configuration registers. The PUB APB interface is compatible to APB 2.0. Refer to the AMBA Specification Rev 2.0 for details about

the APB protocol.

6.12.6.7 DDR PHY Initialization for PLL, Delay Lines, and impedance of PHY SSTL I/Os

The initialization block is used to initialize the PHY and the SDRAM before the DDR interface can be used by the memory controller. Specifically, the initialization block performs four main tasks:

(1) Initializes the PHY PLLs. The PLLs have to be locked before the high speed logic in the PHY can be used.

(2) Calibrates and VT compensates the delay lines. The delay lines have to be calibrated before the high speed logic in the PHY can be used.

(3) Calibrates the impedance of the PHY SSTL I/Os.

(4) Executes the SDRAM initialization sequence.

Figure 6.12-4 shows the block diagram of the initialization block. The initialization block is also used to generate the timing parameters used for SDRAM command-to-command spacing based on the timing parameters and interface configurations programmed in the configuration registers.

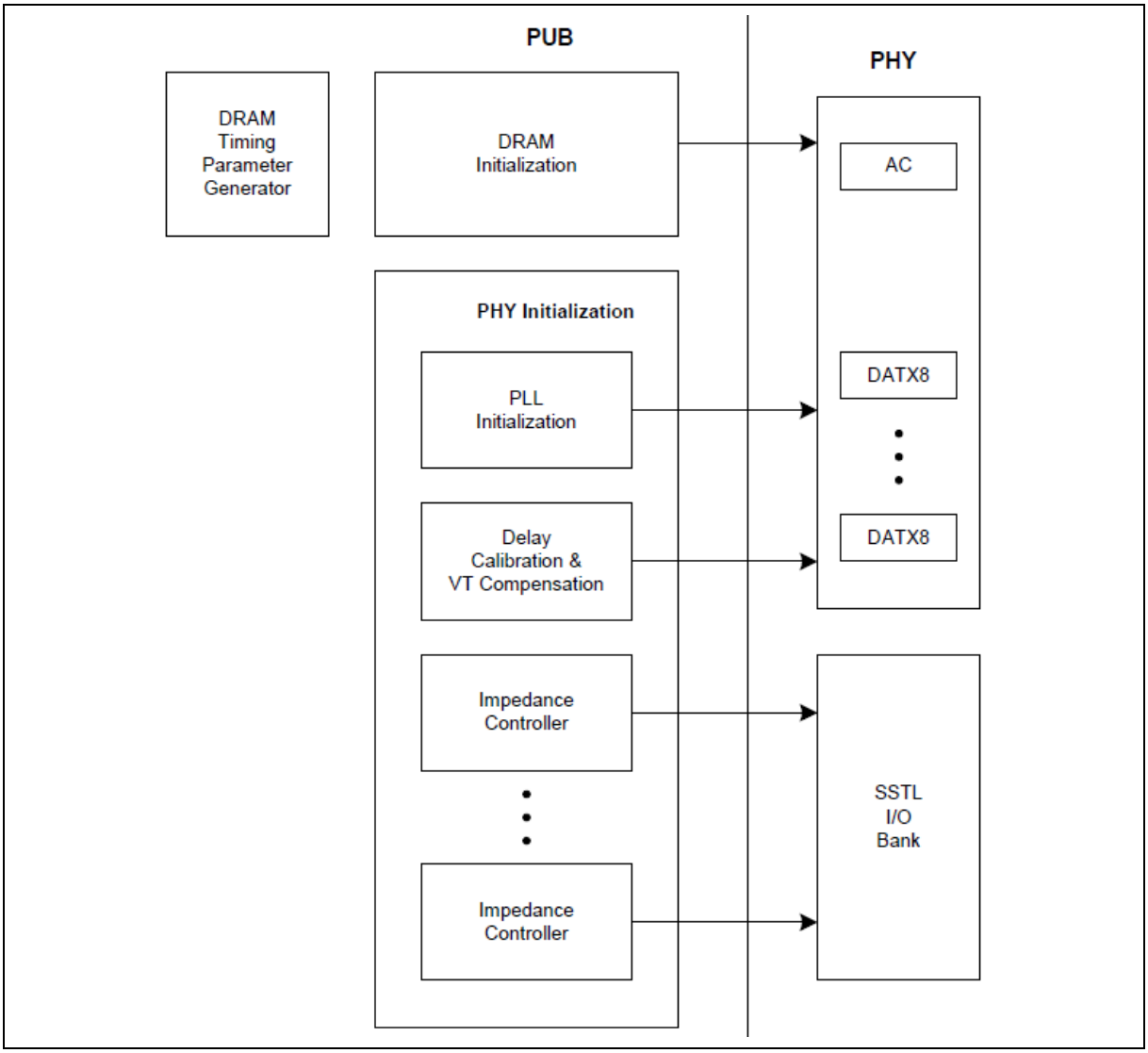


Figure 6.12-4 Initialization Block Diagram

6.12.7 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
DDRPHY Base Address: **DDRPHY_BA = 0x404C_0000**				
RIDR	DDRPHY_BA+0x000	R	Revision Identification Register	0x0013_1318
PIR	DDRPHY_BA+0x004	R/W	PHY Initialization Register	0x0000_0000
PGCR0	DDRPHY_BA+0x008	R/W	PHY General Configuration Registers 0	0xa800_3e3f
PGCR1	DDRPHY_BA+0x00c	R/W	PHY General Configuration Registers 1	0x0300_c461
PGSR0	DDRPHY_BA+0x010	R	PHY General Status Registers 0	0x0000_0000
PGSR1	DDRPHY_BA+0x014	R	PHY General Status Registers 1	0x0000_0000
PLLCR	DDRPHY_BA+0x018	R/W	PLL Control Register	0x0001_c000
PTR0	DDRPHY_BA+0x01c	R/W	PHY Timing Registers 0	0x42c2_1590
PTR1	DDRPHY_BA+0x020	R/W	PHY Timing Registers 1	0xd056_12c0
PTR2	DDRPHY_BA+0x024	R/W	PHY Timing Registers 2	0x0008_3def
PTR3	DDRPHY_BA+0x028	R/W	PHY Timing Registers 3	0x1808_2356
PTR4	DDRPHY_BA+0x02c	R/W	PHY Timing Registers 4	0x0aaf_4156
ACMDLR	DDRPHY_BA+0x030	R/W	AC Master Delay Line Register	0x0000_0000
ACBDLR	DDRPHY_BA+0x034	R/W	AC Bit Delay Line Register	0x0000_0000
ACIOCR	DDRPHY_BA+0x038	R/W	AC I/O Configuration Register	0x30c0_0812
DXCCR	DDRPHY_BA+0x03c	R/W	DATX8 Common Configuration Register	0x4418_1884
DSGCR	DDRPHY_BA+0x040	R/W	DDR System General Configuration Register	0xf004_649f
DCR	DDRPHY_BA+0x044	R/W	DRAM Configuration Register	0x0000_040b
DTPR0	DDRPHY_BA+0x048	R/W	DRAM Timing Parameters Register 0	0xc9e4_ee88

DTPR1	DDRPHY_BA+0x04c	R/W	DRAM Timing Parameters Register 1	0x228b_b4d2
DTPR2	DDRPHY_BA+0x050	R/W	DRAM Timing Parameters Register 2	0x1003_6a00
MR0	DDRPHY_BA+0x054	R/W	Mode Register 0	0x0000_0a52
MR1	DDRPHY_BA+0x058	R/W	Mode Register 1	0x0000_0000
MR2	DDRPHY_BA+0x05c	R/W	Mode Register 2/Extended Mode Register 2	0x0000_0000
MR3	DDRPHY_BA+0x060	R/W	Mode Register 3	0x0000_0000
ODTCR	DDRPHY_BA+0x064	R/W	ODT Configuration Register	0x0021_0000
DTCR	DDRPHY_BA+0x068	R/W	Data Training Configuration Register	0x9300_3587
DTAR0	DDRPHY_BA+0x06c	R/W	Data Training Address Register 0	0x0000_0000
DTAR1	DDRPHY_BA+0x070	R/W	Data Training Address Register 1	0x0000_0008
DTAR2	DDRPHY_BA+0x074	R/W	Data Training Address Register 2	0x0000_0010
DTAR3	DDRPHY_BA+0x078	R/W	Data Training Address Register 3	0x0000_0018
DTDR0	DDRPHY_BA+0x07c	R/W	Data Training Data Register 0	0xdd22_ee11
DTDR1	DDRPHY_BA+0x080	R/W	Data Training Data Register 1	0x7788_bb44
DTEDR0	DDRPHY_BA+0x084	R	Data Training Eye Data Register 0	0x0000_0000
DTEDR1	DDRPHY_BA+0x088	R	Data Training Eye Data Register 1	0x0000_0000
PGCR2	DDRPHY_BA+0x08c	R/W	PHY General Configuration Register 2	0x00f1_2480
RDIMMGCR0	DDRPHY_BA+0x0b0	R	RDIMM General Configuration Register 0	0x3c55_0000
RDIMMGCR1	DDRPHY_BA+0x0b4	R	RDIMM General Configuration Register 1	0xffff_0c80
RDIMMCR0	DDRPHY_BA+0x0b8	R	RDIMM Control Register 0	0x0000_0000
RDIMMCR1	DDRPHY_BA+0x0bc	R	RDIMM Control Register 1	0x0000_0000
DCUAR	DDRPHY_BA+0x0c0	R/W	DCU Address Register	0x0000_0000
DCUDR	DDRPHY_BA+0x0c4	R/W	DCU Data Register	0x0000_0000
DCURR	DDRPHY_BA+0x0c8	R/W	DCU Run Register	0x0000_0000
DCULR	DDRPHY_BA+0x0cc	R/W	DCU Loop Register	0xf000_0000
DCUGCR	DDRPHY_BA+0x0d0	R/W	DCU General Configuration Register	0x0000_0000
DCUTPR	DDRPHY_BA+0x0d4	R/W	DCU Timing Parameter Register	0x0000_0000
DCUSR0	DDRPHY_BA+0x0d8	R	DCU Status Register 0	0x0000_0000
DCUSR1	DDRPHY_BA+0x0dc	R	DCU Status Register 1	0x0000_0000
AACR	DDRPHY_BA+0x174	R/W	Anti-Aging Control Register	0x0000_00ff
GPR0	DDRPHY_BA+0x178	R/W	General Purpose Register 0	0x0000_0000
GPR1	DDRPHY_BA+0x17c	R/W	General Purpose Register 1	0x0000_0000
ZQ0CR0	DDRPHY_BA+0x180	R/W	Impedance Control Register 0 for ZQ0	0x4000_014a
ZQ0CR1	DDRPHY_BA+0x184	R/W	Impedance Control Register 1 for ZQ0	0x0000_107b

ZQ0SR0	DDRPHY_BA+0x188	R	Impedance Status Register 0 for ZQ0	0x0000_014a
ZQ0SR1	DDRPHY_BA+0x18c	R	Impedance Status Register 1 for ZQ0	0x0000_0000
ZQ1CR0	DDRPHY_BA+0x190	R/W	Impedance Control Register 0 for ZQ1	0x0000_0000
ZQ1CR1	DDRPHY_BA+0x194	R/W	Impedance Control Register 1 for ZQ1	0x0000_0000
ZQ1SR0	DDRPHY_BA+0x198	R	Impedance Status Register 0 for ZQ1	0x0000_0000
ZQ1SR1	DDRPHY_BA+0x19c	R	Impedance Status Register 1 for ZQ1	0x0000_0000
ZQ2CR0	DDRPHY_BA+0x1a0	R/W	Impedance Control Register 0 for ZQ2	0x0000_0000
ZQ2CR1	DDRPHY_BA+0x1a4	R/W	Impedance Control Register 1 for ZQ2	0x0000_0000
ZQ2SR0	DDRPHY_BA+0x1a8	R	Impedance Status Register 0 for ZQ2	0x0000_0000
ZQ2SR1	DDRPHY_BA+0x1ac	R	Impedance Status Register 1 for ZQ2	0x0000_0000
ZQ3CR0	DDRPHY_BA+0x1b0	R/W	Impedance Control Register 0 for ZQ3	0x0000_0000
ZQ3CR1	DDRPHY_BA+0x1b4	R/W	Impedance Control Register 1 for ZQ3	0x0000_0000
ZQ3SR0	DDRPHY_BA+0x1b8	R	Impedance Status Register 0 for ZQ3	0x0000_0000
ZQ3SR1	DDRPHY_BA+0x1bc	R	Impedance Status Register 1 for ZQ3	0x0000_0000
DX0GCR	DDRPHY_BA+0x1c0	R/W	DATX8 General Configuration Register for Byte Lane0	0x7c00_0e81
DX0GSR0	DDRPHY_BA+0x1c4	R	DATX8 General Status Registers 0 for Byte Lane0	0x0000_0000
DX0GSR1	DDRPHY_BA+0x1c8	R	DATX8 General Status Registers 1 for Byte Lane0	0x0000_0000
DX0BDLR0	DDRPHY_BA+0x1cc	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane0	0x0000_0000
DX0BDLR1	DDRPHY_BA+0x1d0	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane0	0x0000_0000
DX0BDLR2	DDRPHY_BA+0x1d4	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane0	0x0000_0000
DX0BDLR3	DDRPHY_BA+0x1d8	R/W	DATX8 Bit Delay Line Register 3 for Byte Lane0	0x0000_0000
DX0BDLR4	DDRPHY_BA+0x1dc	R/W	DATX8 Bit Delay Line Register 4 for Byte Lane0	0x0000_0000
DX0LCDLR0	DDRPHY_BA+0x1e0	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane0	0x0000_0000
DX0LCDLR1	DDRPHY_BA+0x1e4	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane0	0x0000_0000
DX0LCDLR2	DDRPHY_BA+0x1e8	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane0	0x0000_0000
DX0MDLR	DDRPHY_BA+0x1ec	R/W	DATX8 Master Delay Line Register for Byte Lane0	0x0000_0000
DX0GTR	DDRPHY_BA+0x1f0	R/W	DATX8 General Timing Register for Byte Lane0	0x0000_5000
DX0GSR2	DDRPHY_BA+0x1f4	R/W	DATX8 General Status Register 2 for Byte Lane0	0x0000_0000
DX1GCR	DDRPHY_BA+0x200	R/W	DATX8 General Configuration Register for Byte Lane1	0x7c00_0e81
DX1GSR0	DDRPHY_BA+0x204	R	DATX8 General Status Registers 0 for Byte Lane1	0x0000_0000
DX1GSR1	DDRPHY_BA+0x208	R	DATX8 General Status Registers 1 for Byte Lane1	0x0000_0000
DX1BDLR0	DDRPHY_BA+0x20c	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane1	0x0000_0000
DX1BDLR1	DDRPHY_BA+0x210	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane1	0x0000_0000
DX1BDLR2	DDRPHY_BA+0x214	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane1	0x0000_0000

DX1BDLR3	DDRPHY_BA+0x218	R/W	DATX8 Bit Delay Line Register 3 for Byte Lane1	0x0000_0000
DX1BDLR4	DDRPHY_BA+0x21c	R/W	DATX8 Bit Delay Line Register 4 for Byte Lane1	0x0000_0000
DX1LCDLR0	DDRPHY_BA+0x220	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane1	0x0000_0000
DX1LCDLR1	DDRPHY_BA+0x224	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane1	0x0000_0000
DX1LCDLR2	DDRPHY_BA+0x228	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane1	0x0000_0000
DX1MDLR	DDRPHY_BA+0x22c	R/W	DATX8 Master Delay Line Register for Byte Lane1	0x0000_0000
DX1GTR	DDRPHY_BA+0x230	R/W	DATX8 General Timing Register for Byte Lane1	0x0000_5000
DX1GSR2	DDRPHY_BA+0x234	R/W	DATX8 General Status Register 2 for Byte Lane1	0x0000_0000
DX2GCR	DDRPHY_BA+0x240	R/W	DATX8 General Configuration Register for Byte Lane2	0x0000_0000
DX2GSR0	DDRPHY_BA+0x244	R	DATX8 General Status Registers 0 for Byte Lane2	0x0000_0000
DX2GSR1	DDRPHY_BA+0x248	R	DATX8 General Status Registers 1 for Byte Lane2	0x0000_0000
DX2BDLR0	DDRPHY_BA+0x24c	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane2	0x0000_0000
DX2BDLR1	DDRPHY_BA+0x250	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane2	0x0000_0000
DX2BDLR2	DDRPHY_BA+0x254	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane2	0x0000_0000
DX2BDLR3	DDRPHY_BA+0x258	R/W	DATX8 Bit Delay Line Register 3 for Byte Lane2	0x0000_0000
DX2BDLR4	DDRPHY_BA+0x25c	R/W	DATX8 Bit Delay Line Register 4 for Byte Lane2	0x0000_0000
DX2LCDLR0	DDRPHY_BA+0x260	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane2	0x0000_0000
DX2LCDLR1	DDRPHY_BA+0x264	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane2	0x0000_0000
DX2LCDLR2	DDRPHY_BA+0x268	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane2	0x0000_0000
DX2MDLR	DDRPHY_BA+0x26c	R/W	DATX8 Master Delay Line Register for Byte Lane2	0x0000_0000
DX2GTR	DDRPHY_BA+0x270	R/W	DATX8 General Timing Register for Byte Lane2	0x0000_0000
DX2GSR2	DDRPHY_BA+0x274	R/W	DATX8 General Status Register 2 for Byte Lane2	0x0000_0000
DX3GCR	DDRPHY_BA+0x280	R/W	DATX8 General Configuration Register for Byte Lane3	0x0000_0000
DX3GSR0	DDRPHY_BA+0x284	R	DATX8 General Status Registers 0 for Byte Lane3	0x0000_0000
DX3GSR1	DDRPHY_BA+0x288	R	DATX8 General Status Registers 1 for Byte Lane3	0x0000_0000
DX3BDLR0	DDRPHY_BA+0x28c	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane3	0x0000_0000
DX3BDLR1	DDRPHY_BA+0x290	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane3	0x0000_0000
DX3BDLR2	DDRPHY_BA+0x294	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane3	0x0000_0000
DX3BDLR3	DDRPHY_BA+0x298	R/W	DATX8 Bit Delay Line Register 3 for Byte Lane3	0x0000_0000
DX3BDLR4	DDRPHY_BA+0x29c	R/W	DATX8 Bit Delay Line Register 4 for Byte Lane3	0x0000_0000
DX3LCDLR0	DDRPHY_BA+0x2a0	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane3	0x0000_0000
DX3LCDLR1	DDRPHY_BA+0x2a4	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane3	0x0000_0000
DX3LCDLR2	DDRPHY_BA+0x2a8	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane3	0x0000_0000
DX3MDLR	DDRPHY_BA+0x2ac	R/W	DATX8 Master Delay Line Register for Byte Lane3	0x0000_0000

DX3GTR	DDRPHY_BA+0x2b0	R/W	DATX8 General Timing Register for Byte Lane3	0x0000_0000
DX3GSR2	DDRPHY_BA+0x2b4	R/W	DATX8 General Status Register 2 for Byte Lane3	0x0000_0000
DX4GCR	DDRPHY_BA+0x2c0	R/W	DATX8 General Configuration Register for Byte Lane4	0x0000_0000
DX4GSR0	DDRPHY_BA+0x2c4	R	DATX8 General Status Registers 0 for Byte Lane4	0x0000_0000
DX4GSR1	DDRPHY_BA+0x2c8	R	DATX8 General Status Registers 1 for Byte Lane4	0x0000_0000
DX4BDLR0	DDRPHY_BA+0x2cc	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane4	0x0000_0000
DX4BDLR1	DDRPHY_BA+0x2d0	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane4	0x0000_0000
DX4BDLR2	DDRPHY_BA+0x2d4	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane4	0x0000_0000
DX4BDLR3	DDRPHY_BA+0x2d8	R/W	DATX8 Bit Delay Line Register 3 for Byte Lane4	0x0000_0000
DX4BDLR4	DDRPHY_BA+0x2dc	R/W	DATX8 Bit Delay Line Register 4 for Byte Lane4	0x0000_0000
DX4LCDLR0	DDRPHY_BA+0x2e0	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane4	0x0000_0000
DX4LCDLR1	DDRPHY_BA+0x2e4	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane4	0x0000_0000
DX4LCDLR2	DDRPHY_BA+0x2e8	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane4	0x0000_0000
DX4MDLR	DDRPHY_BA+0x2ec	R/W	DATX8 Master Delay Line Register for Byte Lane4	0x0000_0000
DX4GTR	DDRPHY_BA+0x2f0	R/W	DATX8 General Timing Register for Byte Lane4	0x0000_0000
DX4GSR2	DDRPHY_BA+0x2f4	R/W	DATX8 General Status Register 2 for Byte Lane4	0x0000_0000
DX5GCR	DDRPHY_BA+0x300	R/W	DATX8 General Configuration Register for Byte Lane5	0x0000_0000
DX5GSR0	DDRPHY_BA+0x304	R	DATX8 General Status Registers 0 for Byte Lane5	0x0000_0000
DX5GSR1	DDRPHY_BA+0x308	R	DATX8 General Status Registers 1 for Byte Lane5	0x0000_0000
DX5BDLR0	DDRPHY_BA+0x30c	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane5	0x0000_0000
DX5BDLR1	DDRPHY_BA+0x310	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane5	0x0000_0000
DX5BDLR2	DDRPHY_BA+0x314	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane5	0x0000_0000
DX5BDLR3	DDRPHY_BA+0x318	R/W	DATX8 Bit Delay Line Register 3 for Byte Lane5	0x0000_0000
DX5BDLR4	DDRPHY_BA+0x31c	R/W	DATX8 Bit Delay Line Register 4 for Byte Lane5	0x0000_0000
DX5LCDLR0	DDRPHY_BA+0x320	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane5	0x0000_0000
DX5LCDLR1	DDRPHY_BA+0x324	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane5	0x0000_0000
DX5LCDLR2	DDRPHY_BA+0x328	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane5	0x0000_0000
DX5MDLR	DDRPHY_BA+0x32c	R/W	DATX8 Master Delay Line Register for Byte Lane5	0x0000_0000
DX5GTR	DDRPHY_BA+0x330	R/W	DATX8 General Timing Register for Byte Lane5	0x0000_0000
DX5GSR2	DDRPHY_BA+0x334	R/W	DATX8 General Status Register 2 for Byte Lane5	0x0000_0000
DX6GCR	DDRPHY_BA+0x340	R/W	DATX8 General Configuration Register for Byte Lane6	0x0000_0000
DX6GSR0	DDRPHY_BA+0x344	R	DATX8 General Status Registers 0 for Byte Lane6	0x0000_0000
DX6GSR1	DDRPHY_BA+0x348	R	DATX8 General Status Registers 1 for Byte Lane6	0x0000_0000
DX6BDLR0	DDRPHY_BA+0x34c	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane6	0x0000_0000

DX6BDLR1	DDRPHY_BA+0x350	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane6	0x0000_0000
DX6BDLR2	DDRPHY_BA+0x354	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane6	0x0000_0000
DX6BDLR3	DDRPHY_BA+0x358	R/W	DATX8 Bit Delay Line Register 3 for Byte Lane6	0x0000_0000
DX6BDLR4	DDRPHY_BA+0x35c	R/W	DATX8 Bit Delay Line Register 4 for Byte Lane6	0x0000_0000
DX6LCDLR0	DDRPHY_BA+0x360	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane6	0x0000_0000
DX6LCDLR1	DDRPHY_BA+0x364	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane6	0x0000_0000
DX6LCDLR2	DDRPHY_BA+0x368	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane6	0x0000_0000
DX6MDLR	DDRPHY_BA+0x36c	R/W	DATX8 Master Delay Line Register for Byte Lane6	0x0000_0000
DX6GTR	DDRPHY_BA+0x370	R/W	DATX8 General Timing Register for Byte Lane6	0x0000_0000
DX6GSR2	DDRPHY_BA+0x374	R/W	DATX8 General Status Register 2 for Byte Lane6	0x0000_0000
DX7GCR	DDRPHY_BA+0x380	R/W	DATX8 General Configuration Register for Byte Lane7	0x0000_0000
DX7GSR0	DDRPHY_BA+0x384	R	DATX8 General Status Registers 0 for Byte Lane7	0x0000_0000
DX7GSR1	DDRPHY_BA+0x388	R	DATX8 General Status Registers 1 for Byte Lane7	0x0000_0000
DX7BDLR0	DDRPHY_BA+0x38c	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane7	0x0000_0000
DX7BDLR1	DDRPHY_BA+0x390	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane7	0x0000_0000
DX7BDLR2	DDRPHY_BA+0x394	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane7	0x0000_0000
DX7BDLR3	DDRPHY_BA+0x398	R/W	DATX8 Bit Delay Line Register 3 for Byte Lane7	0x0000_0000
DX7BDLR4	DDRPHY_BA+0x39c	R/W	DATX8 Bit Delay Line Register 4 for Byte Lane7	0x0000_0000
DX7LCDLR0	DDRPHY_BA+0x3a0	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane7	0x0000_0000
DX7LCDLR1	DDRPHY_BA+0x3a4	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane7	0x0000_0000
DX7LCDLR2	DDRPHY_BA+0x3a8	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane7	0x0000_0000
DX7MDLR	DDRPHY_BA+0x3ac	R/W	DATX8 Master Delay Line Register for Byte Lane7	0x0000_0000
DX7GTR	DDRPHY_BA+0x3b0	R/W	DATX8 General Timing Register for Byte Lane7	0x0000_0000
DX7GSR2	DDRPHY_BA+0x3b4	R/W	DATX8 General Status Register 2 for Byte Lane7	0x0000_0000
DX8GCR	DDRPHY_BA+0x3c0	R/W	DATX8 General Configuration Register for Byte Lane8	0x0000_0000
DX8GSR0	DDRPHY_BA+0x3c4	R	DATX8 General Status Registers 0 for Byte Lane8	0x0000_0000
DX8GSR1	DDRPHY_BA+0x3c8	R	DATX8 General Status Registers 1 for Byte Lane8	0x0000_0000
DX8BDLR0	DDRPHY_BA+0x3cc	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane8	0x0000_0000
DX8BDLR1	DDRPHY_BA+0x3d0	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane8	0x0000_0000
DX8BDLR2	DDRPHY_BA+0x3d4	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane8	0x0000_0000
DX8BDLR3	DDRPHY_BA+0x3d8	R/W	DATX8 Bit Delay Line Register 3 for Byte Lane8	0x0000_0000
DX8BDLR4	DDRPHY_BA+0x3dc	R/W	DATX8 Bit Delay Line Register 4 for Byte Lane8	0x0000_0000
DX8LCDLR0	DDRPHY_BA+0x3e0	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane8	0x0000_0000
DX8LCDLR1	DDRPHY_BA+0x3e4	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane8	0x0000_0000

DX8LCDLR2	DDRPHY_BA+0x3e8	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane8	0x0000_0000
DX8MDLR	DDRPHY_BA+0x3ec	R/W	DATX8 Master Delay Line Register for Byte Lane8	0x0000_0000
DX8GTR	DDRPHY_BA+0x3f0	R/W	DATX8 General Timing Register for Byte Lane8	0x0000_0000
DX8GSR2	DDRPHY_BA+0x3f4	R/W	DATX8 General Status Register 2 for Byte Lane8	0x0000_0000

6.12.8 Register Description

RIDR Register (RIDR)

Register	Offset	R/W	Description	Reset Value
RIDR	DDRPHY_BA+0x000	R	Revision Identification Register	0x0013_1318

31	30	29	28	27	26	25	24
UDRID							
23	22	21	20	19	18	17	16
PHYMJR				PHYMDR			
15	14	13	12	11	10	9	8
PHYMNR				PUBMJR			
7	6	5	4	3	2	1	0
PUBMDR			PUBMNR				

Bits	Description	
[31:24]	**UDRID**	**User-Defined Revision ID** General purpose revision identification set by the user.
[23:20]	**PHYMJR**	**PHY Major Revision** Indicates major revision of the PHY such addition of the features that make the new version not compatible with previous versions.
[19:16]	**PHYMDR**	**PHY Moderate Revision** Indicates moderate revision of the PHY such as addition of new features. Normally the new version is still compatible with previous versions.
[15:12]	**PHYMNR**	**PHY Minor Revision** Indicates minor update of the PHY such as bug fixes. Normally no new features are included.
[11:8]	**PUBMJR**	**PUB Major Revision** Indicates major revision of the PUB such addition of the features that make the new version not compatible with previous versions.
[7:4]	**PUBMDR**	**PUB Moderate Revision** Indicates moderate revision of the PUB such as addition of new features. Normally the new version is still compatible with previous versions.
[3:0]	**PUBMNR**	**PUB Minor Revision** Indicates minor update of the PUB such as bug fixes. Normally no new features are included.

PIR Register (PIR)

Register	Offset	R/W	Description	Reset Value
PIR	DDRPHY_BA+0x004	R/W	PHY Initialization Register	0x0000_0000

31	30	29	28	27	26	25	24
INITBYP	ZCALBYP	DCALBYP	LOCKBYP	CLRSR	Reserved		
23	22	21	20	19	18	17	16
Reserved				RDIMMINIT	CTLDINIT	PLLBYP	ICPC
15	14	13	12	11	10	9	8
WREYE	RDEYE	WRDSKW	RDDSKW	WLADJ	QSGATE	WL	DRAMINIT
7	6	5	4	3	2	1	0
DRAMRST	PHYRST	DCAL	PLLINIT	Reserved		ZCAL	INIT

Bits	Description	
[31]	**INITBYP**	**Initialization Bypass** Bypasses or stops, if set, all initialization routines currently running, including PHY initialization, DRAM initialization, and PHY training. Initialization may be triggered manually using INIT and the other relevant bits of the PIR register. This bit is self-clearing.
[30]	**ZCALBYP**	**Impedance Calibration Bypass** Bypasses or stops, if set, impedance calibration of all ZQ control blocks that automatically triggers after reset. Impedance calibration may be triggered manually using INIT and ZCAL bits of the PIR register. This bit is self-clearing.
[29]	**DCALBYP**	**Digital Delay Line Calibration Bypass** Bypasses or stops, if set, DDL calibration that automatically triggers after reset. DDL calibration may be triggered manually using INIT and DCAL bits of the PIR register. This bit is self- clearing.
[28]	**LOCKBYP**	**PLL Lock Bypass** Bypasses or stops, if set, the waiting of PLLs to lock. PLL lock wait is automatically triggered after reset. PLL lock wait may be triggered manually using INIT and PLLINIT bits of the PIR register. This bit is self-clearing.

[27]	**CLRSR**	**Clear Status Registers** Writing 1 to this bit clears (reset to 0) select status bits in register PGSR0. This bit is primarily for debug purposes and is typically not needed during normal functional operation. It can be used when PGSR.IDONE=1, to manually clear a selection of the PGSR status bits, although starting a new initialization process (PIR[0].INIT = 1'b1) automatically clears the PGSR status bits associated with the initialization steps enabled. The following list describes which bits within the PGSR0 register are cleared when CLRSR is set to 1'b1 and which bits are not cleared: ■ The following bits are not cleared by PIR[27] (CLRSR): - PGSR0[31] (APLOCK) - PGSR0[29:28] (PLDONE_CHN) - PGSR0[23] (WLAERR) - PGSR0[21] (WLERR) - PGSR0[4] (DIDONE) - PGSR0[2] (DCDONE) - PGSR0[1] (PLDONE) - PGSR0[0] (IDONE) ■ The following bits are always zero: - PGSR0[30] (reserved) - PGSR0[19:12] (reserved) ■ The following bits are cleared unconditionally by PIR[27] (CLRSR): - PGSR0[27] (WEERR) - PGSR0[26] (REERR) - PGSR0[25] (WDERR) - PGSR0[24] (RDERR) - PGSR0[22] (QSGERR) - PGSR0[20] (ZCERR) - PGSR0[11] (WEDONE) - PGSR0[10] (REDONE) - PGSR0[9] (WDDONE) - PGSR0[8] (RDDONE) - PGSR0[7] (WLADONE) - PGSR0[6] (QSGDONE) - PGSR0[5] (WLDONE) - PGSR0[3] (ZCDONE)
[26:20]	**Reserved**	Return zeroes on reads.
[19]	**RDIMMINIT**	**RDIMM Initialization** Executes the RDIMM buffer chip initialization before executing DRAM initialization. The RDIMM buffer chip initialization is run after the DRAM is reset and CKE have been driven high by the DRAM initialization sequence.
[18]	**CTLDINIT**	**Controller DRAM Initialization** Indicates, if set, that DRAM initialization will be performed by the controller. Otherwise if not set it indicates that DRAM initialization will be performed using the built-in initialization sequence or using software through the configuration port.
[17]	**PLLBYP**	**PLL Bypass** A setting of 1 on this bit will put all PHY PLLs in bypass mode.

[16]	**ICPC**	**Initialization Complete Pin Configuration** Specifies how the DFI initialization complete output pin (dfi_init_complete) should be used to indicate the status of initialization. Valid values are: 0 = Asserted after PHY initialization (DLL locking and impedance calibration) is complete. 1 = Asserted after PHY initialization is complete and the triggered the PUB initialization (DRAM initialization, data training, or initialization trigger with no selected initialization) is complete.
[15]	**WREYE**	**Write Data Eye Training** Executes a PUB training routine to maximize the write data eye.
[14]	**RDEYE**	**Read Data Eye Training** Executes a PUB training routine to maximize the read data eye.
[13]	**WRDSKW**	**Write Data Bit Deskew** Executes a PUB training routine to deskew the DQ bits during write.
[12]	**RDDSKW**	**Read Data Bit Deskew** Executes a PUB training routine to deskew the DQ bits during read.
[11]	**WLADJ**	**Write Leveling Adjust for DDR3 Only** Executes a PUB training routine that re- adjusts the write latency used during write in case the write leveling routine changed the expected latency. **Note:** Ensure that the DCU command cache is cleared prior to running WLADJ.
[10]	**QSGATE**	**Read DQS Gate Training** Executes a PUB training routine to determine the optimum position of the read data DQS strobe for maximum system timing margins.
[9]	**WL**	**Write Leveling for DDR3 Only** Executes a PUB write leveling routine.
[8]	**DRAMINIT**	**DRAM Initialization** Executes the DRAM initialization sequence.
[7]	**DRAMRST**	**DRAM Reset for DDR3 Only** Issues a reset to the DRAM (by driving the DRAM reset pin low) and wait 200us. This can be triggered in isolation or with the full DRAM initialization (DRAMINIT). For the latter case, the reset is issued and 200us is waited before starting the full initialization sequence.
[6]	**PHYRST**	**PHY Reset** Resets the AC and DATX8 modules by asserting the AC/DATX8 reset pin.
[5]	**DCAL**	**Digital Delay Line Calibration** Performs PHY delay line calibration.
[4]	**PLLINIT**	**PLL Initialization** Executes the PLL initialization sequence which includes correct driving of PLL power-down, reset and gear shift pins, and then waiting for the PHY PLLs to lock.
[3:2]	**Reserved**	Return zeroes on reads.
[1]	**ZCAL**	**Impedance Calibration** Performs PHY impedance calibration. When set the impedance calibration will be performed in parallel with PHY initialization (PLL initialization + DDL calibration + PHY reset).

[0]	**INIT**	**Initialization Trigger** A write of '1' to this bit triggers the DDR system initialization, including PHY initialization, DRAM initialization, and PHY training. The exact initialization steps to be executed are specified in bits 1 to 15 of this register. A bit setting of 1 means the step will be executed as part of the initialization sequence, while a setting of '0' means the step will be bypassed. The initialization trigger bit is self-clearing.

PGCR0 Register (PGCR0)

Register	Offset	R/W	Description	Reset Value
PGCR0	DDRPHY_BA+0x008	R/W	PHY General Configuration Registers 0	0xa800_3e3f

31	30	29	28	27	26	25	24
CKEN						PUBMODE	Reserved
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved					DTOSEL		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DTOSEL		OSCWDL		OSCDIV			OSCEN
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DLTST	DLTMODE	RDBVT	WDBVT	RGLVT	RDLVT	WDLVT	WLLVT

Bits	Description	
[31:26]	**CKEN**	**CK Enable** Controls whether the CK going to the SDRAM is enabled (toggling) or disabled (static value) and whether the CK is inverted. Two bits for each of the up to three CK pairs. Valid values for the two bits are: 00 = CK disabled (Driven to constant 0). 01 = CK toggling with inverted polarity. 10 = CK toggling with normal polarity (This should be the default setting). 11 = CK disabled (Driven to constant 1).
[25]	**PUBMODE**	**PUB Mode Enable** Enables, if set, the PUB to control the interface to the PHY and SDRAM. In this mode the DFI commands from the controller are ignored. The bit must be set to 0 after the system determines it is convenient to pass control of the DFI bus to the controller. When set to 0 the DFI interface has control of the PHY and SDRAM interface except when triggering pub operations such as BIST, DCU or data training.
[24:19]	**Reserved**	Return zeroes on reads.

[18:14]	**DTOSEL**	**Digital Test Output Select** Selects the PHY digital test output that is driven onto PHY digital test output (phy_dto) pin: Valid values are: 00000 = DATX8 0 PLL digital test output. 00001 = DATX8 1 PLL digital test output. 00010 = DATX8 2 PLL digital test output. 00011 = DATX8 3 PLL digital test output. 00100 = DATX8 4 PLL digital test output. 00101 = DATX8 5 PLL digital test output. 00110 = DATX8 6 PLL digital test output. 00111 = DATX8 7 PLL digital test output. 01000 = DATX8 8 PLL digital test output. 01001 = AC PLL digital test output. 01010 – 01111 = Reserved. 10000 = DATX8 0 delay line digital test output. 10001 = DATX8 1 delay line digital test output. 10010 = DATX8 2 delay line digital test output. 10011 = DATX8 3 delay line digital test output. 10100 = DATX8 4 delay line digital test output. 10101 = DATX8 5 delay line digital test output. 10110 = DATX8 6 delay line digital test output. 10111 = DATX8 7 delay line digital test output. 11000 = DATX8 8 delay line digital test output. 11001 = AC delay line digital test output. 11010 – 11111 = Reserved.
[13:12]	**OSCWDL**	**Oscillator Mode Write-Leveling Delay Line Select** Selects which of the two write leveling LCDLs is active. The delay select value of the inactive LCDL is set to zero while the delay select value of the active LCDL can be varied by the input write leveling delay select pin. Valid values are: 00 = No WL LCDL is active. 01 = DDR WL LCDL is active. 10 = SDR WL LCDL is active. 11 = Both LCDLs are active.
[11:9]	**OSCDIV**	**Oscillator Mode Division** Specifies the factor by which the delay line oscillator mode output is divided down before it is output on the delay line digital test output pin dl_dto. Valid values are: 000 = Divide by 1. 001 = Divide by 256. 010 = Divide by 512. 011 = Divide by 1024. 100 = Divide by 2048. 101 = Divide by 4096. 110 = Divide by 8192. 111 = Divide by 65536.
[8]	**OSCEN**	Oscillator Enable: Enables, if set, the delay line oscillation.
[7]	**DLTST**	**Delay Line Test Start** A write of '1' to this bit will trigger delay line oscillator mode period measurement. This bit is not self clearing and needs to be reset to '0' before the measurement can be re-triggered.

[6]	**DLTMODE**	**Delay Line Test Mode** Selects, if set, the delay line oscillator test mode. Setting this bit also clears all delay line register values. For DL oscillator testing, first set this bit, then apply desired non-zero LCDL and BDL register programmings.
[5]	**RDBVT**	**Read Data BDL VT Compensation** Enables, if set, the VT drift compensation of the read data bit delay lines.
[4]	**WDBVT**	**Write Data BDL VT Compensation** Enables, if set, the VT drift compensation of the write data bit delay lines.
[3]	**RGLVT**	**Read DQS Gating LCDL Delay VT Compensation** Enables, if set, the VT drift compensation of the read DQS gating LCDL.
[2]	**RDLVT**	**Read DQS LCDL Delay VT Compensation** Enables, if set, the VT drift compensation of the read DQS LCDL.
[1]	**WDLVT**	**Write DQ LCDL Delay VT Compensation** Enables, if set, the VT drift compensation of the write DQ LCDL.
[0]	**WLLVT**	**Write Leveling LCDL Delay VT Compensation** Enables, if set, the VT drift compensation of the write leveling LCDL.

PGCR1 Register (PGCR1)

Register	Offset	R/W	Description	Reset Value
PGCR1	DDRPHY_BA+0x00c	R/W	PHY General Configuration Registers 1	0x0300_c461

31	30	29	28	27	26	25	24
LBMODE	LBGDQS		LBDQSS	IOLB	INHVT	DXHRST	ZCKSEL
23	22	21	20	19	18	17	16
ZCKSEL	DLDLMT						
15	14	13	12	11	10	9	8
DLDLMT	FDEPTH		LPFDEPTH		LPFEN	MDLEN	IODDRM
7	6	5	4	3	2	1	0
IODDRM	WLSELT	ACHRST	WSLOPT	Reserved	WLSTEP	WLMODE	PDDISDX

Bits	Description	
[31]	**LBMODE**	**Loopback Mode** Indicates, if set, that the PHY/PUB is in loopback mode.
[30:29]	**LBGDQS**	**Loopback DQS Gating** Selects the DQS gating mode that should be used when the PHY is in loopback mode, including BIST loopback mode. Valid values are: 2'b00 = DQS gate is always on. 2'b01 = DQS gate training will be triggered on the PUB. 2'b10 = DQS gate is set manually using software. 2'b11 = Reserved.
[28]	**LBDQSS**	**Loopback DQS Shift** Selects how the read DQS is shifted during loopback to ensure that the read DQS is centered into the read data eye. Valid values are: 1'b0 = PUB sets the read DQS LCDL to 0 (internally). DQS is already shifted 90 degrees by write path 1'b1 = The read DQS shift is set manually through software.
[27]	**IOLB**	**I/O Loop-Back Select** Selects where inside the I/O the loop-back of signals happens. Valid values are: 1'b0 = Loopback is after output buffer; output enable must be asserted. 1'b1 = Loopback is before output buffer; output enable is don't care.
[26]	**INHVT**	**VT Calculation Inhibit** Inhibits calculation of the next VT compensated delay line values. A value of 1 will inhibit the VT calculation. This bit should be set to 1 during writes to the delay line registers.
[25]	**DXHRST**	**DX PHY High-Speed Reset** a Write of '0' to this bit resets the DX macro without resetting the PUB RTL logic. This bit is not self-clearing and a '1' must be written to de-assert the reset.

[24:23]	**ZCKSEL**	**Impedance Clock Divider Select** Selects the divide ratio for the clock used by the impedance control logic relative to the clock used by the memory controller and SDRAM. Valid values are: 00 = Divide by 2. 01 = Divide by 8. 10 = Divide by 32. 11 = Divide by 64.
[22:15]	**DLDLMT**	**Delay Line VT Drift Limit** Specifies the minimum change in the delay line VT drift in one direction which should result in the assertion of the delay line VT drift status signal (vt_drift). The limit is specified in terms of delay select values. A value of 0 disables the assertion of delay line VT drift status signal.
[14:13]	**FDEPTH**	**Filter Depth** Specifies the number of measurements over which all AC and DATX8 initial period measurements, that happen after reset or when calibration is manually triggered, are averaged. Valid values are: 00 = 2. 01 = 4. 10 = 8. 11 = 16.
[12:11]	**LPFDEPTH**	**Low-Pass Filter Depth** Specifies the number of measurements over which MDL period measurements are filtered. This determines the time constant of the low pass filter. Valid values are: 00 = 2. 01 = 4. 10 = 8. 11 = 16.
[10]	**LPFEN**	**Low-Pass Filter Enable** Enables, if set, the low pass filtering of MDL period measurements.
[9]	**MDLEN**	**Master Delay Line Enable** Enables, if set, the AC master delay line calibration to perform subsequent period measurements following the initial period measurements that are performed after reset or on when calibration is manually triggered. These additional measurements are accumulated and filtered as long as this bit remains high.
[8:7]	**IODDRM**	**I/O DDR Mode for D3F I/O Only** Selects the DDR mode for the I/Os. These bits connect to bits [2:1] of the IOM pin of the SSTL I/O. For more information, refer to the SSTL I/O chapter in the DWC DDR PHY Databook.
[6]	**WLSELT**	**Write Leveling Select Type** Selects the encoding type for the write leveling select signal depending on the desired setup/hold margins for the internal pipelines. Refer to the *DDR PHY Databook* for details of how the select type is used. Valid values are: 0 = Type 1: Setup margin of 90 degrees and hold margin of 90 degrees. 1 = Type 2: Setup margin of 135 degrees and hold margin of 45 degrees.
[5]	**ACHRST**	**AC PHY High-Speed Reset** A Write of '0' to this bit resets the AC macro without resetting the PUB RTL logic. This bit is not self-clearing and a '1' must be written to de-assert the reset.
[4]	**WSLOPT**	**Write System Latency Optimization** Controls the insertion of a pipeline stage on the AC signals from the DFI interface to the PHY to cater for a negative write system latency (WSL) value (only -1 possible). 0x0 = A pipeline stage is inserted only if WL2 training results in a WSL of -1 for any rank. 0x1 = Inserts a pipeline stage.

[3]	**Reserved**	Return zeroes on reads.
[2]	**WLSTEP**	**Write Leveling Step** Specifies the number of delay step-size increments during each step of write leveling. Valid values are: 0 = computed to be 1/2 of the associated lane's DXnGSR0.WLPRD value. 1 = 1 step size.
[1]	**WLMODE**	**Write Leveling Software Mode** Indicates, if set, that the PUB is in software write leveling mode in which software executes single steps of DQS pulsing by writing '1' to PIR.WL. The write leveling DQ status from the DRAM is captured in DXnGSR0.WLDQ.
[0]	**PDDISDX**	**Power Down Disabled Byte** Indicates, if set, that the PLL and I/Os of a disabled byte should be powered down.

PGSR0 Register (PGSR0)

Register	Offset	R/W	Description	Reset Value
PGSR0	DDRPHY_BA+0x010	R	PHY General Status Registers 0	0x0000_0000

31	30	29	28	27	26	25	24
APLOCK	Reserved	PLDONE_CHN		WEERR	REERR	WDERR	RDERR
23	22	21	20	19	18	17	16
WLAERR	QSGERR	WLERR	ZCERR	Reserved			
15	14	13	12	11	10	9	8
Reserved				WEDONE	REDONE	WDDONE	RDDONE
7	6	5	4	3	2	1	0
WLADONE	QSGDONE	WLDONE	DIDONE	ZCDONE	DCDONE	PLDONE	IDONE

Bits	Description	
[31]	**APLOCK**	**AC PLL Lock** Indicates, if set, that AC PLL has locked. This is a direct status of the AC PLL lock pin.
[30]	**Reserved**	Return zeroes on reads.
[29:28]	**PLDONE_CHN**	**PLL Lock Done per Channel** Indicates PLL locking has completed for each underlying channel. Bit 28 represents channel 0 while bit 29 represents channel 1.
[27]	**WEERR**	**Write Eye Training Error** Indicates, if set, that there is an error in write eye training.
[26]	**REERR**	**Read Data Eye Training Error** Indicates, if set, that there is an error in read eye training.
[25]	**WDERR**	**Write Data Bit Deskew Error** Indicates, if set, that there is an error in write bit deskew.
[24]	**RDERR**	**Read Data Bit Deskew Error** Indicates, if set, that there is an error in read bit deskew.
[23]	**WLAERR**	**Write Data Leveling Adjustment Error** Indicates, if set, that there is an error in write leveling adjustment.
[22]	**QSGERR**	**Read DQS Gate Training Error** Indicates, if set, that there is an error in DQS gate training.
[21]	**WLERR**	**Write Leveling Error** Indicates, if set, that there is an error in write leveling.
[20]	**ZCERR**	**Impedance Calibration Error** Indicates, if set, that there is an error in impedance calibration.
[19:12]	**Reserved**	Returns zeroes on reads.
[11]	**WEDONE**	**Write Data Eye Training Done** Indicates, if set, that write eye training has completed.

[10]	**REDONE**	**Read Data Eye Training Done** Indicates, if set, that read eye training has completed.
[9]	**WDDONE**	**Write Data Bit Deskew Done** Indicates, if set, that write bit deskew has completed.
[8]	**RDDONE**	**Read Data Bit Deskew Done** Indicates, if set, that read bit deskew has completed.
[7]	**WLADONE**	**Write Leveling Adjustment Done** Indicates, if set, that write leveling adjustment has completed.
[6]	**QSGDONE**	**Read DQS Gate Training Done** Indicates, if set, that DQS gate training has completed.
[5]	**WLDONE**	**Write Leveling Done** Indicates, if set, that write leveling has completed.
[4]	**DIDONE**	**DRAM Initialization Done** Indicates, if set, that DRAM initialization has completed.
[3]	**ZCDONE**	Impedance Calibration Done: Indicates, if set, that impedance calibration has completed.
[2]	**DCDONE**	**Digital Delay Line Calibration Done** Indicates, if set, that DDL calibration has completed.
[1]	**PLDONE**	**PLL Lock Done** Indicates, if set, that PLL locking has completed.
[0]	**IDONE**	**Initialization Done** Indicates, if set, that the DDR system initialization has completed. This bit is set after all the selected initialization routines in PIR register have completed.

PGSR1 Register (PGSR1)

Register	Offset	R/W	Description	Reset Value
PGSR1	DDRPHY_BA+0x014	R	PHY General Status Registers 1	0x0000_0000

31	30	29	28	27	26	25	24
PARERR	VTSTOP	Reserved					DLTCODE
23	22	21	20	19	18	17	16
DLTCODE							
15	14	13	12	11	10	9	8
DLTCODE							
7	6	5	4	3	2	1	0
DLTCODE							DLTDONE

Bits	Description	
[31]	**PARERR**	**RDIMM Parity Error** Indicates, if set, that there was a parity error (i.e. err_out_n was sampled low) during one of the transactions to the RDIMM buffer chip. This bit remains asserted until cleared by the PIR.CLRSR.
[30]	**VTSTOP**	**VT Stop** Indicates, if set, that the VT calculation logic has stopped computing the next values for the VT compensated delay line values. After assertion of the PGCR.INHVT, the VTSTOP bit should be read to ensure all VT compensation logic has stopped computations before writing to the delay line registers.
[29:25]	**Reserved**	Returns zeroes on reads.
[24:1]	**DLTCODE**	**Delay Line Test Code** Returns the code measured by the PHY control block that corresponds to the period of the AC delay line digital test output.
[0]	**DLTDONE**	**Delay Line Test Done** Indicates, if set, that the PHY control block has finished doing period measurement of the AC delay line digital test output.

PLLCR Register (PLLCR)

Register	Offset	R/W	Description	Reset Value
PLLCR	DDRPHY_BA+0x018	R/W	PLL Control Register	0x0001_c000

31	30	29	28	27	26	25	24
BYP	PLLRST	PLLPD	Reserved				
23	22	21	20	19	18	17	16
Reserved				FRQSEL		QPMODE	CPPC
15	14	13	12	11	10	9	8
CPPC			CPIC		GSHIFT	ATOEN	
7	6	5	4	3	2	1	0
ATOEN		ATC				DTC	

Bits	Description	
[31]	BYP	**PLL Bypass** Bypasses the PLL, if set, to 1.
[30]	PLLRST	**PLL Rest** Resets the PLLs by driving the PLL reset pin. This bit is not self-clearing and a '0' must be written to de-assert the reset.
[29]	PLLPD	**PLL Power Down** Puts the PLLs in Power-down mode by driving the PLL power down pin. This bit is not self-clearing and a '0' must be written to de-assert the power-down.
[28:20]	Reserved	Return zeroes on reads
[19:18]	FRQSEL	**PLL Frequency Select** 00 = PLL reference clock (ctl_clk/REF_CLK) ranges from 250 MHz to 400 MHz. 01 = PLL reference clock (ctl_clk/REF_CLK) ranges from 166 MHz to 300 MHz. 10 = Reserved. 11 = Reserved.
[17]	QPMODE	**PLL Quadrature Phase Mode** Enables, if set, the quadrature phase clock outputs. This mode is not used in this version of the PHY.
[16:13]	CPPC	**Charge Pump Proportional Current Control**
[12:11]	CPIC	**Charge Pump Integrating Current Control**
[10]	GSHIFT	**Gear Shift** Enables, if set, rapid locking mode.

[9:6]	**ATOEN**	**Analog Test Enable** Selects the analog test signal that is driven on the analog test output pin. Otherwise the analog test output is tri-stated. This allows analog test output pins from multiple PLLs to be connected together. Valid values are: 0000 = All PLL analog test signals are tri-stated. 0001 = AC PLL analog test signal is driven out. 0010 = DATX8 0 PLL analog test signal is driven out. 0011 = DATX8 1 PLL analog test signal is driven out. 0100 = DATX8 2 PLL analog test signal is driven out. 0101 = DATX8 3 PLL analog test signal is driven out. 0110 = DATX8 4 PLL analog test signal is driven out. 0111 = DATX8 5 PLL analog test signal is driven out. 1000 = DATX8 6 PLL analog test signal is driven out. 1001 = DATX8 7 PLL analog test signal is driven out. 1010 = DATX8 8 PLL analog test signal is driven out. 1011 – 1111 = Reserved.
[5:2]	**ATC**	**Analog Test Control** Selects various PLL analog test signals to be brought out via PLL analog test output pin (pll_ato). Valid values are: 0000 = Reserved. 0001 = vdd_ckin. 0010 = vrfbf. 0011 = vdd_cko. 0100 = vp_cp. 0101 = vpfil(vp). 0110 = Reserved. 0111 = gd. 1000 = vcntrl_atb. 1001 = vref_atb. 1010 = vpsf_atb. 1011 – 1111 = Reserved.
[1:0]	**DTC**	**Digital Test Control** Selects various PLL digital test signals and other test mode signals to be brought out via bit [1] of the PLL digital test output (pll_dto[1]). Valid values are: 00 = ‘0’ (Test output is disabled). 01 = PLL x1 clock (X1). 10 = PLL reference (input) clock (REF_CLK). 11 = PLL feedback clock (FB_X1).

PTR0 Register (PTR0)

Register	Offset	R/W	Description	Reset Value
PTR0	DDRPHY_BA+0x01c	R/W	PHY Timing Registers 0	0x42c2_1590

31	30	29	28	27	26	25	24
tPLLPD							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
tPLLPD			tPLLGS				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
tPLLGS							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
tPLLGS		tPHYRST					

Bits	Description	
[31:21]	**tPLLPD**	**PLL Power-Down Time** Number of configuration or APB clock cycles that the PLL must remain in power-down mode, i.e. number of clock cycles from when PLL power-down pin is asserted to when PLL power-down pin is de-asserted. This must correspond to a value that is equal to or more than 1us. Default value corresponds to 1us.
[20:6]	**tPLLGS**	**PLL Gear Shift Time** Number of configuration or APB clock cycles from when the PLL reset pin is de-asserted to when the PLL gear shift pin is de-asserted. This must correspond to a value that is equal to or more than 4us. Default value corresponds to 4us.
[5:0]	**tPHYRST**	**PHY Reset Time** Number of configuration or APB clock cycles that the PHY reset must remain asserted after PHY calibration is done before the reset to the PHY is de-asserted. This is used to extend the reset to the PHY so that the reset is asserted for some clock cycles after the clocks are stable. Valid values are from 1 to 63 (the value must be non-zero).

PTR1 Register (PTR1)

Register	Offset	R/W	Description	Reset Value
PTR1	DDRPHY_BA+0x020	R/W	PHY Timing Registers 1	0xd056_12c0

31	30	29	28	27	26	25	24
tPLLLOCK							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
tPLLLOCK							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			tPLLRST				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
tPLLRST							

Bits	Description	
[31:16]	**tPLLLOCK**	**PLL Lock Time** Number of configuration or APB clock cycles for the PLL to stabilize and lock, i.e. number of clock cycles from when the PLL reset pin is de-asserted to when the PLL has lock and is ready for use. This must correspond to a value that is equal to or more than 100us. Default value corresponds to 100us.
[15:13]	**Reserved**	Returns zeroes on reads.
[12:0]	**tPLLRST**	**PLL Reset Time** Number of configuration or APB clock cycles that the PLL must remain in reset mode, i.e. number of clock cycles from when PLL power-down pin is de-asserted and PLL reset pin is asserted to when PLL reset pin is de-asserted. The setting must correspond to a value that is equal to, or greater than 3us.

PTR2 Register (PTR2)

Register	Offset	R/W	Description	Reset Value
PTR2	DDRPHY_BA+0x024	R/W	PHY Timing Registers 2	0x0008_3def

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				tWLDLYS			
15	14	13	12	11	10	9	8
tWLDLYS	tCALH					tCALS	
7	6	5	4	3	2	1	0
tCALS			tCALON				

Bits	Description	
[31:20]	**Reserved**	Return zeroes on reads.
[19:15]	**t_{WLDLYS}**	**Write Leveling Delay Settling Time** Number of controller clock cycles from when a new value of the write leveling delay is applied to the LCDL to when to DQS high is driven high. This allows the delay to settle.
[14:10]	**t_{CALH}**	**Calibration Hold Time** Number of controller clock cycles from when the clock was disabled (cal_clk_en deasserted) to when calibration is enable (cal_en asserted).
[9:5]	**t_{CALS}**	**Calibration Setup Time** Number of controller clock cycles from when calibration is enabled (cal_en asserted) to when the calibration clock is asserted again (cal_clk_en asserted).
[4:0]	**t_{CALON}**	**Calibration On Time** Number of clock cycles that the calibration clock is enabled (cal_clk_en asserted).

PTR3 Register (PTR3)

Register	Offset	R/W	Description	Reset Value
PTR3	DDRPHY_BA+0x028	R/W	PHY Timing Registers 3	0x1808_2356

31	30	29	28	27	26	25	24
Reserved			tDINIT1				
23	22	21	20	19	18	17	16
tDINIT1				tDINIT0			
15	14	13	12	11	10	9	8
tDINIT0							
7	6	5	4	3	2	1	0
tDINIT0							

Bits	Description	
[31:29]	**Reserved**	Return zeroes on reads.
[28:20]	**t_{DINIT1}**	**DRAM Initialization Time 1** DRAM initialization time in DRAM clock cycles corresponding to the following: DDR3 = CKE high time to first command (t_{RFC} + 10 ns or 5 tCK, whichever is bigger) DDR2 = CKE high time to first command (400 ns). Default value corresponds to DDR3 t_{RFC} of 360ns at 1066 MHz.
[19:0]	**t_{DINIT0}**	**DRAM Initialization Time 0** DRAM initialization time in DRAM clock cycles corresponding to the following: DDR3 = CKE low time with power and clock stable (500 us) DDR2 = CKE low time with power and clock stable (200 us) Default value corresponds to DDR3 500 us at 1066 MHz. During Verilog simulations, it is recommended that this value is changed to a much smaller value in order to avoid long simulation times. However, this may cause a memory model error, due to a violation of the CKE setup sequence. This violation is expected if this value is not programmed to the required SDRAM CKE low time, but memory models should be able to tolerate this violation without malfunction of the model.

PTR4 Register (PTR4)

Register	Offset	R/W	Description	Reset Value
PTR4	DDRPHY_BA+0x02c	R/W	PHY Timing Registers 4	0x0aaf_4156

31	30	29	28	27	26	25	24
Reserved				tDINIT3			
23	22	21	20	19	18	17	16
tDINIT3						tDINIT2	
15	14	13	12	11	10	9	8
tDINIT2							
7	6	5	4	3	2	1	0
tDINIT2							

Bits	Description	
[31:28]	**Reserved**	Return zeroes on reads.
[27:18]	**tDINIT3**	**DRAM Initialization Time 3** DRAM initialization time in DRAM clock cycles corresponding to the following: DDR3 = Time from ZQ initialization command to first command (1 us) Default value corresponds to the DDR3 640ns at 1066 MHz.
[17:0]	**tDINIT2**	**DRAM Initialization Time 2** DRAM initialization time in DRAM clock cycles corresponding to the following: DDR3 = Reset low time (200 us on power-up or 100 ns after power-up) Default value corresponds to DDR3 200 us at 1066 MHz.

ACMDLR Register (ACMDLR)

Register	Offset	R/W	Description	Reset Value
ACMDLR	DDRPHY_BA+0x030	R/W	AC Master Delay Line Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
MDLD							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TPRD							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
IPRD							

Bits	Description	
[31:24]	**Reserved**	Return zeroes on reads.
[23:16]	**MDLD**	**MDL Delay** Delay select for the LCDL for the Master Delay Line.
[15:8]	**TPRD**	**Target Period** Target period measured by the master delay line calibration for VT drift compensation. This is the current measured value of the period and is continuously updated if the MDL is enabled to do so.
[7:0]	**IPRD**	**Initial Period** Initial period measured by the master delay line calibration for VT drift compensation. This value is used as the denominator when calculating the ratios of updates during VT compensation.

ACBDLR Register (ACBDLR)

Register	Offset	R/W	Description	Reset Value
ACBDLR	DDRPHY_BA+0x034	R/W	AC Bit Delay Line Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ACBD						CK2BD	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CK2BD				CK1BD			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CK1BD		CK0BD					

Bits	Description	
[31:24]	**Reserved**	Return zeroes on reads.
[23:18]	**ACBD**	**Address/Command Bit Delay** Delay select for the BDLs on address and command signals.
[17:12]	**CK2BD**	**CK2 Bit Delay** Delay select for the BDL on CK2.
[11:6]	**CK1BD**	**CK1 Bit Delay** Delay select for the BDL on CK1.
[5:0]	**CK0BD**	**CK0 Bit Delay** Delay select for the BDL on CK0.

ACIOCR Register (ACIOCR)

Register	Offset	R/W	Description	Reset Value
ACIOCR	DDRPHY_BA+0x038	R/W	AC I/O Configuration Register	0x30c0_0812

31	30	29	28	27	26	25	24
ACSR		RSTIOM	RSTPDR	RSTPDD	RSTODT	RANKPDR	
23	22	21	20	19	18	17	16
RANKPDR		CSPDD				RANKODT	
15	14	13	12	11	10	9	8
RANKODT		CKPDR			CKPDD		
7	6	5	4	3	2	1	0
CKODT			ACPDR	ACPDD	ACODT	ACOE	ACIOM

Bits	Description	
[31:30]	**ACSR**	**Address/Command Slew Rate for D3F I/O Only** Selects slew rate of the I/O for all address and command pins.
[29]	**RSTIOM**	**SDRAM Reset I/O Mode** Selects SSTL mode (when set to 0) or CMOS mode (when set to 1) of the I/O for SDRAM Reset.
[28]	**RSTPDR**	**SDRAM Reset Power Down Receiver** Powers down, when set, the input receiver on the I/O for SDRAM RST# pin.
[27]	**RSTPDD**	**SDRAM Reset Power Down Driver** Powers down, when set, the output driver on the I/O for SDRAM RST# pin. **Note:** Asserting PDD puts the I/O driver cell into a lower power, lower speed mode of operation. However, it will still drive if its OE is asserted. ODT will be disabled (if used). Asserting PDD does not prevent the I/O from driving.
[26]	**RSTODT**	**SDRAM Reset On-Die Termination** Enables, when set, the on-die termination on the I/O for SDRAM RST# pin.
[25:22]	**RANKPDR**	**Rank Power Down Receiver** Powers down, when set, the input receiver on the I/O CKE[3:0], ODT[3:0], and CS#[3:0] pins. RANKPDR[0] controls the power down for CKE[0], ODT[0], and CS#[0], RANKPDR[1] controls the power down for CKE[1], ODT[1], and CS#[1], and so on.
[21:18]	**CSPDD**	**CS# Power Down Driver** Powers down, when set, the output driver on the I/O for CS#[3:0] pins. CSPDD[0] controls the power down for CS#[0], CSPDD[1] controls the power down for CS#[1], and so on. CKE and ODT driver power down is controlled by DSGCR register. **Note:** Asserting PDD puts the I/O driver cell into a lower power, lower speed mode of operation. However, it will still drive if its OE is asserted. ODT will be disabled (if used). Asserting PDD does not prevent the I/O from driving.
[17:14]	**RANKODT**	**Rank On-Die Termination** Enables, when set, the on-die termination on the I/O for CKE[3:0], ODT[3:0], and CS#[3:0] pins. RANKODT[0] controls the on-die termination for CKE[0], ODT[0], and CS#[0], RANKODT[1] controls the on-die termination for CKE[1], ODT[1], and CS#[1], and so on.

[13:11]	**CKPDR**	**CK Power Down Receiver** Powers down, when set, the input receiver on the I/O for CK[0], CK[1], and CK[2] pins, respectively.
[10:8]	**CKPDD**	**CK Power Down Driver** Powers down, when set, the output driver on the I/O for CK[0], CK[1], and CK[2] pins, respectively. **Note:** Asserting PDD puts the I/O driver cell into a lower power, lower speed mode of operation. However, it will still drive if its OE is asserted. ODT will be disabled (if used). Asserting PDD does not prevent the I/O from driving.
[7:5]	**CKODT**	**CK On-Die Termination** Enables, when set, the on-die termination on the I/O for CK[0], CK[1], and CK[2] pins, respectively.
[4]	**ACPDR**	**AC Power Down Receiver** Powers down, when set, the input receiver on the I/O for RAS#, CAS#, WE#, BA[2:0], and A[15:0] pins.
[3]	**ACPDD**	**AC Power Down Driver** Powers down, when set, the output driver on the I/O for RAS#, CAS#, WE#, BA[2:0], and A[15:0] pins. **Note:** Asserting PDD puts the I/O driver cell into a lower power, lower speed mode of operation. However, it will still drive if its OE is asserted. ODT will be disabled (if used). Asserting PDD does not prevent the I/O from driving.
[2]	**ACODT**	**Address/Command On-Die Termination** Enables, when set, the on-die termination on the I/O for RAS#, CAS#, WE#, BA[2:0], and A[15:0] pins.
[1]	**ACOE**	**Address/Command Output Enable** Enables, when set, the output driver on the I/O for all address and command pins.
[0]	**ACIOM**	**Address/Command I/O Mode** Selects SSTL mode (when set to 0) or CMOS mode (when set to 1) of the I/O for all address and command pins. This bit connects to bit [0] of the IOM pin on the D3F I/Os, and for other I/O libraries, it connects to the IOM pin of the I/O.

DXCCR Register (DXCCR)

Register	Offset	R/W	Description	Reset Value
DXCCR	DDRPHY_BA+0x03c	R/W	DATX8 Common Configuration Register	0x4418_1884

31	30	29	28	27	26	25	24
DDPDRCDO				DDPDDCDO			
23	22	21	20	19	18	17	16
DYNDXPDR	DYNDXPDD	UDQIOM	UDQPDR	UDQPDD	UDQODT	MSBUDQ	
15	14	13	12	11	10	9	8
MSBUDQ	DXSR		DQSNRES				DQSRES
7	6	5	4	3	2	1	0
DQSRES			DXPDR	DXPDD	MDLEN	DXIOM	DXODT

Bits	Description	
[31:28]	**DDPDRCDO**	**Dynamic Data Power Down Receiver Count Down Offset** Offset applied in calculating window of time where receiver is powered up
[27:24]	**DDPDDCDO**	**Dynamic Data Power Down Driver Count Down Offset** Offset applied in calculating window of time where driver is powered up
[23]	**DYNDXPDR**	**Data Power Down Receiver** Dynamically powers down, when set, the input receiver on I/O for the DQ pins of the active DATX8 macros. Applies only when DXPDR and DXnGCR.DXPDR are not set to 1. Receiver is powered-up on a DFI READ command and powered-down (trddata_en + fixed_read_latency + n) HDR cycles after the last DFI READ command. Note that n is defined by the register bit field DXCCR[31:28] (DDPDRCDO).
[22]	**DYNDXPDD**	**Dynamic Data Power Down Driver** Dynamically powers down, when set, the output driver on I/O for the DQ pins of the active DATX8 macros. Applies only when DXPDD and DXnGCR.DXPDD are not set to 1. Driver is powered-up on a DFI WRITE command and powered-down (twrlat + WL/2 + n) HDR cycles after the last DFI WRITE command. Note that n is defined by the register bit field DXCCR[27:24] (DDPDDCDO). **Note:** Asserting PDD puts the I/O driver cell into a lower power, lower speed mode of operation. However, it will still drive if its OE is asserted. ODT will be disabled (if used). Asserting PDD does not prevent the I/O from driving.
[21]	**UDQIOM**	**Unused DQ I/O Mode** Selects SSTL mode (when set to 0) or CMOS mode (when set to 1) of the I/O for unused DQ pins.
[20]	**UDQPDR**	**Unused DQ Power Down Receiver** Powers down, when set, the input receiver on the I/O for unused DQ pins.
[19]	**UDQPDD**	**Unused DQ Power Down Driver** Powers down, when set, the output driver on the I/O for unused DQ pins. **Note:** Asserting PDD puts the I/O driver cell into a lower power, lower speed mode of operation. However, it will still drive if its OE is asserted. ODT will be disabled (if used). Asserting PDD does not prevent the I/O from driving.
[18]	**UDQODT**	**Unused DQ On-Die Termination** Enables, when set, the on-die termination on the I/O for unused DQ pins.

[17:15]	**MSBUDQ**	**Most Significant Byte Unused DQs** Specifies the number of DQ bits that are not used in the most significant byte. The used (valid) bits for this byte are [8-MSBDQ-1:0]. To disable the whole byte, use the DXnGCR.DXEN register.
[14:13]	**DXSR**	**Data Slew Rate for D3F I/O Only** Selects slew rate of the I/O for DQ, DM, and DQS/DQS# pins of all DATX8 macros.
[12:9]	**DQSNRES**	**DQS# Resistor** Selects the on-die pull-up/pull-down resistor for DQS# pins. Same encoding as DQSRES. Refer PHY databook for pull-down/pull-up resistor values (RA_SEL/RB_SEL) for DQS/DQS_b.
[8:5]	**DQSRES**	**DQS Resistor** Selects the on-die pull-down/pull-up resistor for DQS pins. DQSRES[3] selects pull-down (when set to 0) or pull-up (when set to 1). DQSRES[2:0] selects the resistor value. Refer PHY databook for pull-down/pull-up resistor values (RA_SEL/RB_SEL) for DQS/DQS_b.
[4]	**DXPDR**	**Data Power Down Receiver** Powers down, when set, the input receiver on I/O for DQ, DM, and DQS/DQS# pins of all DATX8 macros. This bit is ORed with the PDR configuration bit of the individual DATX8.
[3]	**DXPDD**	**Data Power Down Driver** Powers down, when set, the output driver on I/O for DQ, DM, and DQS/DQS# pins of all DATX8 macros. This bit is ORed with the PDD configuration bit of the individual DATX8. **Note:** Asserting PDD puts the I/O driver cell into a lower power, lower speed mode of operation. However, it will still drive if its OE is asserted. ODT will be disabled (if used). Asserting PDD does not prevent the I/O from driving.
[2]	**MDLEN**	**Master Delay Line Enable** Enables, if set, all DATX8 master delay line calibration to perform subsequent period measurements following the initial period measurements that are performed after reset or on when calibration is manually triggered. These additional measurements are accumulated and filtered as long as this bit remains high. This bit is ANDed with the MDLEN bit in the individual DATX8.
[1]	**DXIOM**	**Data I/O Mode** Selects SSTL mode (when set to 0) or CMOS mode (when set to 1) of the I/O for DQ, DM, and DQS/DQS# pins of all DATX8 macros. This bit is ORed with the IOM configuration bit of the individual DATX8.
[0]	**DXODT**	**Data On-Die Termination** Enables, when set, the on-die termination on the I/O for DQ, DM, and DQS/DQS# pins of all DATX8 macros. This bit is ORed with the ODT configuration bit of the individual DATX8 (“DATX8 General Configuration Register (DXnGCR)”)

DSGCR Register (DSGCR)

Register	Offset	R/W	Description	Reset Value
DSGCR	DDRPHY_BA+0x040	R/W	DDR System General Configuration Register	0xf004_649f

31	30	29	28	27	26	25	24
CKEOE	RSTOE	ODTOE	CKOE	ODTPDD			
23	22	21	20	19	18	17	16
CKEPDD				SDRMODE	RRMODE	ATOAE	DTOOE
15	14	13	12	11	10	9	8
DTOIOM	DTOPDR	DTOPDD	DTOODT	PUAD			
7	6	5	4	3	2	1	0
BRRMODE	DQSGX	CUAEN	LPPLLPD	LPIOPD	ZUEN	BDISEN	PUREN

Bits	Description	
[31]	**CKEOE**	**SDRAM CKE Output Enable** Enables, when set, the output driver on the I/O for SDRAM CKE pins.
[30]	**RSTOE**	**SDRAM Reset Output Enable** Enables, when set, the output driver on the I/O for SDRAM RST# pin.
[29]	**ODTOE**	**SDRAM ODT Output Enable** Enables, when set, the output driver on the I/O for SDRAM ODT pins.
[28]	**CKOE**	**SDRAM CK Output Enable** Enables, when set, the output driver on the I/O for SDRAM CK/CK# pins.
[27:24]	**ODTPDD**	**ODT Power Down Driver** Powers down, when set, the output driver on the I/O for ODT[3:0] pins. ODTPDD[0] controls the power down for ODT[0], ODTPDD[1] controls the power down for ODT[1], and so on. **Note:** Asserting PDD puts the I/O driver cell into a lower power, lower speed mode of operation. However, it will still drive if its OE is asserted. ODT will be disabled (if used). Asserting PDD does not prevent the I/O from driving.
[23:20]	**CKEPDD**	**CKE Power Down Driver** Powers down, when set, the output driver on the I/O for CKE[3:0] pins. CKEPDD[0] controls the power down for CKE[0], CKEPDD[1] controls the power down for CKE[1], and so on. **Note:** Asserting PDD puts the I/O driver cell into a lower power, lower speed mode of operation. However, it will still drive if its OE is asserted. ODT will be disabled (if used). Asserting PDD does not prevent the I/O from driving.
[19]	**SDRMODE**	**Single Data Rate Mode** Indicates, if set, that the external controller is configured to run in single data rate (SDR) mode. Otherwise if not set the controller is running in half data rate (HDR) mode. This bit not supported in the current version of the PUB.
[18]	**RRMODE**	**Rise-to-Rise Mode** Indicates, if set, that the PHY mission mode is configured to run in rise-to-rise mode. Otherwise if not set the PHY mission mode is running in rise-to- fall mode.
[17]	**ATOAE**	**ATO Analog Test Enable** Enables, if set, the analog test output (ATO) I/O.

[16]	**DTOOE**	**DTO Output Enable** Enables, when set, the output driver on the I/O for DTO pins.
[15]	**DTOIOM**	**DTO I/O Mode** Selects SSTL mode (when set to 0) or CMOS mode (when set to 1) of the I/O for DTO pins.
[14]	**DTOPDR**	**DTO Power Down Receiver** Powers down, when set, the input receiver on the I/O for DTO pins.
[13]	**DTOPDD**	**DTO Power Down Driver** Powers down, when set, the output driver on the I/O for DTO pins. **Note:** Asserting PDD puts the I/O driver cell into a lower power, lower speed mode of operation. However, it will still drive if its OE is asserted. ODT will be disabled (if used). Asserting PDD does not prevent the I/O from driving.
[12]	**DTOODT**	**DTO On-Die Termination** Enables, when set, the on-die termination on the I/O for DTO pins.
[11:8]	**PUAD**	**PHY Update Acknowledge Delay** Specifies the number of clock cycles that the indication for the completion of PHY update from the PHY to the controller should be delayed. This essentially delays, by this many clock cycles, the de-assertion of dfi_ctrlup_ack and dfi_phyupd_req signals relative to the time when the delay lines or I/Os are updated.
[7]	**BRRMODE**	**Bypass Rise-to-Rise Mode** Indicates, if set, that the PHY bypass mode is configured to run in rise-to-rise mode. Otherwise if not set the PHY bypass mode is running in rise-to-fall mode.
[6]	**DQSGX**	**DQS Gate Extension** Specifies, if set, that the DQS gating must be extended by two DRAM clock cycles and then re-centered, i.e. one clock cycle extension on either side.
[5]	**CUAEN**	**Controller Update Acknowledge Enable** Specifies, if set, that the PHY should issue controller update acknowledge when the DFI controller update request is asserted. By default, the PHY does not acknowledge controller initiated update requests but simply does an update whenever there is a controller update request. This speeds up the update.
[4]	**LPPLLPD**	**Low Power PLL Power Down** Specifies, if set, that the PHY should respond to the DFI low power opportunity request and power down the PLL of the byte if the wakeup time request satisfies the PLL lock time.
[3]	**LPIOPD**	**Low Power I/O Power Down** Specifies, if set, that the PHY should respond to the DFI low power opportunity request and power down the I/Os of the byte.
[2]	**ZUEN**	**Impedance Update Enable** Specifies, if set, that in addition to DDL VT update, the PHY could also perform impedance calibration (update). Refer to the “Impedance Control Register 0-1 (ZQnCR0-1)” bit fields DFICU0, DFICU1 and DFICCU bits to control if an impedance calibration is performed (update) with a DFI controller update request. Refer to the “Impedance Control Register 0-1 (ZQnCR0-1)” bit fields DFIPU0 and DFIPU1 bits to control if an impedance calibration is performed (update) with a DFI PHY update request.
[1]	**BDISEN**	**Byte Disable Enable** Specifies, if set, that the PHY should respond to DFI byte disable request. Otherwise the byte disable from the DFI is ignored in which case bytes can only be disabled using the DXnGCR register.
[0]	**PUREN**	**PHY Update Request Enable** Specifies if set, that the PHY should issue PHY- initiated update request when there is DDL VT drift.

DCR Register (DCR)

Register	Offset	R/W	Description	Reset Value
DCR	DDRPHY_BA+0x044	R/W	DRAM Configuration Register	0x0000_040b

31	30	29	28	27	26	25	24
Reserved		UDIMM	DDR2T	NOSRA	Reserved		
23	22	21	20	19	18	17	16
Reserved						BYTEMASK	
15	14	13	12	11	10	9	8
BYTEMASK						Reserved	
7	6	5	4	3	2	1	0
MPRDQ	PDQ			DDR8BNK	DDRMD		

Bits	Description	
[31:30]	**Reserved**	Return zeroes on reads.
[29]	**UDIMM**	**Un-buffered DIMM Address Mirroring** Indicates, if set, that there is address mirroring on the second rank of an un-buffered DIMM (the rank connected to CS#[1]). In this case, the PUB re-scrambles the bank and address when sending mode register commands to the second rank. This only applies to PUB internal SDRAM transactions. Transactions generated by the controller must make its own adjustments when using an un-buffered DIMM. DCR[NOSRA] must be set if address mirroring is enabled.
[28]	**DDR2T**	**DDR 2T Timing** Indicates, if set, that 2T timing should be used by PUB internally generated SDRAM transactions.
[27]	**NOSRA**	**No Simultaneous Rank Access** Specifies, if set, that simultaneous rank access on the same clock cycle is not allowed. This means that multiple chip select signals should not be asserted at the same time. This may be required on some DIMM systems.
[26:18]	**Reserved**	Return zeroes on reads.
[17:10]	**BYTEMASK**	**Byte Mask** Mask applied to all beats of read data on all byte lanes during read DQS gate training. This allows training to be conducted based on selected bit(s) from the byte lanes. Valid values for each bit are: 0 = Compare for that bit Disabled. 1 = Compare for that bit Enabled. Note that this mask applies in DDR3 MPR operation mode as well and must be in keeping with the PDQ field setting.
[9:8]	**Reserved**	These bits are reserved but are writable and readable
[7]	**MPRDQ**	**Multi-Purpose Register DQ for DDR3 Only** Specifies the value that is driven on non-primary DQ pins during MPR reads. Valid values are: 0 = Primary DQ drives out the data from MPR (0-1-0-1); non-primary DQs drive ‘0’. 1 = Primary DQ and non-primary DQs all drive the same data from MPR (0-1-0-1).

[6:4]	**PDQ**	**Primary DQ for DDR3 Only** Specifies the DQ pin in a byte that is designated as a primary pin for Multi-Purpose Register (MPR) reads. Valid values are 0 to 7 for DQ[0] to DQ[7], respectively.
[3]	**DDR8BNK**	**DDR 8-Bank** Indicates, if set, that the SDRAM used has 8 banks. tRPA = tRP+1 and tFAW are used for 8-bank DRAMs, otherwise tRPA = tRP and no tFAW is used. Note that a setting of 1 for DRAMs that have fewer than 8 banks results in correct functionality, but less tight DRAM command spacing for the parameters.
[2:0]	**DDRMD**	**DDR Mode** SDRAM DDR mode. Valid values are: 000 = Reserved. 001 = Reserved. 010 = DDR2. 011 = DDR3. 100 – 111 = Reserved.

DTPR0 Register (DTPR0)

Register	Offset	R/W	Description	Reset Value
DTPR0	DDRPHY_BA+0x048	R/W	DRAM Timing Parameters Register 0	0xc9e4_ee88

31	30	29	28	27	26	25	24
tRC						tRRD	
23	22	21	20	19	18	17	16
tRRD		tRAS					
15	14	13	12	11	10	9	8
tRCD				tRP			
7	6	5	4	3	2	1	0
tWTR				tRTP			

Bits	Description	
[31:26]	t_{RC}	**Activate to activate command delay for the same bank** Valid values are 2 to 63.
[25:22]	t_{RRD}	**Activate to activate command delay for the different banks** Valid values are 1 to 15.
[21:16]	t_{RAS}	**Activate to precharge command delay** Valid values are 2 to 63.
[15:12]	t_{RCD}	**Activate to read or write delay** Minimum time from when an activate command is issued to when a read or write to the activated row can be issued. Valid values are 2 to 15.
[11:8]	t_{RP}	**Precharge command period** The minimum time between a precharge command and any other command. Note that the Controller automatically derives t_{RPA} for 8- bank DDR2 devices by adding 1 to t_{RP} Valid values are 2 to 15.
[7:4]	t_{WTR}	**Internal write to read command delay** Valid values are 1 to 15.
[3:0]	t_{RTP}	**Internal read to precharge command delay** Valid values are 2 to 15.

DTPR1 Register (DTPR1)

Register	Offset	R/W	Description	Reset Value
DTPR1	DDRPHY_BA+0x04c	R/W	DRAM Timing Parameters Register 1	0x228b_b4d2

31	30	29	28	27	26	25	24
tAONOFFD		tWLO				tWLMRD	
23	22	21	20	19	18	17	16
tWLMRD				tRFC			
15	14	13	12	11	10	9	8
tRFC					tFAW		
7	6	5	4	3	2	1	0
tFAW			tMOD			tMRD	

Bits	Description	
[31:30]	$t_{AONOFFD}$	**ODT turn-on turn-off delays for DDR2 only** Valid values are: 00 = 2/2.5. 01 = 3/3.5. 10 = 4/4.5. 11 = 5/5.5. Most DDR2 devices utilize a fixed value of 2/2.5. For non-standard SDRAMs, the user must ensure that the operational Write Latency is always greater than or equal to the ODT turn-on delay. For example, a DDR2 SDRAM with CAS latency set to 3 and CAS additive latency set to 0 has a Write Latency of 2. Thus 2/2.5 can be used, but not 3/3.5 or higher.
[29:26]	t_{WLO}	**Write leveling output delay** Number of clock cycles from when write leveling DQS is driven high by the control block to when the results from the SDRAM on DQ is sampled by the control block. This must include the SDRAM t_{WLO} timing parameter plus the round trip delay from control block to SDRAM back to control block.
[25:20]	t_{WLMRD}	**Minimum delay from when write leveling mode is programmed to the first** DQS/DQS# rising edge.
[19:11]	t_{RFC}	**Refresh-to-Refresh** Indicates the minimum time between two refresh commands or between a refresh and an active command. This is derived from the minimum refresh interval from the Datasheet, $t_{RFC(min)}$, divided by the clock cycle time. The default number of clock cycles is for the largest JEDEC $t_{RFC(min}$ parameter value supported.
[10:5]	t_{FAW}	**4-bank activate period** No more than 4-bank activate commands may be issued in a given t_{FAW} period. Only applies to 8-bank devices. Valid values are 2 to 63.

[4:2]	t_{MOD}	**Load mode update delay for DDR3 only** The minimum time between a load mode register command and a non-load mode register command. Valid values are: 000 = 12. 001 = 13. 010 = 14. 011 = 15. 100 = 16. 101 = 17. 110 – 111 = Reserved.
[1:0]	t_{MRD}	**Load mode cycle time** The minimum time between a load mode register command and any other command. For DDR3 this is the minimum time between two load mode register commands. Valid values for DDR2 are 2 to 3. For DDR3, the value used for t_{MRD} is 4 plus the value programmed in these bits, i.e. t_{MRD} value for DDR3 ranges from 4 to 7.

DTPR2 Register (DTPR2)

Register	Offset	R/W	Description	Reset Value
DTPR2	DDRPHY_BA+0x050	R/W	DRAM Timing Parameters Register 2	0x1003_6a00

31	30	29	28	27	26	25	24
tCCD	tRTW	tRTODT	tDLLK				
23	22	21	20	19	18	17	16
tDLLK					tCKE		
15	14	13	12	11	10	9	8
tCKE	tXP					tXS	
7	6	5	4	3	2	1	0
tXS							

Bits	Description	
[31]	t_{CCD}	**Read to read and write to write command delay** Valid values are: 0 = BL/2 for DDR2 and 4 for DDR3. 1 = BL/2 + 1 for DDR2 and 5 for DDR3.
[30]	t_{RTW}	**Read to Write command delay** Valid values are: 0 = standard bus turn around delay. 1 = add 1 clock to standard bus turn around delay. This parameter allows the user to increase the delay between issuing Write commands to the SDRAM when preceded by Read commands. This provides an option to increase bus turn-around margin for high frequency systems.
[29]	t_{RTODT}	**Read to ODT delay for DDR3 only** Specifies whether ODT can be enabled immediately after the read post-amble or one clock delay has to be added. Valid values are: 0 = ODT may be turned on immediately after read post-amble. 1 = ODT may not be turned on until one clock after the read post-amble. If t_{RTODT} is set to 1, then the read-to-write latency is increased by 1 if ODT is enabled.
[28:19]	t_{DLLK}	**DLL locking time** Valid values are 2 to 1023.
[18:15]	t_{CKE}	**CKE minimum pulse width** Also specifies the minimum time that the SDRAM must remain in power down or self refresh mode. For DDR3 this parameter must be set to the value of t_{CKESR} which is usually bigger than the value of t_{CKE}. Valid values are 2 to 15.
[14:10]	t_{XP}	**Power down exit delay** The minimum time between a power down exit command and any other command. This parameter must be set to the maximum of the various minimum power down exit delay parameters specified in the SDRAM Datasheet, i.e. max(t_{XP} t_{XARD}, t_{XARDS}) for DDR2 and max(t_{XP} t_{XPDLL}) for DDR3. Valid values are 2 to 31.

[9:0]	**t_{XS}**	**Self refresh exit delay** The minimum time between a self refresh exit command and any other command. This parameter must be set to the maximum of the various minimum self refresh exit delay parameters specified in the SDRAM Datasheet, i.e. max(t_{XSNR}, t_{XSRD}) for DDR2 and max(t_{XS}, t_{XSDLL}) for DDR3. Valid values are 2 to 1023.

MR0 Register (MR0)

Register	Offset	R/W	Description	Reset Value
MR0	DDRPHY_BA+0x054	R/W	Mode Register 0	0x0000_0a52

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			PD	WR			DR
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TM	CL_b321			BT	CL_b0	BL	

Bits	Description	
[31:16]	**Reserved**	Return zeroes on reads.
[15:13]	**Reserved**	Reserved.
[12]	**PD**	**Power-Down Control** Controls the exit time for power-down modes. Refer to the SDRAM Datasheet for details on power-down modes. Valid values are: 0 = Slow exit (DLL off). 1 = Fast exit (DLL on).
[11:9]	**WR**	**Write Recovery** This is the value of the write recovery. It is calculated by dividing the Datasheet write recovery time, t_{WR} (ns) by the Datasheet clock cycle time, t_{CK} (ns) and rounding up a non-integer value to the next integer. Valid values are: 000 = 16. 001 = 5. 010 = 6. 011 = 7. 100 = 8. 101 = 10. 110 = 12. 111 = 14. All other settings are reserved and should not be used. **Note:** t_{WR} (ns) is the time from the first SDRAM positive clock edge after the last data-in pair of a write command, to when a precharge of the same bank can be issued.
[8]	**DR**	**DLL Reset** Writing a ‘1’ to this bit will reset the SDRAM DLL. This bit is self- clearing, i.e. it returns back to ‘0’ after the DLL reset has been issued.
[7]	**TM**	**Operating Mode** Selects either normal operating mode (0) or test mode (1). Test mode is reserved for the manufacturer and should not be used.

[6:4]	**CL_b321**	**CAS Latency** The delay between when the SDRAM registers a read command to when data is available. Valid values are: 0001 = 12. 0010 = 5. 0011 = 13. 0100 = 6. 0101 = 14. 0110 = 7. 1000 = 8. 1010 = 9. 1100 = 10. 1110 = 11. All other settings are reserved and should not be used.
[3]	**BT**	**Burst Type** Indicates whether a burst is sequential (0) or interleaved (1).
[2]	**CL_b0**	**CAS Latency** The delay between when the SDRAM registers a read command to when data is available. Valid values are: 0001 = 12. 0010 = 5. 0011 = 13. 0100 = 6. 0101 = 14. 0110 = 7. 1000 = 8. 1010 = 9. 1100 = 10. 1110 = 11. All other settings are reserved and should not be used.
[1:0]	**BL**	**Burst Length** Determines the maximum number of column locations that can be accessed during a given read or write command. Valid values are: Valid values for DDR3 are: 00 = 8 (Fixed). 01 = 4 or 8 (On the fly). 10 = 4 (Fixed). 11 = Reserved.

MR1 Register (MR1)

Register	Offset	R/W	Description	Reset Value
MR1	DDRPHY_BA+0x058	R/W	Mode Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			QOFF	TDQS	Reserved	RTT_b2	Reserved
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
LEVEL	RTT_b1	DIC_b1	AL		RTT_b0	DIC_b0	DE

Bits	Description	
[31:13]	**Reserved**	Reserved.
[12]	**QOFF**	**Output Enable/Disable** When ‘0’, all outputs function normal; when ‘1’ all SDRAM outputs are disabled removing output buffer current. This feature is intended to be used for I_{DD} characterization of read current and should not be used in normal operation.
[11]	**TDQS**	**Termination Data Strobe** When enabled (‘1’) TDQS provides additional termination resistance outputs that may be useful in some system configurations. Refer to the SDRAM Datasheet for details.
[10]	**Reserved**	Reserved.
[9]	**RTT_b2**	**On Die Termination** Selects the effective resistance for SDRAM on die termination. Valid values are: 000 = ODT disabled. 001 = RZQ/4. 010 = RZQ/2. 011 = RZQ/6. 100 = RZQ/12. 101 = RZQ/8. All other settings are reserved and should not be used.
[8]	**Reserved**	Reserved.
[7]	**LEVEL**	**Write Leveling Enable** Enables write-leveling when set.

[6]	**RTT_b1**	**On Die Termination** Selects the effective resistance for SDRAM on die termination. Valid values are: 000 = ODT disabled. 001 = RZQ/4. 010 = RZQ/2. 011 = RZQ/6. 100 = RZQ/12. 101 = RZQ/8. All other settings are reserved and should not be used.
[5]	**DIC_b1**	**Output Driver Impedance Control** Controls the output drive strength. Valid values are: 00 = RZQ/6. 01 = RZQ7. 10 = Reserved. 11 = Reserved.
[4:3]	**AL**	**Posted CAS Additive Latency** Setting additive latency that allows read and write commands to be issued to the SDRAM earlier than normal (refer to the SDRAM Datasheet for details). Valid values are: 00 = 0 (AL disabled). 01 = CL - 1. 10 = CL - 2. 11 = Reserved.
[2]	**RTT_b0**	**On Die Termination** Selects the effective resistance for SDRAM on die termination. Valid values are: 000 = ODT disabled. 001 = RZQ/4. 010 = RZQ/2. 011 = RZQ/6. 100 = RZQ/12. 101 = RZQ/8. All other settings are reserved and should not be used.
[1]	**DIC_b0**	**Output Driver Impedance Control** Controls the output drive strength. Valid values are: 00 = RZQ/6. 01 = RZQ7. 10 = Reserved. 11 = Reserved.
[0]	**DE**	**DLL Enable/Disable** Enable (0) or disable (1) the DLL. DLL must be enabled for normal operation. **Note:** SDRAM DLL off mode is not supported

MR2 Register (MR2)

Register	Offset	R/W	Description	Reset Value
MR2	DDRPHY_BA+0x05c	R/W	Mode Register 2/Extended Mode Register 2	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved					RTTWR		Reserved
7	6	5	4	3	2	1	0
SRT	ASR	CWL			PASR		

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:11]	**Reserved**	Reserved.
[10:9]	**RTTWR**	**Dynamic ODT** Selects RTT for dynamic ODT. Valid values are: 00 = Dynamic ODT off. 01 = RZQ/4. 10 = RZQ/2. 11 = Reserved.
[8]	**Reserved**	Reserved.
[7]	**SRT**	**Self-Refresh Temperature Range** Selects either normal (‘0’) or extended (‘1’) operating temperature range during self-refresh.
[6]	**ASR**	**Auto Self-Refresh** When enabled (‘1’), SDRAM automatically provides self-refresh power management functions for all supported operating temperature values. Otherwise the SRT bit must be programmed to indicate the temperature range.
[5:3]	**CWL**	**CAS Write Latency** The delay between when the SDRAM registers a write command to when write data is available. Valid values are: 000 = 5 (t_{CK} > 2.5ns). 001 = 6 (2.5ns > t_{CK} > 1.875ns). 010 = 7 (1.875ns > t_{CK}> 1.5ns). 011 = 8 (1.5ns > t_{CK} > 1.25ns). 100 = 9 (1.25ns > tCK > 1.07ns). 101 = 10 (1.07ns > tCK > 0.935ns). 110 = 11 (0.935ns > tCK > 0.833ns). 111 = 12 (0.833ns > tCK > 0.75ns). All other settings are reserved and should not be used.

[2:0]	**PASR**	**Partial Array Self Refresh** Specifies that data located in areas of the array beyond the specified location will be lost if self refresh is entered. Valid settings for 4 banks are: 000 = Full Array. 001 = Half Array (BA[1:0] = 00 and 01). 010 = Quarter Array (BA[1:0] = 00). 011 = Not defined. 100 = 3/4 Array (BA[1:0] = 01, 10, and 11). 101 = Half Array (BA[1:0] = 10 and 11). 110 = Quarter Array (BA[1:0] = 11). 111 = Not defined. Valid settings for 8 banks are: 000 = Full Array. 001 = Half Array (BA[2:0] = 000, 001, 010 and 011). 010 = Quarter Array (BA[2:0] = 000, 001). 011 = 1/8 Array (BA[2:0] = 000). 100 = 3/4 Array (BA[2:0] = 010, 011, 100, 101, 110 and 111). 101 = Half Array (BA[2:0] = 100, 101, 110 and 111). 110 = Quarter Array (BA[2:0] = 110 and 111). 111 = 1/8 Array (BA[2:0] 111).

MR3 Register (MR3)

Register	Offset	R/W	Description	Reset Value
MR3	DDRPHY_BA+0x060	R/W	Mode Register 3	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					MPR	MPRLOC	

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:3]	**Reserved**	Reserved.
[2]	**MPR**	**Multi-Purpose Register Enable** Enables, if set, that read data should come from the Multi-Purpose Register. Otherwise read data come from the DRAM array.
[1:0]	**MPRLOC**	**Multi-Purpose Register Location** Selects MPR data location: Valid value are: 00 = Predefined pattern for system calibration. All other settings are reserved and should not be used.

ODTCR Register (ODTCR)

Register	Offset	R/W	Description	Reset Value
ODTCR	DDRPHY_BA+0x064	R/W	ODT Configuration Register	0x0021_0000

31	30	29	28	27	26	25	24
WRODT3				**WRODT2**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
WRODT1				**WRODT0**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RDODT3				**RDODT2**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RDODT1				**RDODT0**			

Bits	Description	
[31:28]	**WRODT3**	**Write ODT** Specifies whether ODT should be enabled ('1') or disabled ('0') on each of the up to four ranks when a write command is sent to rank *n*. WRODT0, WRODT1, WRODT2, and WRODT3 specify ODT settings when a write is to rank 0, rank 1, rank 2, and rank 3, respectively. The four bits of each field each represent a rank, the LSB being rank 0 and the MSB being rank 3. Default is to enable ODT only on rank being written to.
[27:24]	**WRODT2**	**Write ODT** Specifies whether ODT should be enabled ('1') or disabled ('0') on each of the up to four ranks when a write command is sent to rank *n*. WRODT0, WRODT1, WRODT2, and WRODT3 specify ODT settings when a write is to rank 0, rank 1, rank 2, and rank 3, respectively. The four bits of each field each represent a rank, the LSB being rank 0 and the MSB being rank 3. Default is to enable ODT only on rank being written to.
[23:20]	**WRODT1**	**Write ODT** Specifies whether ODT should be enabled ('1') or disabled ('0') on each of the up to four ranks when a write command is sent to rank *n*. WRODT0, WRODT1, WRODT2, and WRODT3 specify ODT settings when a write is to rank 0, rank 1, rank 2, and rank 3, respectively. The four bits of each field each represent a rank, the LSB being rank 0 and the MSB being rank 3. Default is to enable ODT only on rank being written to.
[19:16]	**WRODT0**	**Write ODT** Specifies whether ODT should be enabled ('1') or disabled ('0') on each of the up to four ranks when a write command is sent to rank *n*. WRODT0, WRODT1, WRODT2, and WRODT3 specify ODT settings when a write is to rank 0, rank 1, rank 2, and rank 3, respectively. The four bits of each field each represent a rank, the LSB being rank 0 and the MSB being rank 3. Default is to enable ODT only on rank being written to.
[15:12]	**RDODT3**	**Read ODT** Specifies whether ODT should be enabled ('1') or disabled ('0') on each of the up to four ranks when a read command is sent to rank *n*. RDODT0, RDODT1, RDODT2, and RDODT3 specify ODT settings when a read is to rank 0, rank 1, rank 2, and rank 3, respectively. The four bits of each field each represent a rank, the LSB being rank 0 and the MSB being rank 3. Default is to disable ODT during reads.
[11:8]	**RDODT2**	**Read ODT** Specifies whether ODT should be enabled ('1') or disabled ('0') on each of the up to four ranks when a read command is sent to rank *n*. RDODT0, RDODT1, RDODT2, and RDODT3 specify ODT settings when a read is to rank 0, rank 1, rank 2, and rank 3, respectively. The four bits of each field each represent a rank, the LSB being rank 0 and the MSB being rank 3. Default is to disable ODT during reads.

[7:4]	**RDODT1**	**Read ODT** Specifies whether ODT should be enabled ('1') or disabled ('0') on each of the up to four ranks when a read command is sent to rank *n*. RDODT0, RDODT1, RDODT2, and RDODT3 specify ODT settings when a read is to rank 0, rank 1, rank 2, and rank 3, respectively. The four bits of each field each represent a rank, the LSB being rank 0 and the MSB being rank 3. Default is to disable ODT during reads.
[3:0]	**RDODT0**	**Read ODT** Specifies whether ODT should be enabled ('1') or disabled ('0') on each of the up to four ranks when a read command is sent to rank *n*. RDODT0, RDODT1, RDODT2, and RDODT3 specify ODT settings when a read is to rank 0, rank 1, rank 2, and rank 3, respectively. The four bits of each field each represent a rank, the LSB being rank 0 and the MSB being rank 3. Default is to disable ODT during reads.

DTCR Register (DTCR)

Register	Offset	R/W	Description	Reset Value
DTCR	DDRPHY_BA+0x068	R/W	Data Training Configuration Register	0x9300_3587

31	30	29	28	27	26	25	24
RFSHDT				RANKEN			
23	22	21	20	19	18	17	16
Reserved	DTEXD	DTDSTP	DTDEN	DTDBS			
15	14	13	12	11	10	9	8
Reserved	DTWDQMO	DTBDC	DTWBDDM	DTWDQM			
7	6	5	4	3	2	1	0
DTCMPD	DTMPR	DTRANK		DTRPTN			

Bits	Description	
[31:28]	**RFSHDT**	**Refresh During Training** A non-zero value specifies that a burst of refreshes equal to the number specified in this field should be sent to the SDRAM after training each rank except the last rank.
[27:24]	**RANKEN**	**Rank Enable** Specifies the ranks that are enabled for data-training. Bit 0 controls rank 0, bit 1 controls rank 1, bit 2 controls rank 2, and bit 3 controls rank 3. Setting the bit to '1' enables the rank, and setting it to '0' disables the rank.
[23]	**Reserved**	Reserved.
[22]	**DTEXD**	**Data Training Extended Write DQS** Enables, if set, an extended write DQS whereby two additional pulses of DQS are added as post-amble to a burst of writes. Generally this should only be enabled when running read bit deskew with the intention of performing read eye deskew prior to running write leveling adjustment.
[21]	**DTDSTP**	**Data Training Debug Step** A write of 1 to this bit steps the data training algorithm through a single step. This bit is used to initiate one step of the data training algorithm in question. This bit is self-clearing. To trigger the next step, this bit must be written to again. **Note:** The training steps must be repeated in order to get new data in the "Data Training Eye Data Register 0-1 (DTEDR0-1)". For example, to see the training results for a different lane, select that lane and repeat the training steps to populate DTEDR0 and DTEDR1 with the correct data.
[20]	**DTDEN**	**Data Training Debug Enable** Enables, if set, the data training single step debug mode.
[19:16]	**DTDBS**	**Data Training Debug Byte Select** Selects the byte during data training single step debug mode. **Note:** DTDEN is not used to enable this feature.
[15]	**Reserved**	Reserved.

[14]	**DTWDQMO**	**Data Training WDQ Margin Override** If set, the Training WDQ Margin value specified in DTCR[11:8] (DTWDQM) is used during data training. Otherwise the value is computed as ¼ of the ddr_clk period measurement found during calibration of the WDQ LCDL.
[13]	**DTBDC**	**Data Training Bit Deskew Centering** Enables, if set, eye centering capability during write and read bit deskew training.
[12]	**DTWBDDM**	**Data Training Write Bit Deskew Data Mask** if set, it enables write bit deskew of the data mask
[11:8]	**DTWDQM**	**Training WDQ Margin** Defines how close to 0 or how close to 2*(wdq calibration_value) the WDQ LCDL can be moved during training. Basically defines how much timing margin.
[7]	**DTCMPD**	**Read Data Training Compare Data** Specifies, if set, that DQS gate training should also check if the returning read data is correct. Otherwise data-training only checks if the correct number of DQS edges were returned.
[6]	**DTMPR**	**Read Data Training Using MPR for DDR3 Only** Specifies, if set, that DQS gate training should use the SDRAM Multi-Purpose Register (MPR) register. Otherwise data-training is performed by first writing to some locations in the SDRAM and then reading them back.
[5:4]	**DTRANK**	**Data Training Rank** Select the SDRAM rank to be used during Read DQS gate training, Read/Write Data Bit Deskew, Read/Write Eye Training.
[3:0]	**DTRPTN**	**Data Training Repeat Number** Repeat number used to confirm stability of DDR write or read. **Note:** The minimum value should be 0x4 and the maximum value should be 0x14.

DTAR0 Register (DTAR0)

Register	Offset	R/W	Description	Reset Value
DTAR0	DDRPHY_BA+0x06c	R/W	Data Training Address Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	DTBANK			DTROW			
23	22	21	20	19	18	17	16
DTROW							
15	14	13	12	11	10	9	8
DTROW				DTCOL			
7	6	5	4	3	2	1	0
DTCOL							

Bits	Description	
[31]	**Reserved**	Reserved.
[30:28]	**DTBANK**	**Data Training Bank Address** Selects the SDRAM bank address to be used during data training.
[27:12]	**DTROW**	**Data Training Row Address** Selects the SDRAM row address to be used during data training.
[11:0]	**DTCOL**	**Data Training Column Address** Selects the SDRAM column address to be used during data training. The lower four bits of this address must always be “000”.

DTAR1 Register (DTAR1)

Register	Offset	R/W	Description	Reset Value
DTAR1	DDRPHY_BA+0x070	R/W	Data Training Address Register 1	0x0000_0008

31	30	29	28	27	26	25	24
Reserved	**DTBANK**			**DTROW**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DTROW							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DTROW				**DTCOL**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DTCOL							

Bits	Description	
[31]	**Reserved**	Reserved.
[30:28]	**DTBANK**	**Data Training Bank Address** Selects the SDRAM bank address to be used during data training.
[27:12]	**DTROW**	**Data Training Row Address** Selects the SDRAM row address to be used during data training.
[11:0]	**DTCOL**	**Data Training Column Address** Selects the SDRAM column address to be used during data training. The lower four bits of this address must always be “000”.

DTAR2 Register (DTAR2)

Register	Offset	R/W	Description	Reset Value
DTAR2	DDRPHY_BA+0x074	R/W	Data Training Address Register 2	0x0000_0010

31	30	29	28	27	26	25	24
Reserved	DTBANK			DTROW			
23	22	21	20	19	18	17	16
DTROW							
15	14	13	12	11	10	9	8
DTROW				DTCOL			
7	6	5	4	3	2	1	0
DTCOL							

Bits	Description	
[31]	**Reserved**	Reserved.
[30:28]	**DTBANK**	**Data Training Bank Address** Selects the SDRAM bank address to be used during data training.
[27:12]	**DTROW**	**Data Training Row Address** Selects the SDRAM row address to be used during data training.
[11:0]	**DTCOL**	**Data Training Column Address** Selects the SDRAM column address to be used during data training. The lower four bits of this address must always be “000”.

DTAR3 Register (DTAR3)

Register	Offset	R/W	Description	Reset Value
DTAR3	DDRPHY_BA+0x078	R/W	Data Training Address Register 3	0x0000_0018

31	30	29	28	27	26	25	24
Reserved	DTBANK			DTROW			
23	22	21	20	19	18	17	16
DTROW							
15	14	13	12	11	10	9	8
DTROW				DTCOL			
7	6	5	4	3	2	1	0
DTCOL							

Bits	Description	
[31]	**Reserved**	Reserved.
[30:28]	**DTBANK**	**Data Training Bank Address** Selects the SDRAM bank address to be used during data training.
[27:12]	**DTROW**	**Data Training Row Address** Selects the SDRAM row address to be used during data training.
[11:0]	**DTCOL**	**Data Training Column Address** Selects the SDRAM column address to be used during data training. The lower four bits of this address must always be “000”.

DTDR0 Register (DTDR0)

Register	Offset	R/W	Description	Reset Value
DTDR0	DDRPHY_BA+0x07c	R/W	Data Training Data Register 0	0xdd22_ee11

31	30	29	28	27	26	25	24
DTBYTE3							
23	22	21	20	19	18	17	16
DTBYTE2							
15	14	13	12	11	10	9	8
DTBYTE0							
7	6	5	4	3	2	1	0
DTBYTE1							

Bits	Description	
[31:24]	**DTBYTE3**	**Data Training Data** The first 4 bytes of data used during data training. This same data byte is used for each Byte Lane. Default sequence is a walking 1 while toggling data every data cycle.
[23:16]	**DTBYTE2**	**Data Training Data** The first 4 bytes of data used during data training. This same data byte is used for each Byte Lane. Default sequence is a walking 1 while toggling data every data cycle.
[15:8]	**DTBYTE0**	**Data Training Data** The first 4 bytes of data used during data training. This same data byte is used for each Byte Lane. Default sequence is a walking 1 while toggling data every data cycle.
[7:0]	**DTBYTE1**	**Data Training Data** The first 4 bytes of data used during data training. This same data byte is used for each Byte Lane. Default sequence is a walking 1 while toggling data every data cycle.

DTDR1 Register (DTDR1)

Register	Offset	R/W	Description	Reset Value
DTDR1	DDRPHY_BA+0x080	R/W	Data Training Data Register 1	0x7788_bb44

31	30	29	28	27	26	25	24
DTBYTE7							
23	22	21	20	19	18	17	16
DTBYTE6							
15	14	13	12	11	10	9	8
DTBYTE5							
7	6	5	4	3	2	1	0
DTBYTE4							

Bits	Description	
[31:24]	**DTBYTE7**	**Data Training Data** The second 4 bytes of data used during data training. This same data byte is used for each Byte Lane. Default sequence is a walking 1 while toggling data every data cycle.
[23:16]	**DTBYTE6**	**Data Training Data** The second 4 bytes of data used during data training. This same data byte is used for each Byte Lane. Default sequence is a walking 1 while toggling data every data cycle.
[15:8]	**DTBYTE5**	**Data Training Data** The second 4 bytes of data used during data training. This same data byte is used for each Byte Lane. Default sequence is a walking 1 while toggling data every data cycle.
[7:0]	**DTBYTE4**	**Data Training Data** The second 4 bytes of data used during data training. This same data byte is used for each Byte Lane. Default sequence is a walking 1 while toggling data every data cycle.

DTEDR0 Register (DTEDR0)

Register	Offset	R/W	Description	Reset Value
DTEDR0	DDRPHY_BA+0x084	R	Data Training Eye Data Register 0	0x0000_0000

31	30	29	28	27	26	25	24
DTWBMX							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DTWBMN							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DTWLMX							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DTWLMN							

Bits	Description	
[31:24]	**DTWBMX**	Data Training Write BDL Shift Maximum.
[23:16]	**DTWBMN**	Data Training Write BDL Shift Minimum.
[15:8]	**DTWLMX**	Data Training WDQ LCDL Maximum.
[7:0]	**DTWLMN**	Data Training WDQ LCDL Minimum.

DTEDR1 Register (DTEDR1)

Register	Offset	R/W	Description	Reset Value
DTEDR1	DDRPHY_BA+0x088	R	Data Training Eye Data Register 1	0x0000_0000

31	30	29	28	27	26	25	24
DTRBMX							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DTRBMN							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DTRLMX							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DTRLMN							

Bits	Description	
[31:24]	**DTRBMX**	Data Training Read BDL Shift Maximum.
[23:16]	**DTRBMN**	Data Training Read BDL Shift Minimum.
[15:8]	**DTRLMX**	Data Training RDQS LCDL Maximum.
[7:0]	**DTRLMN**	Data Training RDQS LCDL Minimum.

PGCR2 Register (PGCR2)

Register	Offset	R/W	Description	Reset Value
PGCR2	DDRPHY_BA+0x08c	R/W	PHY General Configuration Register 2	0x00f1_2480

31	30	29	28	27	26	25	24
DYNACPDD	LPMSTRC0	ACPDDC	SHRAC	DTPMXTMR			
23	22	21	20	19	18	17	16
DTPMXTMR				FXDLAT	NOBUB	tREFPRD	
15	14	13	12	11	10	9	8
tREFPRD							
7	6	5	4	3	2	1	0
tREFPRD							

Bits	Description	
[31]	**DYNACPDD**	**Dynamic AC Power Down Driver** Powers down, when set, the output driver on I/O for ADDR and BA. This bit is ORed with bit ACIOCR[3] (ACPDD). **Note:** Asserting PDD puts the I/O driver cell into a lower power, lower speed mode of operation. However, it will still drive if its OE is asserted. ODT will be disabled (if used). Asserting PDD does not prevent the I/O from driving.
[30]	**LPMSTRC0**	**Low-Power Master Channel 0** Set to 1 to have channel 0 act as master to drive channel 1 low-power functions simultaneously. Only valid in shared-AC mode.
[29]	**ACPDDC**	**AC Power-Down with Dual Channels** Set to 1 to power-down address/command lane when both data channels are powered-down. Only valid in shared-AC mode.
[28]	**SHRAC**	**Shared-AC mode** Set to 1 to enable shared address/command mode with two independent data channels – available only if shared address/command mode support is compiled in.
[27:20]	**DTPMXTMR**	**Data Training PUB Mode Timer Exit** Specifies the number of controller clocks to wait when entering and exiting pub mode data training. The default value ensures controller refreshes do not cause memory model errors when entering and exiting data training. The value should be increased if controller initiated SDRAM ZQ short or long operation may occur just before or just after the execution of data training.
[19]	**FXDLAT**	**Fixed Latency** Specified whether all reads should be returned to the controller with a fixed read latency. Enabling fixed read latency increases the read latency. Valid values are: 0 = Fixed read latency Disabled. 1 = Fixed read latency Enabled. Fixed read latency is calculated as (12 + (maximum DXnGTR.RxDGSL)/2) HDR clock cycles

[18]	**NOBUB**	**No Bubbles** Specified whether reads should be returned to the controller with no bubbles. Enabling no-bubble reads increases the read latency. Valid values are: 0 = Bubbles are allowed during reads. 1 = Bubbles are not allowed during reads.
[17:0]	**t_{REFPRD}**	**Refresh Period** Indicates the period, after which the PUB has to issue a refresh command to the SDRAM. This is derived from the maximum refresh interval from the Datasheet, tRFC(max) or REFI, divided by the clock cycle time. A further 400 clocks must be subtracted from the derived number to account for command flow and missed slots of refreshes in the internal PUB blocks. The default corresponds to DDR3 9*7.8us at 1066 MHz when a burst of 9 refreshes are issued at every refresh interval.

RDIMMGCR0 Register (RDIMMGCR0)

Register	Offset	R/W	Description	Reset Value
RDIMMGCR0	DDRPHY_BA+0x0b0	R	RDIMM General Configuration Register 0	0x3c55_0000

31	30	29	28	27	26	25	24
MIRROR	QCSEN	MIRROROE	QCSENOE	RDIMMIOM	RDIMMPDR	RDIMMPDD	RDIMMODT
23	22	21	20	19	18	17	16
ERROUTOE	ERROUTIOM	ERROUTPDR	ERROUTPDD	ERROUTODT	PARINOE	PARINIOM	PARINPDR
15	14	13	12	11	10	9	8
PARINPDD	PARINODT	Reserved					
7	6	5	4	3	2	1	0
Reserved					SOPERR	ERRNOREG	RDIMM

Bits	Description	
[31]	MIRROR	**RDIMM Mirror** Selects between two different ballouts of the RDIMM buffer chip for front or back operation. This register bit controls the buffer chip MIRROR signal.
[30]	QCSEN	**RDMIMM Quad CS Enable** Enables, if set, the Quad CS mode for the RDIMM registering buffer chip. This register bit controls the buffer chip QCSEN# signal.
[29]	MIRROROE	**MIRROR Output Enable** Enables, when set, the output driver on the I/O for MIRROR pin.
[28]	QCSENOE	**QCSEN# Output Enable** Enables, when set, the output driver on the I/O for QCSEN# pin.
[27]	RDIMMIOM	**RDIMM Outputs I/O Mode** Selects SSTL mode (when set to 0) or CMOS mode (when set to 1) of the I/O for QCSEN# and MIRROR pins.
[26]	RDIMMPDR	**RDIMM Outputs Power Down Receiver** Powers down, when set, the input receiver on the I/O for QCSEN# and MIRROR pins.
[25]	RDIMMPDD	**RDIMM Outputs Power Down Driver** Powers down, when set, the output driver on the I/O for QCSEN# and MIRROR pins.
[24]	RDIMMODT	**RDIMM Outputs On-Die Termination** Enables, when set, the on-die termination on the I/O for QCSEN# and MIRROR pins.
[23]	ERROUTOE	**ERROUT# Output Enable** Enables, when set, the output driver on the I/O for ERROUT# pin.
[22]	ERROUTIOM	**ERROUT# I/O Mode** Selects SSTL mode (when set to 0) or CMOS mode (when set to 1) of the I/O for ERROUT# pin.
[21]	ERROUTPDR	**ERROUT# Power Down Receiver** Powers down, when set, the input receiver on the I/O for ERROUT# pin.

[20]	**ERROUTPDD**	**ERROUT# Power Down Driver** Powers down, when set, the output driver on the I/O for ERROUT# pin.
[19]	**ERROUTODT**	**ERROUT# On-Die Termination** Enables, when set, the on-die termination on the I/O for ERROUT# pin.
[18]	**PARINOE**	**PAR_IN Output Enable** Enables, when set, the output driver on the I/O for PAR_IN pin.
[17]	**PARINIOM**	**PAR_IN I/O Mode** Selects SSTL mode (when set to 0) or CMOS mode (when set to 1) of the I/O for PAR_IN pin.
[16]	**PARINPDR**	**PAR_IN Power Down Receiver** Powers down, when set, the input receiver on the I/O for PAR_IN pin.
[15]	**PARINPDD**	**PAR_IN Power Down Driver** Powers down, when set, the output driver on the I/O for PAR_IN pin.
[14]	**PARINODT**	**PAR_IN On-Die Termination** Enables, when set, the on-die termination on the I/O for PAR_IN pin.
[13:3]	**Reserved**	Return zeroes on reads.
[2]	**SOPERR**	**Stop On Parity Error** Indicates, if set, that the PUB is to stop driving commands to the DRAM upon encountering a parity error. Transactions can resume only after status is cleared via PIR.CLRSR.
[1]	**ERRNOREG**	**Parity Error No Registering** Indicates, if set, that parity error signal from the RDIMM should be passed to the DFI controller without any synchronization or registering.
[0]	**RDIMM**	**Registered DIMM** Indicates, if set, that a registered DIMM is used. In this case, the PUB increases the SDRAM write and read latencies (WL/RL) by 1 and also enforces that accesses adhere to RDIMM buffer chip. This only applies to PUB internal SDRAM transactions. Transactions generated by the controller must make its own adjustments to WL/RL when using a registered DIMM. The DCR.NOSRA register bit must be set to '1' if using the standard RDIMM buffer chip so that normal DRAM accesses do not assert multiple chip select bits at the same time.

RDIMMGCR1 Register (RDIMMGCR1)

Register	Offset	R/W	Description	Reset Value
RDIMMGCR1	DDRPHY_BA+0x0b4	R	RDIMM General Configuration Register 1	0xffff_0c80

31	30	29	28	27	26	25	24
CRINIT							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CRINIT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	tBCMRD			tBCSTAB			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
tBCSTAB							

Bits	Description	
[31:16]	**CRINIT**	**Control Registers Initialization Enable** Indicates which RDIMM buffer chip control registers (RC0 to RC15) should be initialized (written) when the PUB is triggered to initialize the buffer chip. A setting of '1' on CRINIT[n] bit means that CRn should be written during initialization.
[15]	**Reserved**	Return zeroes on reads.
[14:12]	**t_{BCMRD}**	**Command word to command word programming delay** Number of DRAM clock cycles between two RDIMM buffer chip command programming accesses. The value used for t_{BCMRD} is 8 plus the value programmed in these bits, i.e. t_{BCMRD} value ranges from 8 to 15. This parameter corresponds to the buffer chip t_{MRD} parameter.
[11:0]	**t_{BCSTAB}**	**Stabilization time** Number of DRAM clock cycles for the RDIMM buffer chip to stabilize. This parameter corresponds to the buffer chip t_{STAB} parameter. Default value is in decimal format and corresponds to 6us at 533 MHz.

RDIMMCR0 Register (RDIMMCR0)

Register	Offset	R/W	Description	Reset Value
RDIMMCR0	DDRPHY_BA+0x0b8	R	RDIMM Control Register 0	0x0000_0000

31	30	29	28	27	26	25	24
RC7				RC6			
23	22	21	20	19	18	17	16
RC5				RC4			
15	14	13	12	11	10	9	8
RC3				RC2			
7	6	5	4	3	2	1	0
RC1				RC0			

Bits	Description	
[31:28]	**RC7**	**Control Word 7** Reserved, free to use by vendor.
[27:24]	**RC6**	**Control Word 6** Reserved, free to use by vendor.
[23:20]	**RC5**	**Control Word 5 for CK Driver Characteristics Control Word** RC5[1:0] is driver settings for clock Y1, Y1#, Y3, and Y3# outputs, and RC5[3:2] is driver settings for clock Y0, Y0#, Y2, and Y2# outputs. Bit definitions are: 00 = Light drive (4 or 5 DRAM loads). 01 = Moderate drive (8 or 10 DRAM loads). 10 = Strong drive (16 or 20 DRAM loads). 11 = Reserved.
[19:16]	**RC4**	**Control Word 4 for Control Signals Driver Characteristics Control Word** RC4[1:0] is driver settings for control A outputs, and RC4[3:2] is driver settings for control B outputs. Bit definitions are: 00 = Light drive (4 or 5 DRAM loads). 01 = Moderate drive (8 or 10 DRAM loads). 10 = Reserved. 11 = Reserved.
[15:12]	**RC3**	**Control Word 3 for Command/Address Signals Driver Characteristics Control Word** RC3[1:0] is driver settings for command/address A outputs, and RC3[3:2] is driver settings for command/address B outputs. Bit definitions are: 00 = Light drive (4 or 5 DRAM loads). 01 = Moderate drive (8 or 10 DRAM loads). 10 = Strong drive (16 or 20 DRAM loads). 11 = Reserved.

[11:8]	**RC2**	**Control Word 2 for Timing Control Word** Bit definitions are: RC2[0]: 0 = Standard (1/2 clock) pre-launch, 1 = Prelaunch controlled by RC1. RC2[1]: 0 = Reserved. RC2[2]: 0 = 100 Ohm input bus termination, 1 = 150 Ohm input bus termination. RC2[3]: 0 = Operation frequency band 1, 1 = Test mode frequency band 2.
[7:4]	**RC1**	**Control Word 1 for Clock Driver Enable Control Word** Bit definitions are: RC1[0]: 0 = Y0/Y0# clock enabled, 1 = Y0/Y0# clock disabled. RC1[1]: 0 = Y1/Y1# clock enabled, 1 = Y1/Y1# clock disabled. RC1[2]: 0 = Y2/Y2# clock enabled, 1 = Y2/Y2# clock disabled. RC1[3]: 0 = Y3/Y3# clock enabled, 1 = Y3/Y3# clock disabled.
[3:0]	**RC0**	**Control Word 0 for Global Features Control Word** Bit definitions are: RC0[0]: 0 = Output inversion enabled, 1 = Output inversion disabled. RC0[1]: 0 = Floating outputs disabled, 1 = Floating outputs enabled. RC0[2]: 0 = A outputs enabled, 1 = A outputs disabled. RC0[3]: 0 = B outputs enabled, 1 = B outputs disabled.

RDIMMCR1 Register (RDIMMCR1)

Register	Offset	R/W	Description	Reset Value
RDIMMCR1	DDRPHY_BA+0x0bc	R	RDIMM Control Register 1	0x0000_0000

31	30	29	28	27	26	25	24
RC15				RC14			
23	22	21	20	19	18	17	16
RC13				RC12			
15	14	13	12	11	10	9	8
RC11				RC10			
7	6	5	4	3	2	1	0
RC9				RC8			

Bits	Description	
[31:28]	**RC15**	**Control Word 15** Reserved for future use
[27:24]	**RC14**	**Control Word 14** Reserved for future use
[23:20]	**RC13**	**Control Word 13** Reserved for future use.
[19:16]	**RC12**	**Control Word 12** Reserved for future use.
[15:12]	**RC11**	**Control Word 11 for Operating Voltage V_{DD} Control Word** RC10[1:0] is V_{DD} operating voltage setting as follows: 00 = DDR3 1.5V mode. 01 = DDR3L 1.35V mode. 10 = Reserved. 11 = Reserved. RC10[3:2]: Reserved.
[11:8]	**RC10**	**Control Word 10 for RDIMM Operating Speed Control Word** RC10[2:0] is RDIMM operating speed setting as follows: 000 = DDR3/DDR3L-800. 001 = DDR3/DDR3L-1066. 010 = DDR3/DDR3L-1333. 011 = DDR3/DDR3L-1600. 100 = Reserved. 101 = Reserved. 110 = Reserved. 111 = Reserved. RC10[3]: Don't care.

[7:4]	**RC9**	**Control Word 9 for Power Saving Settings Control Word** Bit definitions are: RC9[0]: 0 = Floating outputs as defined in RC0, 1 = Weak drive enabled. RC9[1]: 0 = Reserved. RC9[2]: 0 = CKE power down with IBT ON, QxODT is a function of DxODT, 1 = CKE power down with IBT off, QxODT held LOW. RC9[2] is valid only when RC9[3] is 1. RC9[3]: 0 = CKE Power-down mode disabled, 1 = CKE Power-down mode enabled.
[3:0]	**RC8**	**Control Word 8 for Additional Input Bus Termination Setting Control Word** RC8[2:0] is Input Bus Termination (IBT) setting as follows: 000 = IBT as defined in RC2. 001 = Reserved. 010 = 200 Ohm. 011 = Reserved. 100 = 300 Ohm. 101 = Reserved. 110 = Reserved. 111 = Off. RC8[3]: 0 = IBT off when MIRROR is HIGH, 1 = IBT on when MIRROR is high.

DCUAR Register (DCUAR)

Register	Offset	R/W	Description	Reset Value
DCUAR	DDRPHY_BA+0x0c0	R/W	DCU Address Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				ATYPE	INCA	CSEL	
7	6	5	4	3	2	1	0
CSADDR				CWADDR			

Bits	Description	
[31:12]	**Reserved**	Reserved.
[11]	**ATYPE**	**Access Type** Specifies the type of access to be performed using this address. Valid values are: 0 = Write access. 1 = Read access.
[10]	**INCA**	**Increment Address** Specifies, if set, that the cache address specified in WADDR and SADDR should be automatically incremented after each access of the cache. The increment happens in such a way that all the slices of a selected word are first accessed before going to the next word.
[9:8]	**CSEL**	**Cache Select** Selects the cache to be accessed. Valid values are: 00 = Command cache. 01 = Expected data cache. 10 = Read data cache. 11 = Reserved.
[7:4]	**CSADDR**	**Cache Slice Address** Address of the cache slice to be accessed.
[3:0]	**CWADDR**	**Cache Word Address** Address of the cache word to be accessed.

DCUDR Register (DCUDR)

Register	Offset	R/W	Description	Reset Value
DCUDR	DDRPHY_BA+0x0c4	R/W	DCU Data Register	0x0000_0000

31	30	29	28	27	26	25	24
CDATA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CDATA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CDATA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CDATA							

Bits	Description	
[31:0]	**CDATA**	**Cache Data** Data to be written to or read from a cache. This data corresponds to the cache word slice specified by the DCU Address Register.

DCURR Register (DCURR)

Register	Offset	R/W	Description	Reset Value
DCURR	DDRPHY_BA+0x0c8	R/W	DCU Run Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
XCEN	RCEN	SCOF	SONF	NFAIL			
15	14	13	12	11	10	9	8
NFAIL				EADDR			
7	6	5	4	3	2	1	0
SADDR				DINST			

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23]	**XCEN**	**Expected Compare Enable** Indicates, if set, that read data coming back from the SDRAM should be should be compared with the expected data.
[22]	**RCEN**	**Read Capture Enable** Indicates, if set, that read data coming back from the SDRAM should be captured into the read data cache.
[21]	**SCOF**	**Stop Capture On Full** Specifies, if set, that the capture of read data should stop when the capture cache is full.
[20]	**SONF**	**Stop On Nth Fail** Specifies, if set, that the execution of commands and the capture of read data should stop when there are N read data failures. The number of failures is specified by NFAIL. Otherwise commands execute until the end of the program or until manually stopped using a STOP command.
[19:12]	**NFAIL**	**Number of Failures** Specifies the number of failures after which the execution of commands and the capture of read data should stop if SONF bit of this register is set. Execution of commands and the capture of read data will stop after (NFAIL+1) failures if SONF is set. Valid values are from 0 to 254.
[11:8]	**EADDR**	**End Address** Cache word address where the execution of command should end.
[7:4]	**SADDR**	**Start Address** Cache word address where the execution of commands should begin.

[3:0]	**DINST**	**DCU Instruction** Selects the DCU command to be executed: Valid values are: 0000 = NOP: No operation. 0001 = Run: Triggers the execution of commands in the command cache. 0010 = Stop: Stops the execution of commands in the command cache. 0011 = Stop Loop: Stops the execution of an infinite loop in the command cache. 0100 = Reset: Resets all DCU run time registers. See “DCU Status” for details. 0101 – 1111 Reserved

DCULR Register (DCULR)

Register	Offset	R/W	Description	Reset Value
DCULR	DDRPHY_BA+0x0cc	R/W	DCU Loop Register	0xf000_0000

31	30	29	28	27	26	25	24
XLEADDR				Reserved			
23	22	21	20	19	18	17	16
Reserved						IDA	LINF
15	14	13	12	11	10	9	8
LCNT							
7	6	5	4	3	2	1	0
LEADDR				LSADDR			

Bits	Description	
[31:28]	**XLEADDR**	**Expected Data Loop End Address** The last expected data cache word address that contains valid expected data. Expected data should be looped between 0 and this address. XLEADDR field uses only the following bits based on the cache depth: ■ DCU expected data cache = 4, XLEADDR[1:0]. ■ DCU expected data cache = 8, XLEADDR[2:0]. ■ DCU expected data cache = 16, XLEADDR[3:0].
[27:18]	**Reserved**	Reserved.
[17]	**IDA**	**Increment DRAM Address** Indicates, if set, that DRAM addresses should be incremented every time a DRAM read/write command inside the loop is executed.
[16]	**LINF**	**Loop Infinite** Indicates, if set, that the loop should be executed indefinitely until stopped by the STOP command. Otherwise the loop is execute LCNT times.
[15:8]	**LCNT**	**Loop Count** The number of times that the loop should be executed if LINF is not set.
[7:4]	**LEADDR**	**Loop End Address** Command cache word address where the loop should end.
[3:0]	**LSADDR**	**Loop Start Address** Command cache word address where the loop should start.

DCUGCR Register (DCUGCR)

Register	Offset	R/W	Description	Reset Value
DCUGCR	DDRPHY_BA+0x0d0	R/W	DCU General Configuration Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RCSW							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RCSW							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**RCSW**	**Read Capture Start Word** The capture and compare of read data should start after Nth word. For example setting this value to 12 will skip the first 12 read data.

DCUTPR Register (DCUTPR)

Register	Offset	R/W	Description	Reset Value
DCUTPR	DDRPHY_BA+0x0d4	R/W	DCU Timing Parameter Register	0x0000_0000

31	30	29	28	27	26	25	24
tDCUT3							
23	22	21	20	19	18	17	16
tDCUT2							
15	14	13	12	11	10	9	8
tDCUT1							
7	6	5	4	3	2	1	0
tDCUT0							

Bits	Description	
[31:24]	**t_{DCUT3}**	DCU Generic Timing Parameter 3
[23:16]	**t_{DCUT2}**	DCU Generic Timing Parameter 2
[15:8]	**t_{DCUT1}**	DCU Generic Timing Parameter 1
[7:0]	**t_{DCUT0}**	DCU Generic Timing Parameter 0

DCUSR0 Register (DCUSR0)

Register	Offset	R/W	Description	Reset Value
DCUSR0	DDRPHY_BA+0x0d8	R	DCU Status Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					CFULL	CFAIL	RDONE

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**CFULL**	**Capture Full** Indicates, if set, that the capture cache is full.
[1]	**CFAIL**	**Capture Fail** Indicates, if set, that at least one read data word has failed.
[0]	**RDONE**	**Run Done** Indicates, if set, that the DCU has finished executing the commands in the command cache. This bit is also set to indicate that a STOP command has successfully been executed and command execution has stopped.

DCUSR1 Register (DCUSR1)

Register	Offset	R/W	Description	Reset Value
DCUSR1	DDRPHY_BA+0x0dc	R	DCU Status Register 1	0x0000_0000

31	30	29	28	27	26	25	24
LPCNT							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
FLCNT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RDCNT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RDCNT							

Bits	Description	
[31:24]	**LPCNT**	**Loop Count** Indicates the value of the loop count. This is useful when the program has stopped because of failures to assess how many reads were executed before first fail.
[23:16]	**FLCNT**	**Fail Count** Number of read words that have failed.
[15:0]	**RDCNT**	**Read Count** Number of read words returned from the SDRAM.

AACR Register (AACR)

Register	Offset	R/W	Description	Reset Value
AACR	DDRPHY_BA+0x174	R/W	Anti-Aging Control Register	0x0000_00ff

31	30	29	28	27	26	25	24
AAOENC	AAENC	AATR					
23	22	21	20	19	18	17	16
AATR							
15	14	13	12	11	10	9	8
AATR							
7	6	5	4	3	2	1	0
AATR							

Bits	Description	
[31]	AAOENC	**Anti-Aging PAD Output Enable Control** Enables, if set, anti-aging toggling on the pad output enable signal “ctl_oe_n” going into the DATX8s. This will increase power consumption for the anti-aging feature.
[30]	AAENC	**Anti-Aging Enable Control** Enables, if set, the automatic toggling of the data going to the DATX8 when the data channel from the controller/PUB to DATX8 is idle for programmable number of clock cycles.
[29:0]	AATR	**Anti-Aging Toggle Rate** Defines the number of controller clock (ctl_clk) cycles after which the PUB will toggle the data going to DATX8 if the data channel between the controller/PUB and DATX8 has been idle for this long. The default value corresponds to a toggling count of 4096 ctl_clk cycles. For a ctl_clk running at 533 MHz the toggle rate will be approximately 7.68us. The default value may also be overridden by the macro DWC_AACR_AATR_DFLT.

GPR0 Register (GPR0)

Register	Offset	R/W	Description	Reset Value
GPR0	DDRPHY_BA+0x178	R/W	General Purpose Register 0	0x0000_0000

31	30	29	28	27	26	25	24
GPR0							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
GPR0							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
GPR0							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR0							

Bits	Description	
[31:0]	**GPR0**	**General Purpose Register 0** General purpose register bits.

GPR1 Register (GPR1)

Register	Offset	R/W	Description	Reset Value
GPR1	DDRPHY_BA+0x17c	R/W	General Purpose Register 1	0x0000_0000

31	30	29	28	27	26	25	24
GPR1							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
GPR1							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
GPR1							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR1							

Bits	Description	
[31:0]	**GPR1**	**General Purpose Register 1** General purpose register bits.

ZQ0~3CR0 Register (ZQ0~3CR0)

Register	Offset	R/W	Description	Reset Value
ZQ0CR0	DDRPHY_BA+0x180	R/W	Impedance Control Register 0 for ZQ0	0x4000_014a
ZQ1CR0	DDRPHY_BA+0x190	R/W	Impedance Control Register 0 for ZQ1	0x0000_0000
ZQ2CR0	DDRPHY_BA+0x1a0	R/W	Impedance Control Register 0 for ZQ2	0x0000_0000
ZQ3CR0	DDRPHY_BA+0x1b0	R/W	Impedance Control Register 0 for ZQ3	0x0000_0000

31	30	29	28	27	26	25	24
ZQPD	ZCALEN	ZCALBYP	ZDEN	ZDATA			
23	22	21	20	19	18	17	16
ZDATA							
15	14	13	12	11	10	9	8
ZDATA							
7	6	5	4	3	2	1	0
ZDATA							

Bits	Description	
[31]	**ZQPD**	**ZQ Power Down** Powers down, if set, the PZQ cell.
[30]	**ZCALEN**	**Impedance Calibration Enable** Enables, if set, the impedance calibration of this ZQ control block when impedance calibration is triggered using either the ZCAL bit of PIR register or the DFI update interface.
[29]	**ZCALBYP**	**Impedance Calibration Bypass** Bypasses, if set, impedance calibration of this ZQ control block when impedance calibration is already in progress. Impedance calibration can be disabled prior to trigger by using the ZCALEN bit.
[28]	**ZDEN**	**Impedance Over-Ride Enable** When this bit is set, it allows users to directly drive the impedance control using the data programmed in the ZDATA field. Otherwise, the control is generated automatically by the impedance control logic.
[27:0]	**ZDATA**	**Impedance Over-Ride Data** Data used to directly drive the impedance control. ZDATA field mapping for D3R I/O is as follows: ZDATA[27:20] is reserved and returns zeros on reads. ZDATA[19:15] is used to select the pull-up on-die termination impedance. ZDATA[14:10] is used to select the pull-down on-die termination impedance. ZDATA[9:5] is used to select the pull-up output impedance ZDATA[4:0] is used to select the pull-down output impedance. **Note:** The default value is 0x000014A for I/O type D3R.

ZQ0~3CR1 Register (ZQ0~3CR1)

Register	Offset	R/W	Description	Reset Value
ZQ0CR1	DDRPHY_BA+0x184	R/W	Impedance Control Register 1 for ZQ0	0x0000_107b
ZQ1CR1	DDRPHY_BA+0x194	R/W	Impedance Control Register 1 for ZQ1	0x0000_0000
ZQ2CR1	DDRPHY_BA+0x1a4	R/W	Impedance Control Register 1 for ZQ2	0x0000_0000
ZQ3CR1	DDRPHY_BA+0x1b4	R/W	Impedance Control Register 1 for ZQ3	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						DFIPU1	DFIPU0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	DFICCU	DFICU1	DFICU0	Reserved			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ZPROG							

Bits	Description	
[31:18]	**Reserved**	Return zeroes on reads.
[17]	**DFIPU1**	**DFI Update Interface 1** Sets this impedance controller to be enabled for calibration when the DFI PHY update interface 1 (channel 1) requests an update. Only valid in shared-AC mode.
[16]	**DFIPU0**	**DFI Update Interface 0** Sets this impedance controller to be enabled for calibration when the DFI PHY update interface 0 (channel 0) requests an update.
[15]	**Reserved**	Return zeroes on reads.
[14]	**DFICCU**	**DFI Concurrent Controller Update Interface** Sets this impedance controller to be enabled for calibration when both of the DFI controller update interfaces request an update on the same clock. This provides the ability to enable impedance calibration updates for the Address/Command lane. Only valid in shared-AC mode.
[13]	**DFICU1**	**DFI Controller Update Interface 1** Sets this impedance controller to be enabled for calibration when the DFI controller update interface 1 (channel 1) requests an update. Only valid in shared-AC mode.
[12]	**DFICU0**	**DFI Controller Update Interface 0** Sets this impedance controller to be enabled for calibration when the DFI controller update interface 0 (channel 0) requests an update.
[11:8]	**Reserved**	Return zeroes on reads.
[7:0]	**ZPROG**	**Impedance Divide Ratio** Selects the external resistor divide ratio to be used to set the output impedance and the on-die termination as follows: ZPROG[7:4] = On-die termination divide select. ZPROG[3:0] = Output impedance divide select.

ZQ0~3SR0 Register (ZQ0~3SR0)

Register	Offset	R/W	Description	Reset Value
ZQ0SR0	DDRPHY_BA+0x188	R	Impedance Status Register 0 for ZQ0	0x0000_014a
ZQ1SR0	DDRPHY_BA+0x198	R	Impedance Status Register 0 for ZQ1	0x0000_0000
ZQ2SR0	DDRPHY_BA+0x1a8	R	Impedance Status Register 0 for ZQ2	0x0000_0000
ZQ3SR0	DDRPHY_BA+0x1b8	R	Impedance Status Register 0 for ZQ3	0x0000_0000

31	30	29	28	27	26	25	24
ZDONE	ZERR	Reserved		ZCTRL			
23	22	21	20	19	18	17	16
ZCTRL							
15	14	13	12	11	10	9	8
ZCTRL							
7	6	5	4	3	2	1	0
ZCTRL							

Bits	Description	
[31]	**ZDONE**	**Impedance Calibration Done** Indicates that impedance calibration has completed.
[30]	**ZERR**	**Impedance Calibration Error** If set, indicates that there was an error during impedance calibration.
[29:28]	**Reserved**	Return zeroes on reads.
[27:0]	**ZCTRL**	**Impedance Control** Current value of impedance control. ZCTRL field mapping for D3R I/O is as follows: ZCTRL[27:20] is reserved and returns zeros on reads. ZCTRL[19:15] is used to select the pull-up on-die termination impedance. ZCTRL[14:10] is used to select the pull-down on-die termination impedance. ZCTRL[9:5] is used to select the pull-up output impedance. ZCTRL[4:0] is used to select the pull-down output impedance. **Note:** The default value is 0x000014A for I/O type D3R.

ZQ0~3SR1 Register (ZQ0~3SR1)

Register	Offset	R/W	Description	Reset Value
ZQ0SR1	DDRPHY_BA+0x18c	R	Impedance Status Register 1 for ZQ0	0x0000_0000
ZQ1SR1	DDRPHY_BA+0x19c	R	Impedance Status Register 1 for ZQ1	0x0000_0000
ZQ2SR1	DDRPHY_BA+0x1ac	R	Impedance Status Register 1 for ZQ2	0x0000_0000
ZQ3SR1	DDRPHY_BA+0x1bc	R	Impedance Status Register 1 for ZQ3	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
OPU		OPD		ZPU		ZPD	

Bits	Description	
[31:8]	**Reserved**	Return zeroes on reads.
[7:6]	**OPU**	**On-die termination pull-up calibration status** Similar status encodings as ZPD.
[5:4]	**OPD**	**On-die termination pull-down calibration status** Similar status encodings as ZPD.
[3:2]	**ZPU**	**Output impedance pull-up calibration status** Similar status encodings as ZPD.
[1:0]	**ZPD**	**Output impedance pull-down calibration status** Valid status encodings are: 00 = Completed with no errors. 01 = Overflow error. 10 = Underflow error. 11 = Calibration in progress.

DX0~8GCR Register (DX0~8GCR)

Register	Offset	R/W	Description	Reset Value
DX0GCR	DDRPHY_BA+0x1c0	R/W	DATX8 General Configuration Register for Byte Lane0	0x7c00_0e81
DX1GCR	DDRPHY_BA+0x200	R/W	DATX8 General Configuration Register for Byte Lane1	0x7c00_0e81
DX2GCR	DDRPHY_BA+0x240	R/W	DATX8 General Configuration Register for Byte Lane2	0x0000_0000
DX3GCR	DDRPHY_BA+0x280	R/W	DATX8 General Configuration Register for Byte Lane3	0x0000_0000
DX4GCR	DDRPHY_BA+0x2c0	R/W	DATX8 General Configuration Register for Byte Lane4	0x0000_0000
DX5GCR	DDRPHY_BA+0x300	R/W	DATX8 General Configuration Register for Byte Lane5	0x0000_0000
DX6GCR	DDRPHY_BA+0x340	R/W	DATX8 General Configuration Register for Byte Lane6	0x0000_0000
DX7GCR	DDRPHY_BA+0x380	R/W	DATX8 General Configuration Register for Byte Lane7	0x0000_0000
DX8GCR	DDRPHY_BA+0x3c0	R/W	DATX8 General Configuration Register for Byte Lane8	0x0000_0000

31	30	29	28	27	26	25	24
CALBYP	**MDLEN**	**WLRKEN**				**Reserved**	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				**PLLBYP**	**GSHIFT**	**PLLPD**	**PLLRST**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DXOEO		**RTTOAL**	**RTTOH**		**DQRTT**	**DQSRTT**	**DSEN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DSEN	**DQSRPD**	**DXPDR**	**DXPDD**	**DXIOM**	**DQODT**	**DQSODT**	**DXEN**

Bits	Description	
[31]	**CALBYP**	**Calibration Bypass** Prevents, if set, period measurement calibration from automatically triggering after PHY initialization.
[30]	**MDLEN**	**Master Delay Line Enable** Enables, if set, the DATX8 master delay line calibration to perform subsequent period measurements following the initial period measurements that are performed after reset or when calibration is manually triggered. These additional measurements are accumulated and filtered as long as this bit remains high. This bit is ANDed with the common DATX8 MDL enable bit.
[29:26]	**WLRKEN**	**Write Level Rank Enable** Specifies the ranks that should be write leveled for this byte. Write leveling responses from ranks that are not enabled for write leveling for a particular byte are ignored and write leveling is flagged as done for these ranks. WLRKEN[0] enables rank 0, [1] enables rank 1, [2] enables rank 2, and [3] enables rank 3.
[25:20]	**Reserved**	Return zeroes on reads.
[19]	**PLLBYP**	**PLL Bypass** Puts the byte PLL in bypass mode by driving the PLL bypass pin. This bit is not self-clearing and a '0' must be written to de-assert the bypass. This bit is ORed with the global BYP configuration bit.
[18]	**GSHIFT**	**Gear Shift** Enables, if set, rapid locking mode on the byte PLL. This bit is ORed with the global GSHIFT configuration bit.

[17]	**PLLPD**	**PLL Power Down** Puts the byte PLL in Power-down mode by driving the PLL power down pin. This bit is not self-clearing and a '0' must be written to de-assert the power-down. This bit is ORed with the global PLLPD configuration bit.
[16]	**PLLRST**	**PLL Rest** Resets the byte PLL by driving the PLL reset pin. This bit is not self- clearing and a '0' must be written to de-assert the reset. This bit is ORed with the global PLLRST configuration bit.
[15:14]	**DXOEO**	**Data Byte Output Enable Override** Specifies whether the output I/O output enable for the byte lane should be set to a fixed value. Valid values are: 00 = No override. Output enable is controlled by DFI transactions 01 = Output enable is asserted (I/O is forced to output mode). 10 = Output enable is de-asserted (I/O is forced to input mode). 11 = Reserved.
[13]	**RTTOAL**	**RTT On Additive Latency** Indicates when the ODT control of DQ/DQS SSTL I/Os is set to the value in DQODT/DQSODT during read cycles. Valid values are: 0 = ODT control is set to DQSODT/DQODT almost two cycles before read data preamble. 1 = ODT control is set to DQSODT/DQODT almost one cycle before read data preamble.
[12:11]	**RTTOH**	**RTT Output Hold** Indicates the number of clock cycles (from 0 to 3) after the read data postamble for which ODT control should remain set to DQSODT for DQS or DQODT for DQ/DM before disabling it (setting it to '0') when using dynamic ODT control. ODT is disabled almost RTTOH clock cycles after the read postamble.
[10]	**DQRTT**	**DQ Dynamic RTT Control** If set, the on die termination (ODT) control of the DQ/DM SSTL I/O is dynamically generated to enable the ODT during read operation and disabled otherwise. By setting this bit to '0' the dynamic ODT feature is disabled. To control ODT statically this bit must be set to '0' and DXnGCR0[2] (DQODT) is used to enable ODT (when set to '1') or disable ODT(when set to '0').
[9]	**DQSRTT**	**DQS Dynamic RTT Control** If set, the on die termination (ODT) control of the DQS/DQS# SSTL I/O is dynamically generated to enable the ODT during read operation and disabled otherwise. By setting this bit to '0' the dynamic ODT feature is disabled. To control ODT statically this bit must be set to '0' and DXnGCR0[1] (DQSODT) is used to enable ODT (when set to '1') or disable ODT(when set to '0').
[8:7]	**DSEN**	**Write DQS Enable** Controls whether the write DQS going to the SDRAM is enabled (toggling) or disabled (static value) and whether the DQS is inverted. DQS# is always the inversion of DQS. These values are valid only when DQS/DQS# output enable is on, otherwise the DQS/DQS# is tristated. Valid settings are: 00 = Reserved. 01 = DQS toggling with normal polarity (This should be the default setting). 10 = Reserved. 11 = Reserved.
[6]	**DQSRPD**	**DQSR Power Down** Powers down, if set, the PDQSR cell. This bit is ORed with the common PDR configuration bit (see "DATX8 Common Configuration Register (DXCCR)")
[5]	**DXPDR**	**Data Power Down Receiver** Powers down, when set, the input receiver on I/O for DQ, DM, and DQS/DQS# pins of the byte. This bit is ORed with the common PDR configuration bit (see "DATX8 Common Configuration Register (DXCCR)").

[4]	**DXPDD**	**Data Power Down Driver** Powers down, when set, the output driver on I/O for DQ, DM, and DQS/DQS# pins of the byte. This bit is ORed with the common PDD configuration bit (see “DATX8 Common Configuration Register (DXCCR)”). **Note:** Asserting PDD puts the I/O driver cell into a lower power, lower speed mode of operation. However, it will still drive if its OE is asserted. ODT will be disabled (if used). Asserting PDD does not prevent the I/O from driving.
[3]	**DXIOM**	**Data I/O Mode** Selects SSTL mode (when set to 0) or CMOS mode (when set to 1) of the I/O for DQ, DM, and DQS/DQS# pins of the byte. This bit is ORed with the IOM configuration bit of the individual DATX8(see “DATX8 Common Configuration Register (DXCCR)”).
[2]	**DQODT**	**Data On-Die Termination** Enables, when set, the on-die termination on the I/O for DQ and DM pins of the byte. This bit is ORed with the common DATX8 ODT configuration bit (see “DATX8 Common Configuration Register (DXCCR)”). **Note:** This bit is only valid when DXnGCR0[10] is '0'.
[1]	**DQSODT**	**DQS On-Die Termination** Enables, when set, the on-die termination on the I/O for DQS/DQS# pin of the byte. This bit is ORed with the common DATX8 ODT configuration bit (see “DATX8 Common Configuration Register (DXCCR)”). **Note:** This bit is only valid when DXnGCR0[9] is '0'.
[0]	**DXEN**	**Data Byte Enable** Enables, if set, the data byte. Setting this bit to '0' disables the byte, i.e. the byte is not used in PHY initialization or training and is ignored during SDRAM read/write operations.

DX0~8GSR0 Register (DX0~8GSR0)

Register	Offset	R/W	Description	Reset Value
DX0GSR0	DDRPHY_BA+0x1c4	R	DATX8 General Status Registers 0 for Byte Lane0	0x0000_0000
DX1GSR0	DDRPHY_BA+0x204	R	DATX8 General Status Registers 0 for Byte Lane1	0x0000_0000
DX2GSR0	DDRPHY_BA+0x244	R	DATX8 General Status Registers 0 for Byte Lane2	0x0000_0000
DX3GSR0	DDRPHY_BA+0x284	R	DATX8 General Status Registers 0 for Byte Lane3	0x0000_0000
DX4GSR0	DDRPHY_BA+0x2c4	R	DATX8 General Status Registers 0 for Byte Lane4	0x0000_0000
DX5GSR0	DDRPHY_BA+0x304	R	DATX8 General Status Registers 0 for Byte Lane5	0x0000_0000
DX6GSR0	DDRPHY_BA+0x344	R	DATX8 General Status Registers 0 for Byte Lane6	0x0000_0000
DX7GSR0	DDRPHY_BA+0x384	R	DATX8 General Status Registers 0 for Byte Lane7	0x0000_0000
DX8GSR0	DDRPHY_BA+0x3c4	R	DATX8 General Status Registers 0 for Byte Lane8	0x0000_0000

31	30	29	28	27	26	25	24
Reserved			WLDQ	QSGERR			
23	22	21	20	19	18	17	16
GDQSPRD							
15	14	13	12	11	10	9	8
DPLOCK	WLPRD						
7	6	5	4	3	2	1	0
WLPRD	WLERR	WLDONE	WLCAL	GDQSCAL	RDQSNCAL	RDQSCAL	WDQCAL

Bits	Description	
[31:29]	**Reserved**	Return zeroes on reads.
[28]	**WLDQ**	**Write Leveling DQ Status** Captures the write leveling DQ status from the DRAM during software write leveling.
[27:24]	**QSGERR**	**DQS Gate Training Error** Indicates, if set, that there is an error in DQS gate training. One bit for each of the up to 4 ranks.
[23:16]	**GDQSPRD**	**Read DQS gating Period** Returns the DDR clock period measured by the read DQS gating LCDL during calibration. This value is PVT compensated.
[15]	**DPLOCK**	**DATX8 PLL Lock** Indicates, if set, that the DATX8 PLL has locked. This is a direct status of the DATX8 PLL lock pin.
[14:7]	**WLPRD**	**Write Leveling Period** Returns the DDR clock period measured by the write leveling LCDL during calibration. The measured period is used to generate the control of the write leveling pipeline which is a function of the write-leveling delay and the clock period. This value is PVT compensated.
[6]	**WLERR**	**Write Leveling Error** Indicates, if set, that there is a write leveling error in the DATX8.

[5]	**WLDONE**	**Write Leveling Done** Indicates, if set, that the DATX8 has completed write leveling.
[4]	**WLCAL**	**Write Leveling Calibration** Indicates, if set, that the DATX8 has finished doing period measurement calibration for the write leveling slave delay line.
[3]	**GDQSCAL**	**Read DQS gating Calibration** Indicates, if set, that the DATX8 has finished doing period measurement calibration for the read DQS gating LCDL.
[2]	**RDQSNCAL**	**Read DQS# Calibration for Type B/B1 PHY Only** Indicates, if set, that the DATX8 has finished doing period measurement calibration for the read DQS# LCDL.
[1]	**RDQSCAL**	**Read DQS Calibration** Indicates, if set, that the DATX8 has finished doing period measurement calibration for the read DQS LCDL.
[0]	**WDQCAL**	**Write DQ Calibration** Indicates, if set, that the DATX8 has finished doing period measurement calibration for the write DQ LCDL.

DX0~8GSR1 Register (DX0~8GSR1)

Register	Offset	R/W	Description	Reset Value
DX0GSR1	DDRPHY_BA+0x1c8	R	DATX8 General Status Registers 1 for Byte Lane0	0x0000_0000
DX1GSR1	DDRPHY_BA+0x208	R	DATX8 General Status Registers 1 for Byte Lane1	0x0000_0000
DX2GSR1	DDRPHY_BA+0x248	R	DATX8 General Status Registers 1 for Byte Lane2	0x0000_0000
DX3GSR1	DDRPHY_BA+0x288	R	DATX8 General Status Registers 1 for Byte Lane3	0x0000_0000
DX4GSR1	DDRPHY_BA+0x2c8	R	DATX8 General Status Registers 1 for Byte Lane4	0x0000_0000
DX5GSR1	DDRPHY_BA+0x308	R	DATX8 General Status Registers 1 for Byte Lane5	0x0000_0000
DX6GSR1	DDRPHY_BA+0x348	R	DATX8 General Status Registers 1 for Byte Lane6	0x0000_0000
DX7GSR1	DDRPHY_BA+0x388	R	DATX8 General Status Registers 1 for Byte Lane7	0x0000_0000
DX8GSR1	DDRPHY_BA+0x3c8	R	DATX8 General Status Registers 1 for Byte Lane8	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							**DLTCODE**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DLTCODE							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DLTCODE							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DLTCODE							**DLTDONE**

Bits	Description	
[31:25]	**Reserved**	Returns zeroes on reads.
[24:1]	**DLTCODE**	**Delay Line Test Code** Returns the code measured by the PHY control block that corresponds to the period of the DATX8 delay line digital test output.
[0]	**DLTDONE**	**Delay Line Test Done** Indicates, if set, that the PHY control block has finished doing period measurement of the DATX8 delay line digital test output.

DX0~8GSR2 Register (DX0~8GSR2)

Register	Offset	R/W	Description	Reset Value
DX0GSR2	DDRPHY_BA+0x1f4	R/W	DATX8 General Status Register 2 for Byte Lane0	0x0000_0000
DX1GSR2	DDRPHY_BA+0x234	R/W	DATX8 General Status Register 2 for Byte Lane1	0x0000_0000
DX2GSR2	DDRPHY_BA+0x274	R/W	DATX8 General Status Register 2 for Byte Lane2	0x0000_0000
DX3GSR2	DDRPHY_BA+0x2b4	R/W	DATX8 General Status Register 2 for Byte Lane3	0x0000_0000
DX4GSR2	DDRPHY_BA+0x2f4	R/W	DATX8 General Status Register 2 for Byte Lane4	0x0000_0000
DX5GSR2	DDRPHY_BA+0x334	R/W	DATX8 General Status Register 2 for Byte Lane5	0x0000_0000
DX6GSR2	DDRPHY_BA+0x374	R/W	DATX8 General Status Register 2 for Byte Lane6	0x0000_0000
DX7GSR2	DDRPHY_BA+0x3b4	R/W	DATX8 General Status Register 2 for Byte Lane7	0x0000_0000
DX8GSR2	DDRPHY_BA+0x3f4	R/W	DATX8 General Status Register 2 for Byte Lane8	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				ESTAT			
7	6	5	4	3	2	1	0
WEWN	WEERR	REWN	REERR	WDWN	WDERR	RDWN	RDERR

Bits	Description	
[31:12]	**Reserved**	Return zeroes on reads.
[11:8]	**ESTAT**	**Error Status** If an error occurred for this lane as indicated by RDERR, WDERR, REERR or WEERR the error status code can provide additional information regard when the error occurred during the algorithm execution.
[7]	**WEWN**	**Write Data Eye Training Warning** Indicates, if set, that the DATX8 has encountered a warning during execution of the write data eye training.
[6]	**WEERR**	**Write Data Eye Training Error** Indicates, if set, that the DATX8 has encountered an error during execution of the write data eye training.
[5]	**REWN**	**Read Data Eye Training Warning** Indicates, if set, that the DATX8 has encountered a warning during execution of the read data eye training.
[4]	**REERR**	**Read Data Eye Training Error** Indicates, if set, that the DATX8 has encountered an error during execution of the read data eye training.
[3]	**WDWN**	**Write Bit Deskew Warning** Indicates, if set, that the DATX8 has encountered a warning during execution of the write bit deskew training.
[2]	**WDERR**	**Write Bit Deskew Error** Indicates, if set, that the DATX8 has encountered an error during execution of the write bit deskew training.

[1]	**RDWN**	**Read Bit Deskew Warning** Indicates, if set, that the DATX8 has encountered a warning during execution of the read bit deskew training.
[0]	**RDERR**	**Read Bit Deskew Error** Indicates, if set, that the DATX8 has encountered an error during execution of the read bit deskew training.

DX0~8BDLR0 Register (DX0~8BDLR0)

Register	Offset	R/W	Description	Reset Value
DX0BDLR0	DDRPHY_BA+0x1cc	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane0	0x0000_0000
DX1BDLR0	DDRPHY_BA+0x20c	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane1	0x0000_0000
DX2BDLR0	DDRPHY_BA+0x24c	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane2	0x0000_0000
DX3BDLR0	DDRPHY_BA+0x28c	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane3	0x0000_0000
DX4BDLR0	DDRPHY_BA+0x2cc	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane4	0x0000_0000
DX5BDLR0	DDRPHY_BA+0x30c	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane5	0x0000_0000
DX6BDLR0	DDRPHY_BA+0x34c	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane6	0x0000_0000
DX7BDLR0	DDRPHY_BA+0x38c	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane7	0x0000_0000
DX8BDLR0	DDRPHY_BA+0x3cc	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane8	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		DQ4WBD					
23	22	21	20	19	18	17	16
DQ3WBD						DQ2WBD	
15	14	13	12	11	10	9	8
DQ2WBD				DQ1WBD			
7	6	5	4	3	2	1	0
DQ1WBD		DQ0WBD					

Bits	Description	
[31:30]	**Reserved**	Return zeroes on reads.
[29:24]	**DQ4WBD**	**DQ4 Write Bit Delay** Delay select for the BDL on DQ4 write path.
[23:18]	**DQ3WBD**	**DQ3 Write Bit Delay** Delay select for the BDL on DQ3 write path
[17:12]	**DQ2WBD**	**DQ2 Write Bit Delay** Delay select for the BDL on DQ2 write path.
[11:6]	**DQ1WBD**	**DQ1 Write Bit Delay** Delay select for the BDL on DQ1 write path.
[5:0]	**DQ0WBD**	**DQ0 Write Bit Delay** Delay select for the BDL on DQ0 write path.

DX0~8BDLR1 Register (DX0~8BDLR1)

Register	Offset	R/W	Description	Reset Value
DX0BDLR1	DDRPHY_BA+0x1d0	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane0	0x0000_0000
DX1BDLR1	DDRPHY_BA+0x210	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane1	0x0000_0000
DX2BDLR1	DDRPHY_BA+0x250	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane2	0x0000_0000
DX3BDLR1	DDRPHY_BA+0x290	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane3	0x0000_0000
DX4BDLR1	DDRPHY_BA+0x2d0	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane4	0x0000_0000
DX5BDLR1	DDRPHY_BA+0x310	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane5	0x0000_0000
DX6BDLR1	DDRPHY_BA+0x350	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane6	0x0000_0000
DX7BDLR1	DDRPHY_BA+0x390	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane7	0x0000_0000
DX8BDLR1	DDRPHY_BA+0x3d0	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane8	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		DSWBD					
23	22	21	20	19	18	17	16
DMWBD						DQ7WBD	
15	14	13	12	11	10	9	8
DQ7WBD				DQ6WBD			
7	6	5	4	3	2	1	0
DQ6WBD		DQ5WBD					

Bits	Description	
[31:30]	**Reserved**	Return zeroes on reads.
[29:24]	**DSWBD**	**DQS Write Bit Delay** Delay select for the BDL on DQS write path
[23:18]	**DMWBD**	**DM Write Bit Delay** Delay select for the BDL on DM write path.
[17:12]	**DQ7WBD**	**DQ7 Write Bit Delay** Delay select for the BDL on DQ7 write path.
[11:6]	**DQ6WBD**	**DQ6 Write Bit Delay** Delay select for the BDL on DQ6 write path.
[5:0]	**DQ5WBD**	**DQ5 Write Bit Delay** Delay select for the BDL on DQ5 write path.

DX0~8BDLR2 Register (DX0~8BDLR2)

Register	Offset	R/W	Description	Reset Value
DX0BDLR2	DDRPHY_BA+0x1d4	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane0	0x0000_0000
DX1BDLR2	DDRPHY_BA+0x214	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane1	0x0000_0000
DX2BDLR2	DDRPHY_BA+0x254	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane2	0x0000_0000
DX3BDLR2	DDRPHY_BA+0x294	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane3	0x0000_0000
DX4BDLR2	DDRPHY_BA+0x2d4	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane4	0x0000_0000
DX5BDLR2	DDRPHY_BA+0x314	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane5	0x0000_0000
DX6BDLR2	DDRPHY_BA+0x354	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane6	0x0000_0000
DX7BDLR2	DDRPHY_BA+0x394	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane7	0x0000_0000
DX8BDLR2	DDRPHY_BA+0x3d4	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane8	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DSNRBD						DSRBD	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DSRBD				DQOEBD			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DQOEBD		DSOEBD					

Bits	Description	
[31:24]	**Reserved**	Return zeroes on reads.
[23:18]	**DSNRBD**	**DQSN Read Bit Delay for Type B/B1 PHY Only** Delay select for the BDL on DQSN read path
[17:12]	**DSRBD**	**DQS Read Bit Delay** Delay select for the BDL on DQS read path
[11:6]	**DQOEBD**	**DQ Output Enable Bit Delay** Delay select for the BDL on DQ/DM output enable path.
[5:0]	**DSOEBD**	**DQS Output Enable Bit Delay** Delay select for the BDL on DQS output enable path

DX0~8BDLR3 Register (DX0~8BDLR3)

Register	Offset	R/W	Description	Reset Value
DX0BDLR3	DDRPHY_BA+0x1d8	R/W	DATX8 Bit Delay Line Register 3 for Byte Lane0	0x0000_0000
DX1BDLR3	DDRPHY_BA+0x218	R/W	DATX8 Bit Delay Line Register 3 for Byte Lane1	0x0000_0000
DX2BDLR3	DDRPHY_BA+0x258	R/W	DATX8 Bit Delay Line Register 3 for Byte Lane2	0x0000_0000
DX3BDLR3	DDRPHY_BA+0x298	R/W	DATX8 Bit Delay Line Register 3 for Byte Lane3	0x0000_0000
DX4BDLR3	DDRPHY_BA+0x2d8	R/W	DATX8 Bit Delay Line Register 3 for Byte Lane4	0x0000_0000
DX5BDLR3	DDRPHY_BA+0x318	R/W	DATX8 Bit Delay Line Register 3 for Byte Lane5	0x0000_0000
DX6BDLR3	DDRPHY_BA+0x358	R/W	DATX8 Bit Delay Line Register 3 for Byte Lane6	0x0000_0000
DX7BDLR3	DDRPHY_BA+0x398	R/W	DATX8 Bit Delay Line Register 3 for Byte Lane7	0x0000_0000
DX8BDLR3	DDRPHY_BA+0x3d8	R/W	DATX8 Bit Delay Line Register 3 for Byte Lane8	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		DQ4RBD					
23	22	21	20	19	18	17	16
DQ3RBD						DQ2RBD	
15	14	13	12	11	10	9	8
DQ2RBD				DQ1RBD			
7	6	5	4	3	2	1	0
DQ1RBD		DQ0RBD					

Bits	Description	
[31:30]	**Reserved**	Return zeroes on reads.
[29:24]	**DQ4RBD**	**DQ4 Read Bit Delay** Delay select for the BDL on DQ4 read path.
[23:18]	**DQ3RBD**	**DQ3 Read Bit Delay** Delay select for the BDL on DQ3 read path
[17:12]	**DQ2RBD**	**DQ2 Read Bit Delay** Delay select for the BDL on DQ2 read path.
[11:6]	**DQ1RBD**	**DQ1 Read Bit Delay** Delay select for the BDL on DQ1 read path.
[5:0]	**DQ0RBD**	**DQ0 Read Bit Delay** Delay select for the BDL on DQ0 read path.

DX0~8BDLR4 Register (DX0~8BDLR4)

Register	Offset	R/W	Description	Reset Value
DX0BDLR4	DDRPHY_BA+0x1dc	R/W	DATX8 Bit Delay Line Register 4 for Byte Lane0	0x0000_0000
DX1BDLR4	DDRPHY_BA+0x21c	R/W	DATX8 Bit Delay Line Register 4 for Byte Lane1	0x0000_0000
DX2BDLR4	DDRPHY_BA+0x25c	R/W	DATX8 Bit Delay Line Register 4 for Byte Lane2	0x0000_0000
DX3BDLR4	DDRPHY_BA+0x29c	R/W	DATX8 Bit Delay Line Register 4 for Byte Lane3	0x0000_0000
DX4BDLR4	DDRPHY_BA+0x2dc	R/W	DATX8 Bit Delay Line Register 4 for Byte Lane4	0x0000_0000
DX5BDLR4	DDRPHY_BA+0x31c	R/W	DATX8 Bit Delay Line Register 4 for Byte Lane5	0x0000_0000
DX6BDLR4	DDRPHY_BA+0x35c	R/W	DATX8 Bit Delay Line Register 4 for Byte Lane6	0x0000_0000
DX7BDLR4	DDRPHY_BA+0x39c	R/W	DATX8 Bit Delay Line Register 4 for Byte Lane7	0x0000_0000
DX8BDLR4	DDRPHY_BA+0x3dc	R/W	DATX8 Bit Delay Line Register 4 for Byte Lane8	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
DMRBD						DQ7RBD	
15	14	13	12	11	10	9	8
DQ7RBD				DQ6RBD			
7	6	5	4	3	2	1	0
DQ6RBD		DQ5RBD					

Bits	Description	
[31:24]	**Reserved**	Return zeroes on reads.
[23:18]	**DMRBD**	**DM Read Bit Delay** Delay select for the BDL on DM read path.
[17:12]	**DQ7RBD**	**DQ7 Read Bit Delay** Delay select for the BDL on DQ7 read path.
[11:6]	**DQ6RBD**	**DQ6 Read Bit Delay** Delay select for the BDL on DQ6 read path.
[5:0]	**DQ5RBD**	**DQ5 Read Bit Delay** Delay select for the BDL on DQ5 read path.

DX0~8LCDLR0 Register (DX0~8LCDLR0)

Register	Offset	R/W	Description	Reset Value
DX0LCDLR0	DDRPHY_BA+0x1e0	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane0	0x0000_0000
DX1LCDLR0	DDRPHY_BA+0x220	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane1	0x0000_0000
DX2LCDLR0	DDRPHY_BA+0x260	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane2	0x0000_0000
DX3LCDLR0	DDRPHY_BA+0x2a0	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane3	0x0000_0000
DX4LCDLR0	DDRPHY_BA+0x2e0	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane4	0x0000_0000
DX5LCDLR0	DDRPHY_BA+0x320	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane5	0x0000_0000
DX6LCDLR0	DDRPHY_BA+0x360	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane6	0x0000_0000
DX7LCDLR0	DDRPHY_BA+0x3a0	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane7	0x0000_0000
DX8LCDLR0	DDRPHY_BA+0x3e0	R/W	DATX8 Bit Delay Line Register 0 for Byte Lane8	0x0000_0000

31	30	29	28	27	26	25	24
R3WLD							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
R2WLD							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
R1WLD							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R0WLD							

Bits	Description	
[31:24]	**R3WLD**	**Rank 3 Write Leveling Delay** Rank 3 delay select for the write leveling (WL) LCDL
[23:16]	**R2WLD**	**Rank 2 Write Leveling Delay** Rank 2 delay select for the write leveling (WL) LCDL
[15:8]	**R1WLD**	**Rank 1 Write Leveling Delay** Rank 1 delay select for the write leveling (WL) LCDL
[7:0]	**R0WLD**	**Rank 0 Write Leveling Delay** Rank 0 delay select for the write leveling (WL) LCDL

DX0~8LCDLR1 Register (DX0~8LCDLR1)

Register	Offset	R/W	Description	Reset Value
DX0LCDLR1	DDRPHY_BA+0x1e4	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane0	0x0000_0000
DX1LCDLR1	DDRPHY_BA+0x224	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane1	0x0000_0000
DX2LCDLR1	DDRPHY_BA+0x264	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane2	0x0000_0000
DX3LCDLR1	DDRPHY_BA+0x2a4	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane3	0x0000_0000
DX4LCDLR1	DDRPHY_BA+0x2e4	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane4	0x0000_0000
DX5LCDLR1	DDRPHY_BA+0x324	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane5	0x0000_0000
DX6LCDLR1	DDRPHY_BA+0x364	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane6	0x0000_0000
DX7LCDLR1	DDRPHY_BA+0x3a4	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane7	0x0000_0000
DX8LCDLR1	DDRPHY_BA+0x3e4	R/W	DATX8 Bit Delay Line Register 1 for Byte Lane8	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RDQSND							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RDQSD							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WDQD							

Bits	Description	
[31:24]	**Reserved**	Return zeroes on reads.
[23:16]	**RDQSND**	**Read DQSN Delay for Type B/B1 PHY Only** Delay select for the read DQSN (RDQS) LCDL
[15:8]	**RDQSD**	**Read DQS Delay** Delay select for the read DQS (RDQS) LCDL
[7:0]	**WDQD**	**Write Data Delay** Delay select for the write data (WDQ) LCDL

DX0~8LCDLR2 Register (DX0~8LCDLR2)

Register	Offset	R/W	Description	Reset Value
DX0LCDLR2	DDRPHY_BA+0x1e8	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane0	0x0000_0000
DX1LCDLR2	DDRPHY_BA+0x228	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane1	0x0000_0000
DX2LCDLR2	DDRPHY_BA+0x268	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane2	0x0000_0000
DX3LCDLR2	DDRPHY_BA+0x2a8	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane3	0x0000_0000
DX4LCDLR2	DDRPHY_BA+0x2e8	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane4	0x0000_0000
DX5LCDLR2	DDRPHY_BA+0x328	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane5	0x0000_0000
DX6LCDLR2	DDRPHY_BA+0x368	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane6	0x0000_0000
DX7LCDLR2	DDRPHY_BA+0x3a8	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane7	0x0000_0000
DX8LCDLR2	DDRPHY_BA+0x3e8	R/W	DATX8 Bit Delay Line Register 2 for Byte Lane8	0x0000_0000

31	30	29	28	27	26	25	24
R3 DQSGD							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
R2 DQSGD							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
R1 DQSGD							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R0DQSGD							

Bits	Description	
[31:24]	**R3 DQSGD**	**Rank 3 Read DQS Gating Delay** Rank 3 delay select for the read DQS gating (DQSG) LCDL
[23:16]	**R2 DQSGD**	**Rank 2 Read DQS Gating Delay** Rank 2 delay select for the read DQS gating (DQSG) LCDL
[15:8]	**R1 DQSGD**	**Rank 1 Read DQS Gating Delay** Rank 1 delay select for the read DQS gating (DQSG) LCDL
[7:0]	**R0DQSGD**	**Rank 0 Read DQS Gating Delay** Rank 0 delay select for the read DQS gating (DQSG) LCDL

DX0~8MDLR Register (DX0~8MDLR)

Register	Offset	R/W	Description	Reset Value
DX0MDLR	DDRPHY_BA+0x1ec	R/W	DATX8 Master Delay Line Register for Byte Lane0	0x0000_0000
DX1MDLR	DDRPHY_BA+0x22c	R/W	DATX8 Master Delay Line Register for Byte Lane1	0x0000_0000
DX2MDLR	DDRPHY_BA+0x26c	R/W	DATX8 Master Delay Line Register for Byte Lane2	0x0000_0000
DX3MDLR	DDRPHY_BA+0x2ac	R/W	DATX8 Master Delay Line Register for Byte Lane3	0x0000_0000
DX4MDLR	DDRPHY_BA+0x2ec	R/W	DATX8 Master Delay Line Register for Byte Lane4	0x0000_0000
DX5MDLR	DDRPHY_BA+0x32c	R/W	DATX8 Master Delay Line Register for Byte Lane5	0x0000_0000
DX6MDLR	DDRPHY_BA+0x36c	R/W	DATX8 Master Delay Line Register for Byte Lane6	0x0000_0000
DX7MDLR	DDRPHY_BA+0x3ac	R/W	DATX8 Master Delay Line Register for Byte Lane7	0x0000_0000
DX8MDLR	DDRPHY_BA+0x3ec	R/W	DATX8 Master Delay Line Register for Byte Lane8	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
MDLD							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TPRD							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
IPRD							

Bits	Description	
[31:24]	**Reserved**	Return zeroes on reads.
[23:16]	**MDLD**	**MDL Delay** Delay select for the LCDL for the Master Delay Line.
[15:8]	**TPRD**	**Target Period** Target period measured by the master delay line calibration for VT drift compensation. This is the current measured value of the period and is continuously updated if the MDL is enabled to do so.
[7:0]	**IPRD**	**Initial Period** Initial period measured by the master delay line calibration for VT drift compensation. This value is used as the denominator when calculating the ratios of updates during VT compensation.

DX0~8GTR Register (DX0~8GTR)

Register	Offset	R/W	Description	Reset Value
DX0GTR	DDRPHY_BA+0x1f0	R/W	DATX8 General Timing Register for Byte Lane0	0x0000_5000
DX1GTR	DDRPHY_BA+0x230	R/W	DATX8 General Timing Register for Byte Lane1	0x0000_5000
DX2GTR	DDRPHY_BA+0x270	R/W	DATX8 General Timing Register for Byte Lane2	0x0000_0000
DX3GTR	DDRPHY_BA+0x2b0	R/W	DATX8 General Timing Register for Byte Lane3	0x0000_0000
DX4GTR	DDRPHY_BA+0x2f0	R/W	DATX8 General Timing Register for Byte Lane4	0x0000_0000
DX5GTR	DDRPHY_BA+0x330	R/W	DATX8 General Timing Register for Byte Lane5	0x0000_0000
DX6GTR	DDRPHY_BA+0x370	R/W	DATX8 General Timing Register for Byte Lane6	0x0000_0000
DX7GTR	DDRPHY_BA+0x3b0	R/W	DATX8 General Timing Register for Byte Lane7	0x0000_0000
DX8GTR	DDRPHY_BA+0x3f0	R/W	DATX8 General Timing Register for Byte Lane8	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				R3WLSL		R2WLSL	
15	14	13	12	11	10	9	8
R1WLSL		R0WLSL		R3DGSL			R2DGSL
7	6	5	4	3	2	1	0
R2DGSL		R1DGSL			R0DGSL		

Bits	Description	
[31:20]	**Reserved**	Return zeroes on reads.
[19:18]	**R3WLSL**	**Rank n Write Leveling System Latency** This is used to adjust the write latency after write leveling. Power-up default is 01 (i.e. no extra clock cycles required). The SL fields are initially set by the PUB during automatic write leveling but these values can be overwritten by a direct write to this register. Every two bits of this register control the latency of each of the (up to) four ranks. R0WLSL controls the latency of rank 0, R1WLSL controls rank 1, and so on. Valid values: 00 = Write latency = WL - 1. 01 = Write latency = WL. 10 = Write latency = WL + 1. 11 = Reserved.
[17:16]	**R2WLSL**	**Rank n Write Leveling System Latency** This is used to adjust the write latency after write leveling. Power-up default is 01 (i.e. no extra clock cycles required). The SL fields are initially set by the PUB during automatic write leveling but these values can be overwritten by a direct write to this register. Every two bits of this register control the latency of each of the (up to) four ranks. R0WLSL controls the latency of rank 0, R1WLSL controls rank 1, and so on. Valid values: 00 = Write latency = WL - 1. 01 = Write latency = WL. 10 = Write latency = WL + 1. 11 = Reserved.

[15:14]	**R1WLSL**	**Rank n Write Leveling System Latency** This is used to adjust the write latency after write leveling. Power-up default is 01 (i.e. no extra clock cycles required). The SL fields are initially set by the PUB during automatic write leveling but these values can be overwritten by a direct write to this register. Every two bits of this register control the latency of each of the (up to) four ranks. R0WLSL controls the latency of rank 0, R1WLSL controls rank 1, and so on. Valid values: 00 = Write latency = WL - 1. 01 = Write latency = WL. 10 = Write latency = WL + 1. 11 = Reserved.
[13:12]	**R0WLSL**	**Rank n Write Leveling System Latency** This is used to adjust the write latency after write leveling. Power-up default is 01 (i.e. no extra clock cycles required). The SL fields are initially set by the PUB during automatic write leveling but these values can be overwritten by a direct write to this register. Every two bits of this register control the latency of each of the (up to) four ranks. R0WLSL controls the latency of rank 0, R1WLSL controls rank 1, and so on. Valid values: 00 = Write latency = WL - 1. 01 = Write latency = WL. 10 = Write latency = WL + 1. 11 = Reserved.
[11:9]	**R3DGSL**	**Rank n DQS Gating System Latency** This is used to increase the number of clock cycles needed to expect valid DDR read data by up to seven extra clock cycles. This is used to compensate for board delays and other system delays. Power-up default is 000 (i.e. no extra clock cycles required). The SL fields are initially set by the PUB during automatic DQS data training but these values can be overwritten by a direct write to this register. Every three bits of this register control the latency of each of the (up to) four ranks. R0DGSL controls the latency of rank 0, R1DGSL controls rank 1, and so on. Valid values are 0 to 7.
[8:6]	**R2DGSL**	**Rank n DQS Gating System Latency** This is used to increase the number of clock cycles needed to expect valid DDR read data by up to seven extra clock cycles. This is used to compensate for board delays and other system delays. Power-up default is 000 (i.e. no extra clock cycles required). The SL fields are initially set by the PUB during automatic DQS data training but these values can be overwritten by a direct write to this register. Every three bits of this register control the latency of each of the (up to) four ranks. R0DGSL controls the latency of rank 0, R1DGSL controls rank 1, and so on. Valid values are 0 to 7.
[5:3]	**R1DGSL**	**Rank n DQS Gating System Latency** This is used to increase the number of clock cycles needed to expect valid DDR read data by up to seven extra clock cycles. This is used to compensate for board delays and other system delays. Power-up default is 000 (i.e. no extra clock cycles required). The SL fields are initially set by the PUB during automatic DQS data training but these values can be overwritten by a direct write to this register. Every three bits of this register control the latency of each of the (up to) four ranks. R0DGSL controls the latency of rank 0, R1DGSL controls rank 1, and so on. Valid values are 0 to 7.
[2:0]	**R0DGSL**	**Rank n DQS Gating System Latency** This is used to increase the number of clock cycles needed to expect valid DDR read data by up to seven extra clock cycles. This is used to compensate for board delays and other system delays. Power-up default is 000 (i.e. no extra clock cycles required). The SL fields are initially set by the PUB during automatic DQS data training but these values can be overwritten by a direct write to this register. Every three bits of this register control the latency of each of the (up to) four ranks. R0DGSL controls the latency of rank 0, R1DGSL controls rank 1, and so on. Valid values are 0 to 7.

6.13 One Time Programming Memory Controller (OTP)

6.13.1 Overview

The One Time Programming Memory Controller (OTP) has an embedded Synopsys Extra Permanent Memory (XPM). Data stored in XPM is Total Secure. The OTP controller realizes all functions of XPM memory including reading and programming function by 32 bits data format. The OTP controller provides fault tolerant mechanism to revise configuration or data, and provides the read only lock bit to protect configuration or data from being destroyed.

6.13.2 Features

- Supports 32 bits programming function and reading function
- Supports 8kbits Secure OTP memory
- Supports data retention more than 10 years
- Supports fault tolerant mechanism
- Supports read only lock bit
- Supports side-band handshaking signals with Key Store

6.13.3 Functional Description

6.13.3.1 Memory Organization

The OTP controller divides XPM memory into many blocks. Each memory block has its logic address range and access source.

Block Number	OTP Memory Block	OTP Logic Address Range	Access Source
1	Power-on Setting	0x100~0x103	OTP
2	DPM Setting	0x104~0x107	OTP
	PLM Setting	0x108~0x10B	OTP
3	MAC0 Address	0x10C~0x113	OTP
4	MAC1 Address	0x114~0x11B	OTP
5	Deployed Password	0x11C~0x11F	OTP
6	Secure Region	0x120~0x177	OTP
7	Non-secure Region	0x178~0x1CF	OTP
8	Hardware Unique Key 0 (HUK)	0x300~0x317	Key Store
9	Hardware Unique Key 1 (HUK)	0x318~0x32F	Key Store
10	Hardware Unique Key 2 (HUK)	0x330~0x347	Key Store
11	Key Storage 3	0x348~0x36F	Key Store
12	Key Storage 4	0x370~0x397	Key Store
13	Key Storage 5	0x398~0x3BF	Key Store
14	IBR ECC public key	0x3C0~0x3E7	Key Store
15	IBR ECC public key	0x3E8~0x40F	Key Store

16	IBR AES key	0x410~0x437	Key Store

Table 6.13-1 OTP Block Address Range and Access Path

Power-on Setting (OTP Logic Address = 0x100)

31	30	29	28	27	26	25	24
SECBTPSWD							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
MISCCFG		NPAGESEL		BTSRCSEL		BTSRCVOL	Reserved
7	6	5	4	3	2	1	0
Reserved		SD0BKEN	UR0DBGDIS	Reserved	WDT0ON	QSPI0CKSEL	PWRONSRC

Bits	Description	
[31:24]	**SECBTPSWD**	**Secure Boot Disable Password** If SECBTPSWD is 0x5A, the secure boot is Disabled. **Note 1:** SECBTPSWD didn't take effect and PG[0] used as Secure Boot Disable if PWRONSRC = 0. **Note 2:** In RMA mode, SECBTPSWD didn't take effect and PG[0] used as Secure Boot Disable.
[23:16]	**Reserved**	Reserved.
[15:14]	**MISCCFG**	**Miscellaneous Configuration** If BTSRCSEL = 01, boot from SD/eMMC. X0 = SD0/eMMC0 booting. (Default) X1 = SD1/eMMC1 booting. 0X = eMMC 4-bit mode booting. 1X = eMMC 8-bit mode booting. If BTSRCSEL = 10, boot from NAND Flash. 00 = Ignore (Default). 01 = ECC is BCH T12. 10 = ECC is BCH T24. 11 = No ECC. If BTSRCSEL = 00, boot from SPI Flash. 00 = SPI-NAND Flash with 1-bit mode booting (Default). 10 = SPI-NOR Flash with 1-bit mode booting. If BTSRCSEL = 11, boot from USB. X0 = over-current low active detect. X1 = over-current high active detect. **Note 1:** MISCCFG didn’t take effect and PG[7:6] used to indicate Miscellaneous Configuration if PWRONSRC = 0. **Note 2:** In RMA mode, MISCCFG didn't take effect and PG[7:6] used to indicate Miscellaneous Configuration. **Note 3:** X is don’t care.
[13:12]	**NPAGESEL**	**NAND Flash Page Size Selection** If BTSRCSEL = 10, boot from NAND Flash, these two bits indicates NAND Flash Page Size Selection. 00 = Ignore. (Default)

		01 = NAND Flash page size is 2kB. 10 = NAND Flash page size is 4kB. 11 = NAND Flash page size is 8kB. If BTSRCSEL = 11, boot from USB, these two bits indicates USB Role and Port Selection. 00 = USBD booting. 01 = USBH port 0 boot. 10 = USBD booting. 11 = USBH port 1 boot. **Note 1:** NPAGESEL didn't take effect and PG[5:4] used as Page Size Selection if PWRONSRC = 0. **Note 2:** In RMA mode, NPAGESEL didn't take effect and PG[5:4] used as Page Size Selection.
[11:10]	**BTSRCSEL**	**Boot Source Selection** 00 = Boot from SPI Flash (Default). 01 = Boot from SD/eMMC. 10 = Boot from NAND Flash. 11 = Boot from USB. **Note 1:** BTSRCSEL didn't take effect and PG[3:2] used as Boot Source Selection if PWRONSEC = 0. **Note 2:** In RMA mode, BTSRCSEL didn't take effect and PG[3:2] used as Boot Source Selection.
[9]	**BTSRCVOL**	**Boot Source Interface I/O Voltage** 0 = Boot source interface I/O voltage is 3.3V. 1 = Boot source interface I/O voltage is 1.8V. **Note 1:** BTSRCVOL didn't take effect and PG[1] used as Boot Source Interface I/O Voltage if PWRONSEC = 0.
[8:6]	**Reserved**	Reserved.
[5]	**SD0BKEN**	**SD0 Back Up Boot Enable Bit** 0 = SD0 back up boot Disabled (Default). 1 = SD0 back up boot Enabled. **Note:** SD0BKEN didn't take effect if BTSRCSEL = 01 and MISCCFG = 00.
[4]	**UR0DBGDIS**	**UART0 Debug Message Output Disable Bit** 0 = UART0 debug message output Enabled (Default). 1 = UART0 debug message output Disabled.
[3]	**Reserved**	Reserved.
[2]	**WDT0ON**	**Watchdog Timer 0 ON/OFF Selection** 0 = WDT0 function Disabled (Default). 1 = WDT0 function Enabled.
[1]	**QSPI0CKSEL**	**QSPI0_CLK Frequency Selection** 0 = QSPI0_CLK frequency is 30 MHz (Default). 1 = QSPI0_CLK frequency is 50 MHz.
[0]	**PWRONSRC**	**Power on Setting Source Control** 0 = Power on setting values come from pin. (Default) 1 = Power on setting values come from OTP. **Note 1:** Which power on setting bit affected by this bit defined in bit description. **Note 2:** In RMA mode, PWRONSRC didn't take effect and power-on setting always comes from pin.

DPM Setting (OTP Logic Address = 0x104)

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved						GICCFGS_LOCK	GICCFGS_DIS
15	14	13	12	11	10	9	8
EXTT_LOCK	EXTT_DIS	EXT_LOCK	EXT_DIS	M4NDS_LOCK	M4NDS_DIS	M4DS_LOCK	M4DS_DIS
7	6	5	4	3	2	1	0
A35NSNDS_LOCK	A35NSNDS_DIS	A35NSDS_LOCK	A35NSDS_DIS	A35SNDS_LOCK	A35SNDS_DIS	A35SDS_LOCK	A35SDS_DIS

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17]	**GICCFGS_LOCK**	DPM GIC CFGSDISABLE Debug State LOCK OTP Bit
[16]	**GICCFGS_DIS**	DPM GIC CFGSDISABLE Debug State DISABLE OTP Bit
[15]	**EXTT_LOCK**	DPM External Tracing Debug State LOCK OTP Bit
[14]	**EXTT_DIS**	DPM External Tracing Debug State DISABLE OTP Bit
[13]	**EXT_LOCK**	DPM External Debug State LOCK OTP Bit
[12]	**EXT_DIS**	DPM External Debug State DISABLE OTP Bit
[11]	**M4NDS_LOCK**	DPM M4 Non-invasive Debug State LOCK OTP Bit
[10]	**M4NDS_DIS**	DPM M4 Non-invasive Debug State DISABLE OTP Bit
[9]	**M4DS_LOCK**	DPM M4 Debug State LOCK OTP Bit
[8]	**M4DS_DIS**	DPM M4 Debug State DISABLE OTP Bit
[7]	**A35NSNDS_LOCK**	DPM A35 Non-secure Non-invasive Debug State LOCK OTP Bit
[6]	**A35NSNDS_DIS**	DPM A35 Non-secure Non-invasive Debug State DISABLE OTP Bit
[5]	**A35NSDS_LOCK**	DPM A35 Non-secure Debug State LOCK OTP Bit
[4]	**A35NSDS_DIS**	DPM A35 Non-secure Debug State DISABLE OTP Bit
[3]	**A35SNDS_LOCK**	DPM A35 Secure Non-invasive Debug State LOCK OTP Bit
[2]	**A35SNDS_DIS**	DPM A35 Secure Non-invasive Debug State DISABLE OTP Bit
[1]	**A35SDS_LOCK**	DPM A35 Secure Debug State LOCK OTP Bit
[0]	**A35SDS_DIS**	DPM A35 Secure Debug State DISABLE OTP Bit

PLM Setting (OTP Logic Address = 0x108)

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					RMA	Deployed	OEM

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**RMA**	**RMA Stage** CA35 External Debug – Controlled by DPM CM4 Debug – Controlled by DPM WDT0 Disabled Keystore – All OTP keys revoked. Boot Source – Default setting, regardless of OTP setting Secure boot – Disabled
[1]	**Deployed**	**Deployed Stage** CA35 External Debug – Disabled CM4 Debug – Disabled WDT0 Enable – Controlled by OTP bit Boot Source – Controlled by OTP bit or by default Secure boot – Controlled by OTP bit or by default
[0]	**OEM**	**OEM Stage** CA35 External Debug – Controlled by DPM CM4 Debug – Controlled by DPM WDT0 Enable – Controlled by OTP bit Boot Source – Controlled by OTP bit or by default Secure boot – Controlled by OTP bit or by default

6.14 External Bus Interface (EBI)

6.14.1 Overview

This chip is equipped with an external bus interface (EBI) for external device use. To save the connections between an external device and a chip, EBI is operating at address bus and data bus multiplex mode. The EBI supports three chip selects that can connect three external devices with different timing setting requirements.

6.14.2 Features

- Supports up to three memory banks
- Supports dedicated external chip select pin with polarity control for each bank
- Supports accessible space up to 1 Mbytes for each bank, actually external addressable space is dependent on package pin out
- Supports 8-/16-bit data width
- Supports byte write in 16-bit data width mode
- Supports address bus and data bus multiplexed mode
- Supports address bus and data bus separate mode
- Supports Timing parameters individual adjustment for each memory block
- Supports LCD interface i80 mode
- Supports PDMA mode
- Supports variable external bus base clock (MCLK) which based on HCLK
- Supports configurable idle cycle for different access condition: Idle of Write command finish (W2X) and Idle of Read-to-Read (R2R)

6.14.3 Block Diagram

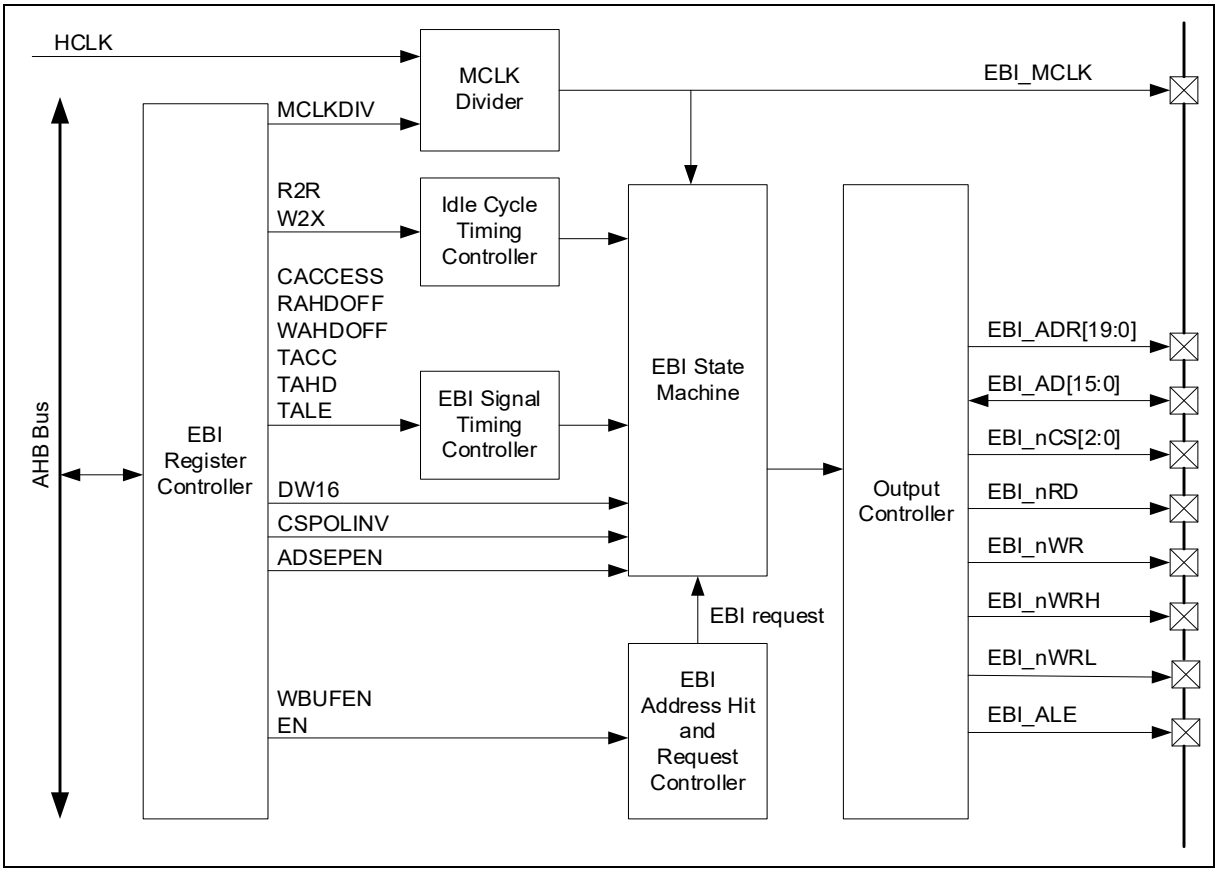


Figure 6.14-1 EBI Block Diagram

6.14.4 Basic Configuration

- Clock Source Configuration
  - Enable EBI controller clock in EBICKEN (CLK_SYSCLK1[7]).
- Reset Configuration
  - Reset EBI controller in EBIRST (SYS_IPRST0[19]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
	EBI_AD0	PL.12	MFP8
		PA.0	MFP7
		PG.11	MFP10
		PK.9	MFP8
	EBI_AD1	PL.13	MFP8
		PA.1	MFP7
		PG.12	MFP10
		PK.10	MFP8
	EBI_AD2	PL.14	MFP8

		PA.2	MFP7
		PG.13	MFP10
		PK.11	MFP8
	EBI_AD3	PL.15	MFP8
		PA.3	MFP7
		PG.14	MFP10
		PM.0	MFP8
	EBI_AD4	PA.4	MFP7
		PD.12	MFP8
		PG.15	MFP10
		PM.1	MFP8
	EBI_AD5	PA.5	MFP7
		PD.13	MFP8
		PL.6	MFP8
		PD.6	MFP9
		PM.2	MFP8
	EBI_AD6	PA.6	MFP7
		PD.14	MFP8
		PL.7	MFP8
		PD.7	MFP9
		PM.3	MFP8
	EBI_AD7	PA.7	MFP7
		PD.15	MFP8
		PL.8	MFP8
		PD.8	MFP9
		PM.4	MFP8
		PJ.0	MFP12
		PG.8	MFP12
	EBI_AD8	PA.8	MFP7
		PM.12	MFP8
		PL.9	MFP8
		PD.9	MFP9
		PM.5	MFP8
		PJ.1	MFP12
		PG.9	MFP12

	EBI_AD9	PA.9	MFP7
		PM.13	MFP8
		PL.10	MFP8
		PD.10	MFP9
		PM.6	MFP8
		PJ.2	MFP12
		PG.10	MFP12
	EBI_AD10	PA.10	MFP7
		PM.14	MFP8
		PL.11	MFP8
		PD.11	MFP9
		PM.7	MFP8
		PJ.3	MFP12
		PK.4	MFP12
	EBI_AD11	PL.12	MFP13
		PA.11	MFP7
		PM.15	MFP8
		PL.0	MFP8
		PM.8	MFP8
		PJ.0	MFP9
		PI.8	MFP12
	EBI_AD12	PL.13	MFP13
		PA.12	MFP7
		PL.1	MFP8
		PM.9	MFP8
		PJ.1	MFP9
		PI.9	MFP12
	EBI_AD13	PL.14	MFP13
		PA.13	MFP7
		PL.2	MFP8
		PM.10	MFP8
		PJ.2	MFP9
		PI.10	MFP12
		PB.9	MFP10
	EBI_AD14	PL.15	MFP13

		PA.14	MFP7
		PL.3	MFP8
		PM.11	MFP8
		PJ.3	MFP9
		PI.11	MFP12
		PB.10	MFP10
	EBI_AD15	PG.0	MFP7
		PK.8	MFP8
	EBI_ADR0	PA.0	MFP9
		PI.0	MFP8
		PK.9	MFP10
	EBI_ADR1	PA.1	MFP9
		PI.1	MFP8
		PK.10	MFP10
	EBI_ADR2	PA.2	MFP9
		PI.2	MFP8
		PK.11	MFP10
	EBI_ADR3	PA.3	MFP9
		PI.3	MFP8
		PM.0	MFP10
	EBI_ADR4	PA.4	MFP9
		PI.4	MFP8
		PM.1	MFP10
	EBI_ADR5	PA.5	MFP9
		PI.5	MFP8
		PM.2	MFP10
	EBI_ADR6	PA.6	MFP9
		PI.6	MFP8
		PM.3	MFP10
	EBI_ADR7	PA.7	MFP9
		PI.7	MFP8
		PM.4	MFP10
	EBI_ADR8	PA.8	MFP9
		PK.0	MFP8
		PM.5	MFP10

	EBI_ADR9	PA.9	MFP9
		PK.1	MFP8
		PM.6	MFP10
	EBI_ADR10	PA.10	MFP9
		PK.2	MFP8
		PM.7	MFP10
	EBI_ADR11	PA.11	MFP9
		PK.3	MFP8
		PM.8	MFP10
	EBI_ADR12	PA.12	MFP9
		PJ.12	MFP8
		PM.9	MFP10
	EBI_ADR13	PA.13	MFP9
		PJ.13	MFP8
		PM.10	MFP10
	EBI_ADR14	PA.14	MFP9
		PJ.14	MFP8
		PM.11	MFP10
	EBI_ADR15	PG.0	MFP15
		PJ.15	MFP8
		PK.8	MFP10
		PB.10	MFP9
	EBI_ADR16	PG.2	MFP7
		PJ.0	MFP10
		PB.12	MFP9
	EBI_ADR17	PG.3	MFP7
		PJ.1	MFP10
		PB.13	MFP9
	EBI_ADR18	PB.14	MFP9
		PG.4	MFP7
		PJ.2	MFP10
	EBI_ADR19	PB.15	MFP9
		PG.5	MFP7
		PJ.3	MFP10
	EBI_ALE	PD.15	MFP7

		PA.15	MFP7
		PG.5	MFP8
		PG.14	MFP7
		PB.9	MFP9
		PB.11	MFP10
	EBI_MCLK	PD.14	MFP7
		PG.3	MFP9
		PG.13	MFP7
		PK.8	MFP9
		PB.10	MFP6
	EBI_nCS0	PG.1	MFP7
		PG.4	MFP8
		PG.15	MFP7
		PJ.0	MFP11
		PG.8	MFP13
		PB.8	MFP9
	EBI_nCS1	PD.12	MFP7
		PG.3	MFP8
		PG.11	MFP8
		PJ.1	MFP11
		PG.9	MFP13
	EBI_nCS2	PD.13	MFP7
		PG.2	MFP8
		PB.11	MFP9
	EBI_nRD	PG.6	MFP7
		PL.4	MFP8
	EBI_nWR	PG.7	MFP7
		PL.5	MFP8
	EBI_nWRH	PG.11	MFP7
		PL.10	MFP11
		PJ.2	MFP11
		PG.10	MFP13
	EBI_nWRL	PG.12	MFP7
		PL.11	MFP11
		PJ.3	MFP11

		PK.4	MFP13

6.14.5 Functional Description

6.14.5.1 EBI Area and Address Hit

The EBI mapping address is located at 0x6800_0000 ~ 0x682F_FFFF and the total memory space is 3 Mbytes. When system request address hits EBI's memory space, the corresponding EBI chip select signal is assert and EBI state machine operates.

Chip Select	Address Mapping
EBI_nCS0	0x6800_0000 ~ 0x680F_FFFF
EBI_nCS1	0x6810_0000 ~ 0x681F_FFFF
EBI_nCS2	0x6820_0000 ~ 0x682F_FFFF

Table 6.14-1 EBI Address Mapping

To map the whole EBI memory space, it requires 20-bit address for 8-bit data width device and 19-bit address for 16-bit data width device. For package that output less than 20-bit address, EBI will map device to mirror space. For example, the package with 18-bit EBI address, EBI will mapped external device (for Bank0/EBI_nCS0) to 0x6800_0000 ~ 0x6803_FFFF, 0x6804_0000 ~ 0x6807_FFFF, 0x6808_0000 ~ 0x680B_FFFF and 0x680C_0000 ~ 0x680F_FFFF simultaneously.

6.14.5.2 EBI Data Width Connection - Address Bus and Data Bus Multiplex Mode

The EBI supports the device whose address bus and data bus are multiplexed. For the external device with separated address and data bus, the connection to device needs additional latch device to latch the address. In this case, the pin EBI_ALE is connected to the latch device to latch the address value. Pin EBI_AD is the input of the latch device, and the output of the latch device is connected to the address of external device.

For 16-bit device, the EBI_AD [15:0] is shared by address and 16-bit data, and EBI_ADR [18:16] is dedicated for address and could be connected to 16-bit device directly. The EBI_ADR[19] will be ignored when EBI data width is set as 16-bit width. For 8-bit device, only EBI_AD [7:0] is shared by address and 8-bit data, EBI_AD[15:8] and EBI_ADR[19:16] are dedicated for address and could be connected to 8-bit device directly. Figure 6.14-2 shows the connection of 16-bit data width device and Figure 6.14-3 shows the connection of 8-bit data width device.

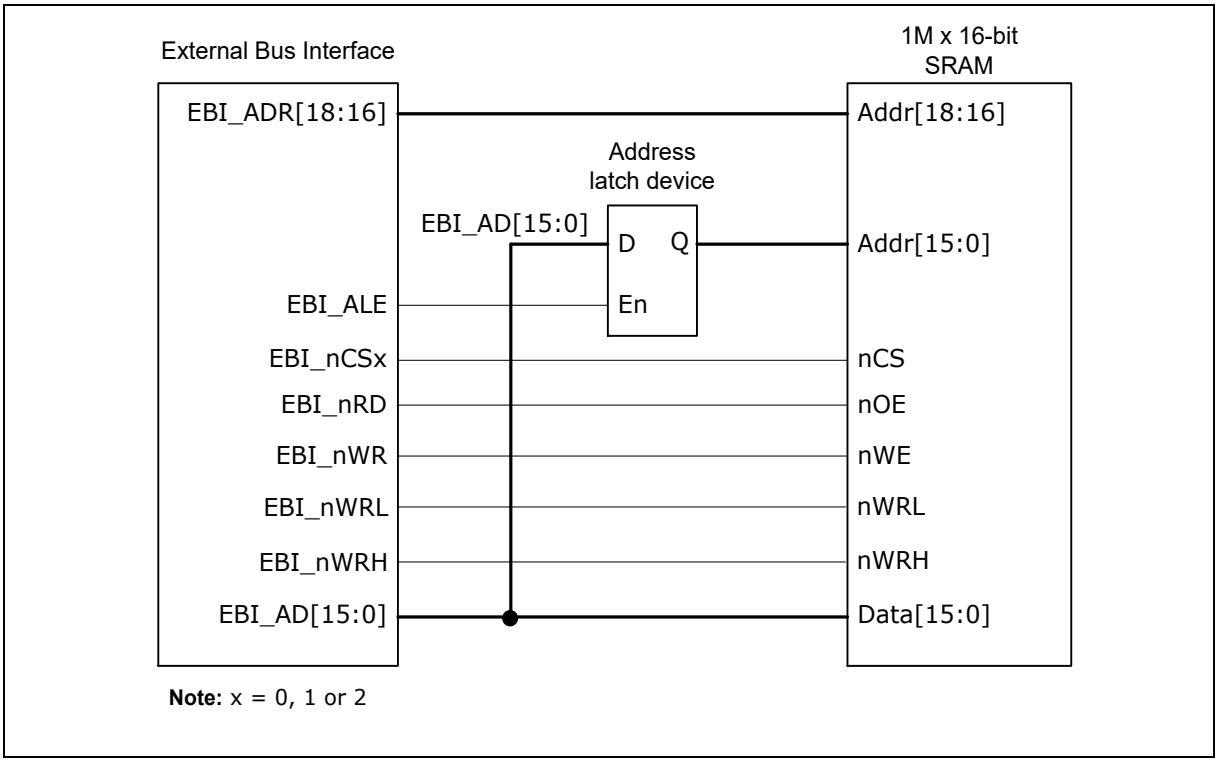


Figure 6.14-2 Connection of 16-bit EBI Data Width with 16-bit Device

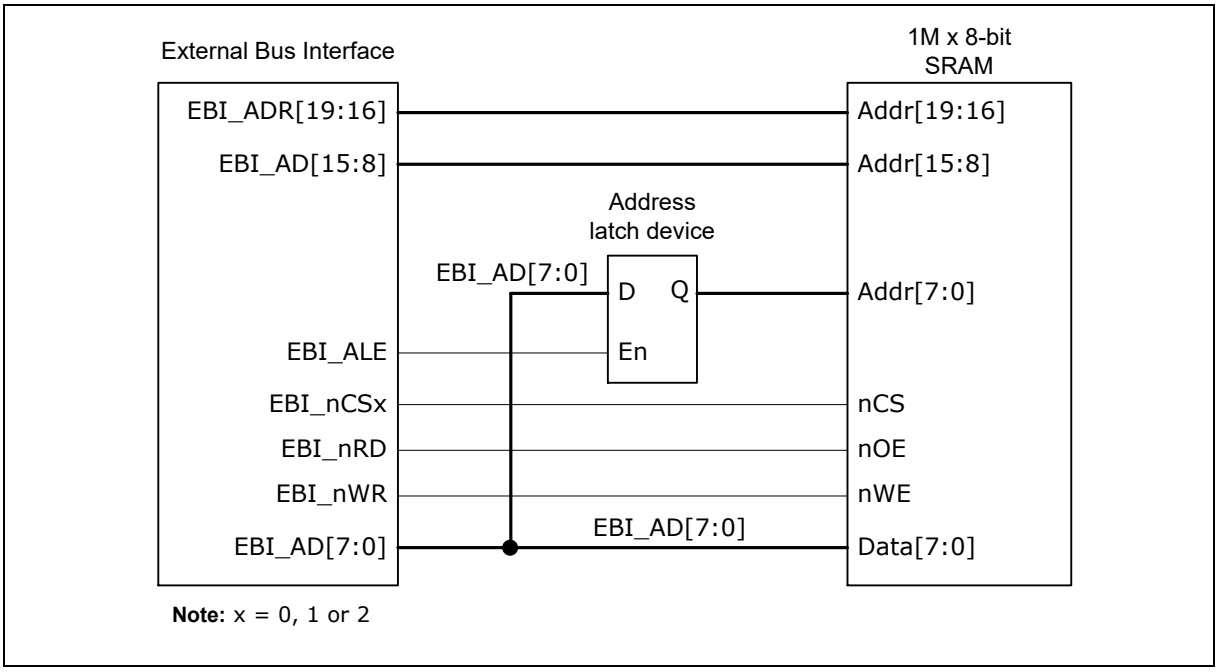


Figure 6.14-3 Connection of 8-bit EBI Data Width with 8-bit Device

When system access data width is larger than EBI data width, the EBI controller will finish a system access command by operating EBI access more than once. For example, if system requests a 32-bit data through EBI device, the EBI controller will operate accessing four times when setting EBI data width with 8-bit.

6.14.5.3 EBI Data Width Connection - Address Bus and Data Bus Separate Mode

The EBI supports address and data bus separate mode. User can enable this mode by setting ADSEPEN (EBI_CTLx[3]). When separate mode is enabled, EBI_AD is dedicated for data bus and connected directly to device data bus, EBI_ADR is dedicated for address bus.

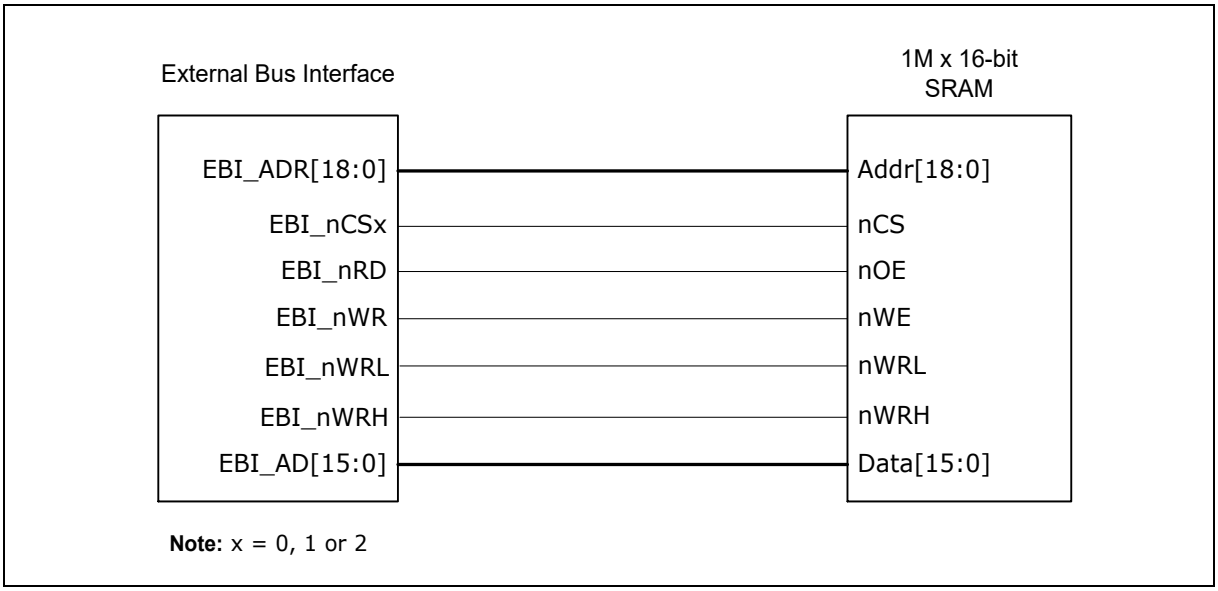


Figure 6.14-4 Connection of 16-bit EBI Data Width with 16-bit Device in Separate mode

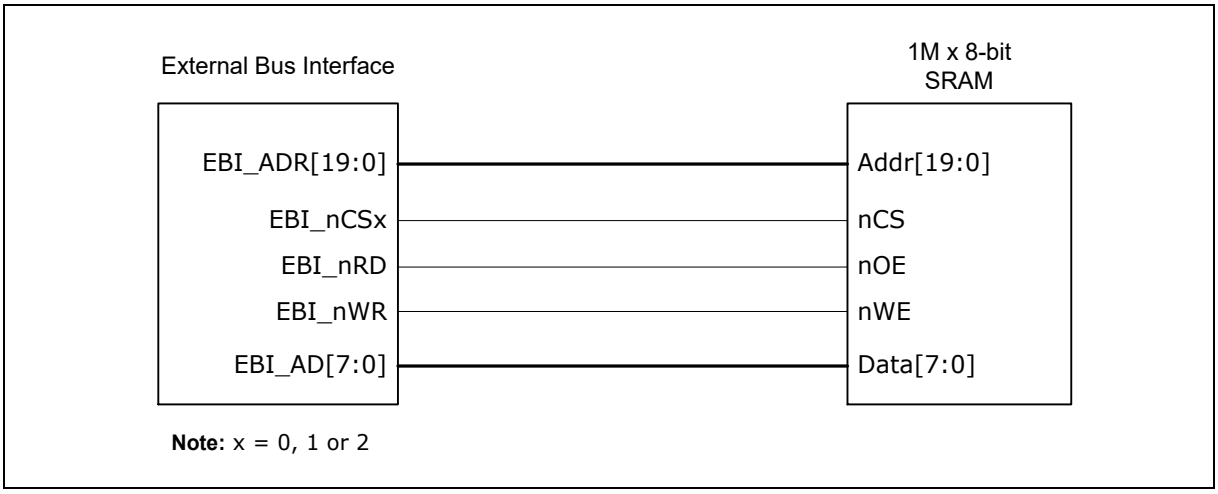


Figure 6.14-5 Connection of 8-bit EBI Data Width with 8-bit Device in Separate mode

6.14.5.4 EBI Operating Control

MCLK Control

In the chip, all EBI signals will be synchronized by EBI_MCLK when EBI is operating. When chip connects to the external device with slower operating frequency, the EBI_MCLK can divide most to HCLK/128 by setting MCLKDIV (EBI_CTLx[10:8]). Therefore, chip can be suitable for a wide frequency range of EBI device. If EBI_MCLK is set to HCLK/1, EBI signals are synchronized by positive edge of EBI_MCLK, else by negative edge of EBI_MCLK.

Operation and Access Timing Control

At the start of EBI access, chip select (EBI_nCS0, EBI_nCS1 and EBI_nCS2) asserts to low and wait one EBI_MCLK for address setup time (tASU) for address stable. Then EBI_ALE asserts to high after address is stable and keeps for a period of time (tALE) for address latch. After latch address, EBI_ALE asserts to low and wait one EBI_MCLK for latch hold time (tLHD) and another one EBI_MCLK cycle (tA2D) that is inserted behind address hold time to be the bus turn-around time for address change to data. Then EBI_nRD asserts to low when read access or EBI_nWR asserts to low when write access. Then EBI_nRD or EBI_nWR asserts to high after keeps access time (tACC) for reading output stable or writing finish. After that, EBI signals keep for data access hold time (tAHD) and chip select asserts to high, address is released by current access control.

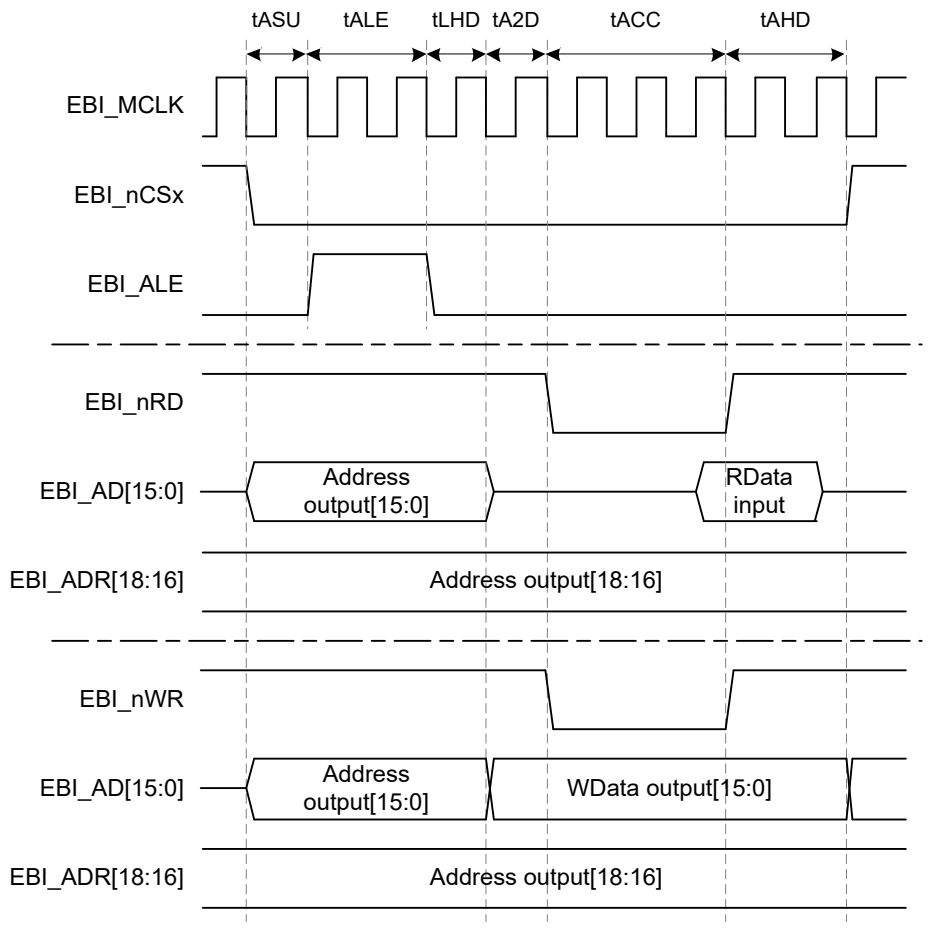
The EBI controller provides a flexible timing control for different external device. In EBI timing control, tASU, tLHD and tA2D are fixed to 1 EBI_MCLK cycle, tAHD can modulate to 1~8 EBI_MCLK cycles by setting TAHD (EBI_TCTLx[10:8]), tACC can modulate to 1~32 EBI_MCLK cycles by setting TACC (EBI_TCTLx[7:3]), and tALE can modulate to 1~8 EBI_MCLK cycles by setting TALE (EBI_CTL0[18:16]). Some external device can support zero data access hold time accessing, the EBI controller can skipped tAHD to increase access speed by setting WAHDOFF (EBI_TCTLx[23]) and RAHDOFF (EBI_TCTLx[22]).

For each chip select, the EBI provides individual register with timing control except that tALE can only be controlled by EBI_CTL0.

Parameter	Value	Unit	Description
tASU	1	MCLK	Address Latch Setup Time.
tALE	1 ~ 8	MCLK	ALE High Period. Controlled by TALE (EBI_CTL0[18:16]).
tLHD	1	MCLK	Address Latch Hold Time.
tA2D	1	MCLK	Address To Data Delay (Bus Turn-Around Time).
tACC	1 ~ 32	MCLK	Data Access Time. Controlled by TACC (EBI_TCTLx[7:3]).
tAHD	1 ~ 8	MCLK	Data Access Hold Time. Controlled by TAHD (EBI_TCTLx[10:8]).
IDLE	0 ~ 15	MCLK	Idle Cycle. Controlled by R2R (EBI_TCTLx[27:24]) and W2X (EBI_TCTLx[15:12]).

Table 6.14-3 Timing Control Parameter

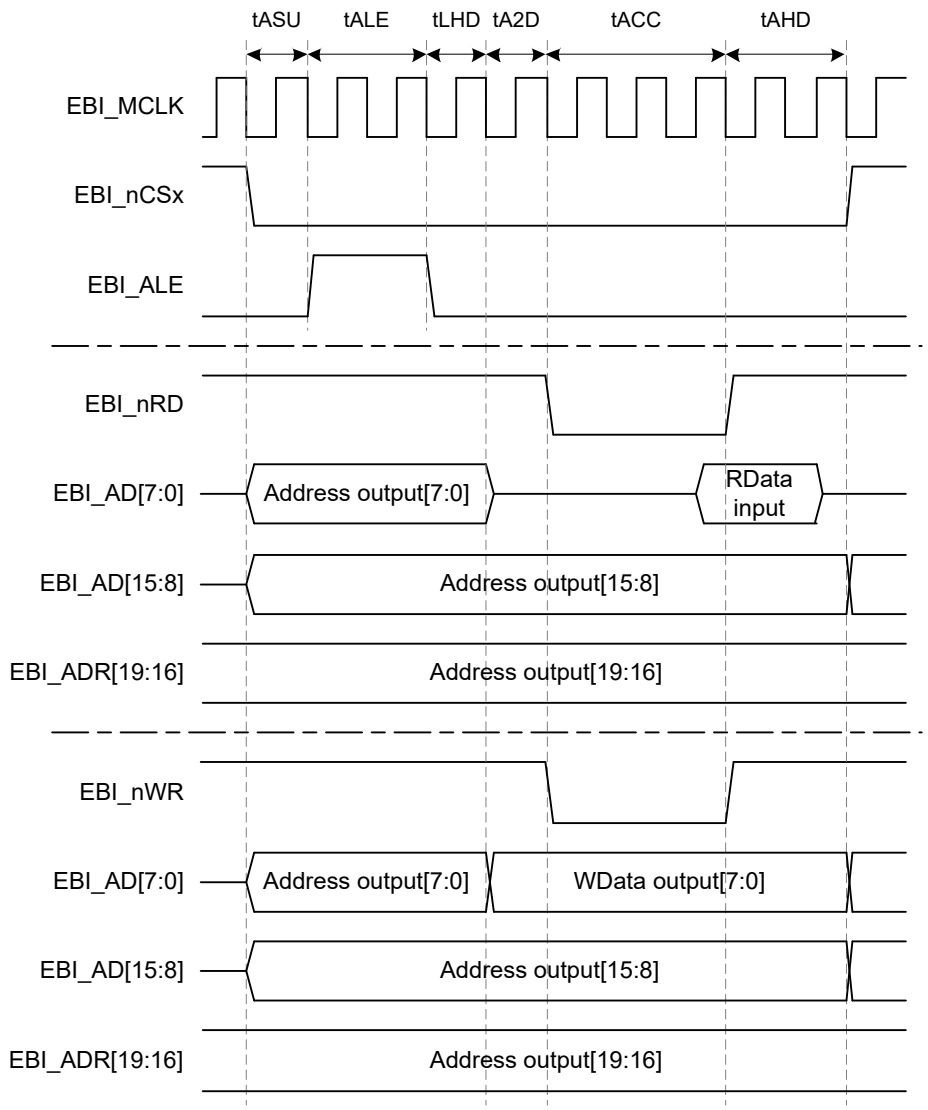
Figure 6.14-6 shows an example of setting 16-bit data width. In this example, EBI_AD bus is used for being address [15:0] and data [15:0]. When EBI_ALE assert to high, EBI_AD is address output. After address is latched, EBI_ALE asserts to low and the EBI_AD bus change to high impedance to wait device output data in read access operation, or it is used for being write data output.



Note1: The EBI_MCLK is HCLK/2, MCLKDIV (EBI_CTLx[10:8]) = 1
Note2: x = 0, 1 or 2

Figure 6.14-6 Timing Control Waveform for 16-bit Data Width

Figure 6.14-7 shows an example of setting 8-bit data width. The difference between 8-bit and 16-bit data width is EBI_AD[15:8]. In 8-bit data width setting, EBI_AD[15:8] is always Address [15:8] output so that external latch needs only 8-bit width.

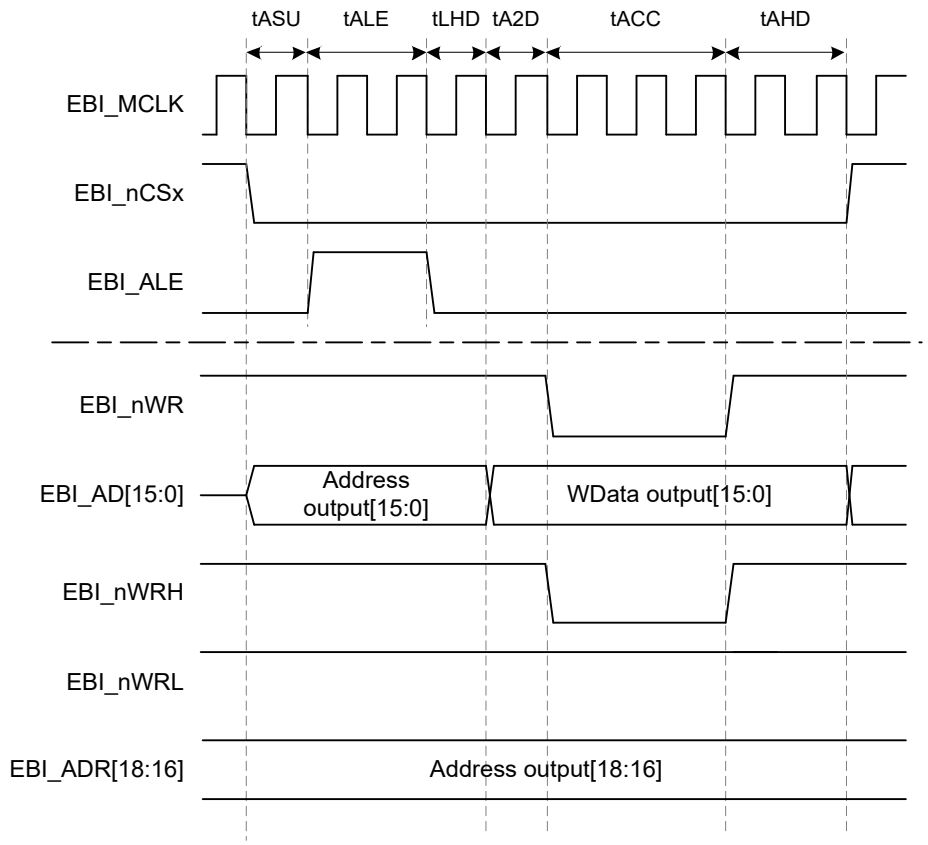


Note1: The EBI_MCLK is HCLK/2, MCLKDIV (EBI_CTLx[10:8]) = 1
Note2: x = 0, 1 or 2

Figure 6.14-7 Timing Control Waveform for 8-bit Data Width

Byte Access

The EBI supports byte access when connected to 16-bit device. The pin EBI_nWRH and EBI_nWRL assertion indicate high byte enable and low byte enable in 16-bit data bus. Figure 6.14-8 shows the write operation of 8-bit width data in EBI_AD[15:8] with EBI_nWRH assertion.



Note1: The EBI_MCLK is HCLK/2, MCLKDIV (EBI_CTLx[10:8]) = 1
Note2: x = 0, 1 or 2

Figure 6.14-8 Timing Control Waveform for Byte Write in 16-bit Data Mode

Insert Idle Cycle

When EBI accesses continuously, there may occur bus conflict if the device access time is much slow with system operating. The EBI controller supplies additional idle cycle to solve this problem. During idle cycle, all control signals of EBI are inactive. Figure 6.14-9 shows idle cycles.

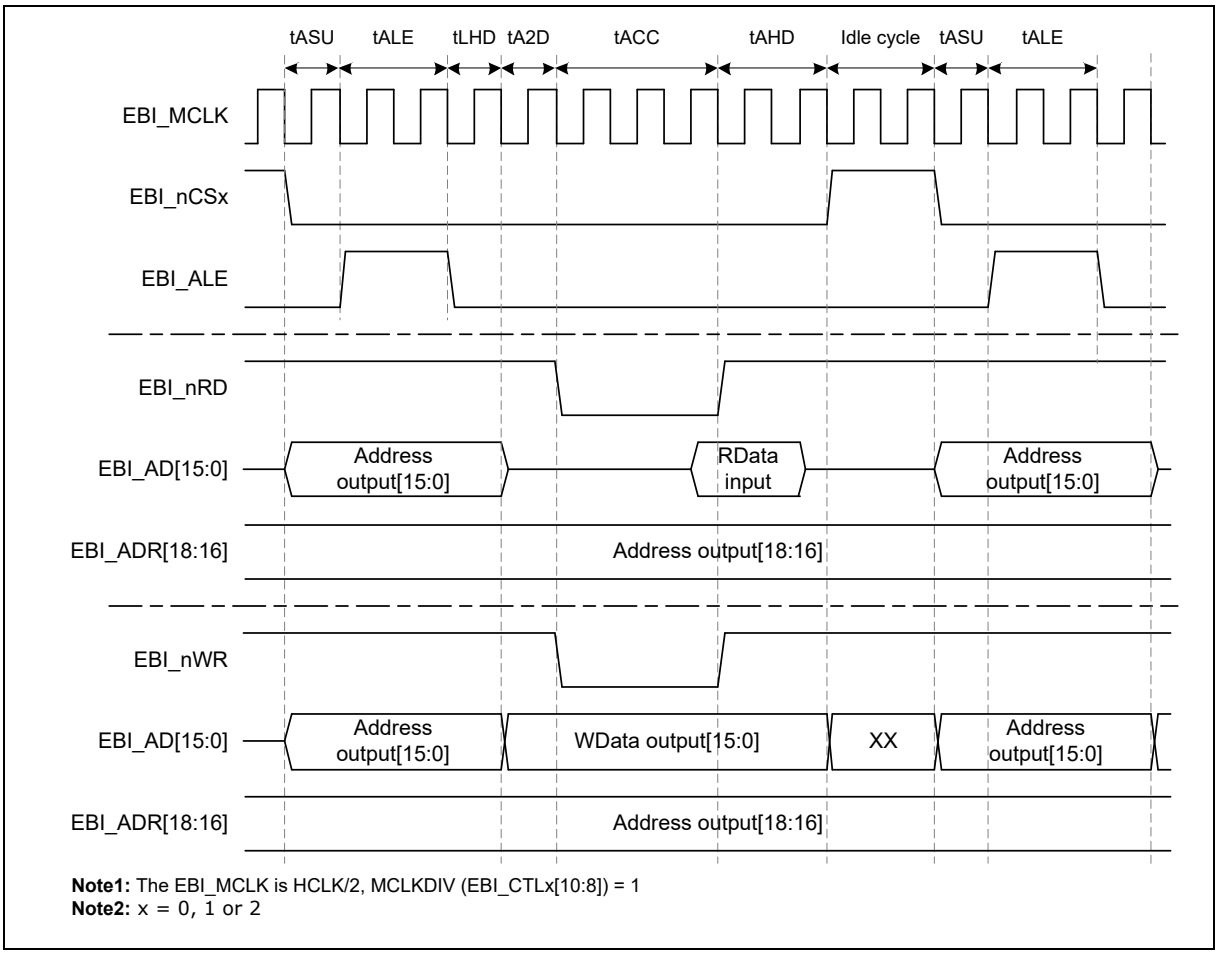


Figure 6.14-9 Timing Control Waveform for Insert Idle Cycle

There are two conditions that EBI can insert idle cycle by timing control:

1. After write access
2. After read access and before next read access (R2R idle cycle)

By setting W2X (EBI_TCTLx[15:12]), and R2R (EBI_TCTLx[27:24]), the time of idle cycle can be specified from 0~15 EBI_MCLK.

Chip Select Polarity Control

The EBI supports chip select polarity control for connecting to variable external device. When CSPOLINV (EBI_CTLx[2]) is set to 0, the chip select pins (EBI_nCSx) works as low active behavior. It means the external device can be access under EBI_nCSx at low state. When CSPOLINV (EBI_CTLx[2]) is set to 1, the chip select pin (EBI_nCS) works as high active behavior. It means the external device can be access under EBI_nCSx at high state.

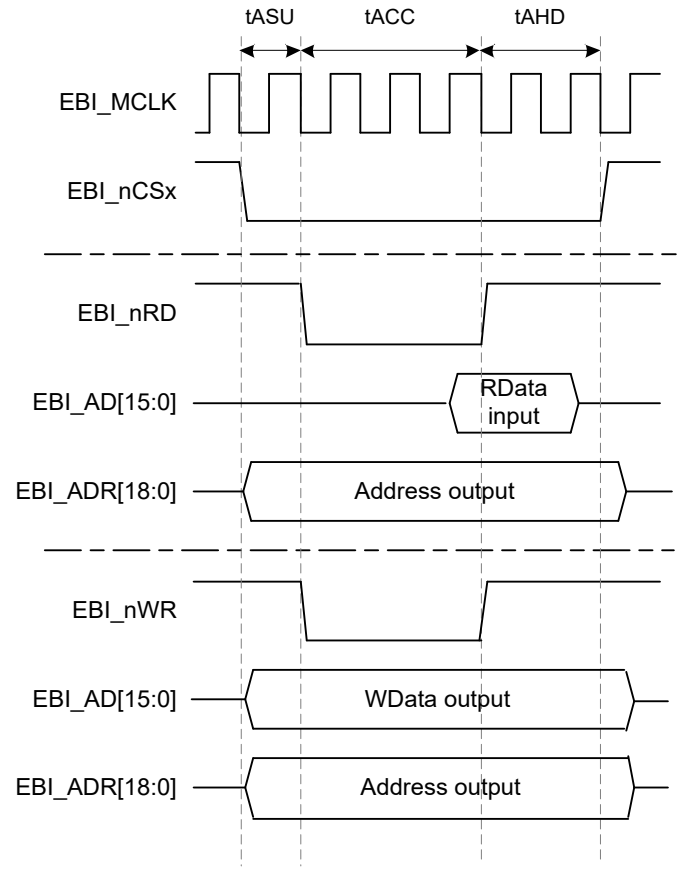
Write Buffer

When user writes data to an external device through EBI bus, the EBI controller will start processing the write action immediately and the CPU is held until current EBI write action is finished. User can enable write buffer function to improve CPU and EBI access performance. When EBI write buffer function is enabled, the CPU can continuously execute other instruction during EBI controller process the write action to external device. There is one exception condition for this case. If CPU executes another data access through EBI bus when EBI process write action, the CPU will be held.

User can enable write buffer by setting WBUFEN (EBI_CTL0[24]).

Address Data Separate Mode

When EBI is set as separate mode, the tALE, tLHD, tA2D cycles are ignored. EBI_AD and EBI_ADR are dedicated for data and address bus separately.



Note1: The EBI_MCLK is HCLK/2, MCLKDIV (EBI_CTLx[10:8]) = 1
Note2: x = 0, 1 or 2

Figure 6.14-10 Timing Control Waveform for 16-bit Data Width for Separate Mode

Continuous Data Access Mode

The EBI supports continuous data access mode for the device which needs faster data access and do not need address control interface. User can enable this mode by setting CACCESS (EBI_CTLx[4]) for each bank. When EBI set as continuous data access mode, the tASU, tALE, tLHD cycles are ignored and EBI can access data continuously within one read or write command. There will be dummy cycle

between each access command. The timing waveform is shown as Figure 6.14-11.

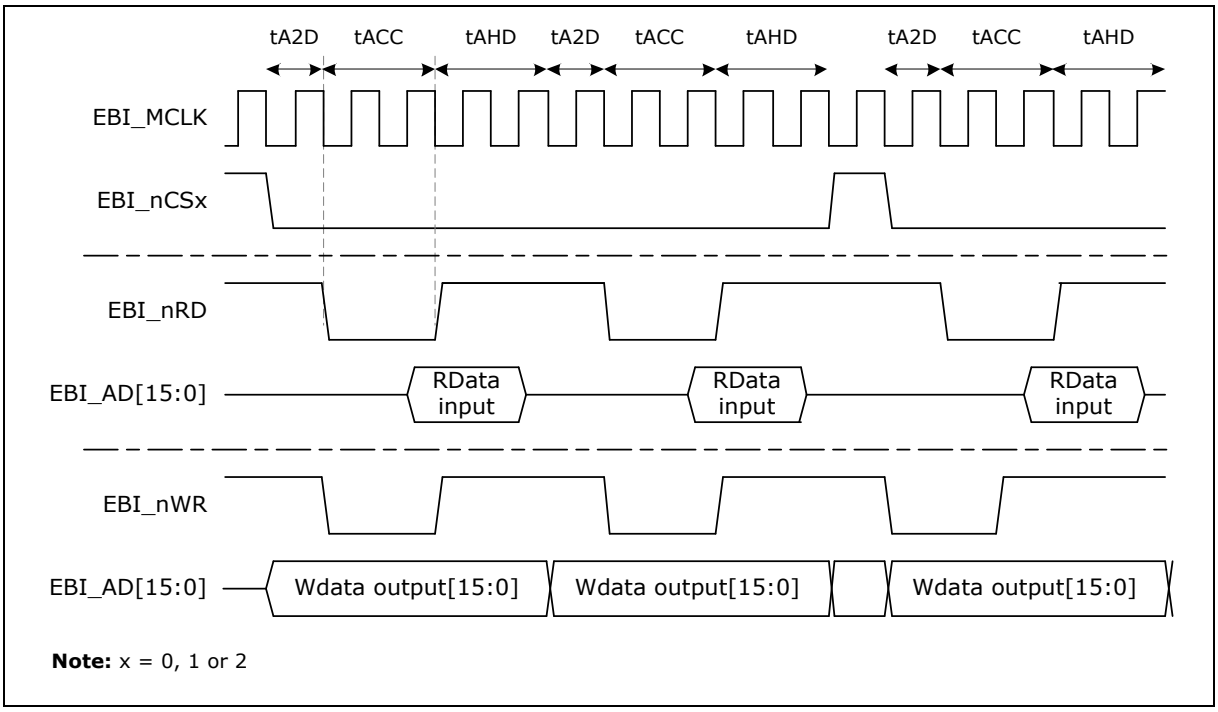


Figure 6.14-11 Timing Control Waveform for Continuous Data Access Mode

6.14.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
EBI Base Address: **EBI_BA = 0x4010_0000**				
EBI_CTL0	EBI_BA+0x00	R/W	External Bus Interface Bank0 Control Register	0x0000_0000
EBI_TCTL0	EBI_BA+0x04	R/W	External Bus Interface Bank0 Timing Control Register	0x0000_0000
EBI_CTL1	EBI_BA+0x10	R/W	External Bus Interface Bank1 Control Register	0x0000_0000
EBI_TCTL1	EBI_BA+0x14	R/W	External Bus Interface Bank1 Timing Control Register	0x0000_0000
EBI_CTL2	EBI_BA+0x20	R/W	External Bus Interface Bank2 Control Register	0x0000_0000
EBI_TCTL2	EBI_BA+0x24	R/W	External Bus Interface Bank2 Timing Control Register	0x0000_0000

6.14.7 Register Description

External Bus Interface Control Register (EBI_CTLx)

Register	Offset	R/W	Description	Reset Value
EBI_CTL0	EBI_BA+0x00	R/W	External Bus Interface Bank0 Control Register	0x0000_0000
EBI_CTL1	EBI_BA+0x10	R/W	External Bus Interface Bank1 Control Register	0x0000_0000
EBI_CTL2	EBI_BA+0x20	R/W	External Bus Interface Bank2 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reversed							WBUFEN
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reversed					TALE		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reversed					MCLKDIV		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reversed			CACCESS	ADSEPEN	CSPOLINV	DW16	EN

Bits	Description	
[31:25]	**Reserved**	Reserved.
[24]	**WBUFEN**	**EBI Write Buffer Enable Bit** 0 = EBI write buffer Disabled. 1 = EBI write buffer Enabled. **Note:** This bit only available in EBI_CTL0 register.
[23:19]	**Reserved**	Reserved.
[18:16]	**TALE**	**Extend Time of ALE** The EBI_ALE high pulse period (tALE) to latch the address can be controlled by TALE. tALE = (TALE+1)*EBI_MCLK. **Note:** This field only available in EBI_CTL0 register.
[15:11]	**Reserved**	Reserved.
[10:8]	**MCLKDIV**	**External Output Clock Divider** The frequency of EBI output clock (MCLK) is controlled by MCLKDIV as follow: 000 = HCLK/1. 001 = HCLK/2. 010 = HCLK/4. 011 = HCLK/8. 100 = HCLK/16. 101 = HCLK/32. 110 = HCLK/64. 111 = HCLK/128.
[7:5]	**Reserved**	Reserved.

[4]	**CACCESS**	**Continuous Data Access Mode** When continuous access mode enabled, the tASU, tALE and tLHD cycles are bypass for continuous data transfer request. 0 = Continuous data access mode Disabled. 1 = Continuous data access mode Enabled.
[3]	**ADSEPEN**	**EBI Address/Data Bus Separate Mode Enable Bit** 0 = Address/Data Bus Separate Mode Disabled. 1 = Address/Data Bus Separate Mode Enabled.
[2]	**CSPOLINV**	**Chip Select Pin Polar Inverse** This bit defines the active level of EBI chip select pin (EBI_nCS). 0 = Chip select pin (EBI_nCS) is active low. 1 = Chip select pin (EBI_nCS) is active high.
[1]	**DW16**	**EBI Data Width 16-bit Select** This bit defines if the EBI data width is 8-bit or 16-bit. 0 = EBI data width is 8-bit. 1 = EBI data width is 16-bit.
[0]	**EN**	**EBI Enable Bit** This bit is the functional enable bit for EBI. 0 = EBI function Disabled. 1 = EBI function Enabled.

External Bus Interface Timing Control Register (EBI_TCTLx)

Register	Offset	R/W	Description	Reset Value
EBI_TCTL0	EBI_BA+0x04	R/W	External Bus Interface Bank0 Timing Control Register	0x0000_0000
EBI_TCTL1	EBI_BA+0x14	R/W	External Bus Interface Bank1 Timing Control Register	0x0000_0000
EBI_TCTL2	EBI_BA+0x24	R/W	External Bus Interface Bank2 Timing Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				R2R			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
WAHDOFF	RAHDOFF	Reserved					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
W2X				Reversed	TAHD		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TACC					Reserved		

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:24]	**R2R**	**Idle Cycle Between Read-to-read** This field defines the number of R2R idle cycle. R2R idle cycle = (R2R * EBI_MCLK). When read action is finished and the next action is going to read, R2R idle cycle is inserted and EBI_nCS return to idle state.
[23]	**WAHDOFF**	**Access Hold Time Disable Control When Write** 0 = Data Access Hold Time (tAHD) during EBI writing Enabled. 1 = Data Access Hold Time (tAHD) during EBI writing Disabled.
[22]	**RAHDOFF**	**Access Hold Time Disable Control When Read** 0 = Data Access Hold Time (tAHD) during EBI reading Enabled. 1 = Data Access Hold Time (tAHD) during EBI reading Disabled.
[21:16]	**Reserved**	Reserved.
[15:12]	**W2X**	**Idle Cycle After Write** This field defines the number of W2X idle cycle. W2X idle cycle = (W2X * EBI_MCLK). When write action is finished, W2X idle cycle is inserted and EBI_nCS return to idle state.
[11]	**Reserved**	Reserved.
[10:8]	**TAHD**	**EBI Data Access Hold Time** TAHD defines data access hold time (tAHD). tAHD = (TAHD +1) * EBI_MCLK.
[7:3]	**TACC**	**EBI Data Access Time** TACC defines data access time (tACC).

		tACC = (TACC +1) * EBI_MCLK.
[2:0]	**Reserved**	Reserved.

6.15 General Purpose I/O (GPIO)

6.15.1 Overview

This chip has up to 208 General-Purpose I/O (GPIO) pins and can be shared with other function pins depending on the chip configuration. These 208 pins are arranged in 16 ports named as PA, PB, PC, PD, PE, PF, PG, PH, PI, PJ, PK, PL, PM, PN. Each port has 16 pins on port, except PB, PC, PH and PN. PB has 8 pins on port. PC has 12 pins on port. PH and PN has 14 pins on port. All of the 208 I/O pins is independent and can be easily configured by user to meet various system configurations and design requirements.

The I/O type of each of I/O pins can be configured by software individually as Input, Push-pull output or Open-drain output. After reset, all 208 I/O pins are configured in General-Purpose I/O Input mode. Each I/O pin has a very weakly individual pull-up resistor. Please refer to the MA35D1 Series Datasheet for detailed pin operation voltage information about V_{DD} electrical characteristics.

6.15.2 Features

- Three I/O modes:
  - Push-Pull Output mode
  - Open-Drain Output mode
  - Input only with high impendence mode
- TTL/Schmitt trigger input selectable
- I/O pin can be configured as interrupt source with edge/level setting
- Supports High Drive and High Slew Rate I/O mode
- Support independent pull-up and pull-down control
- Enabling the pin interrupt function will also enable the wake-up function

6.15.3 Block Diagram

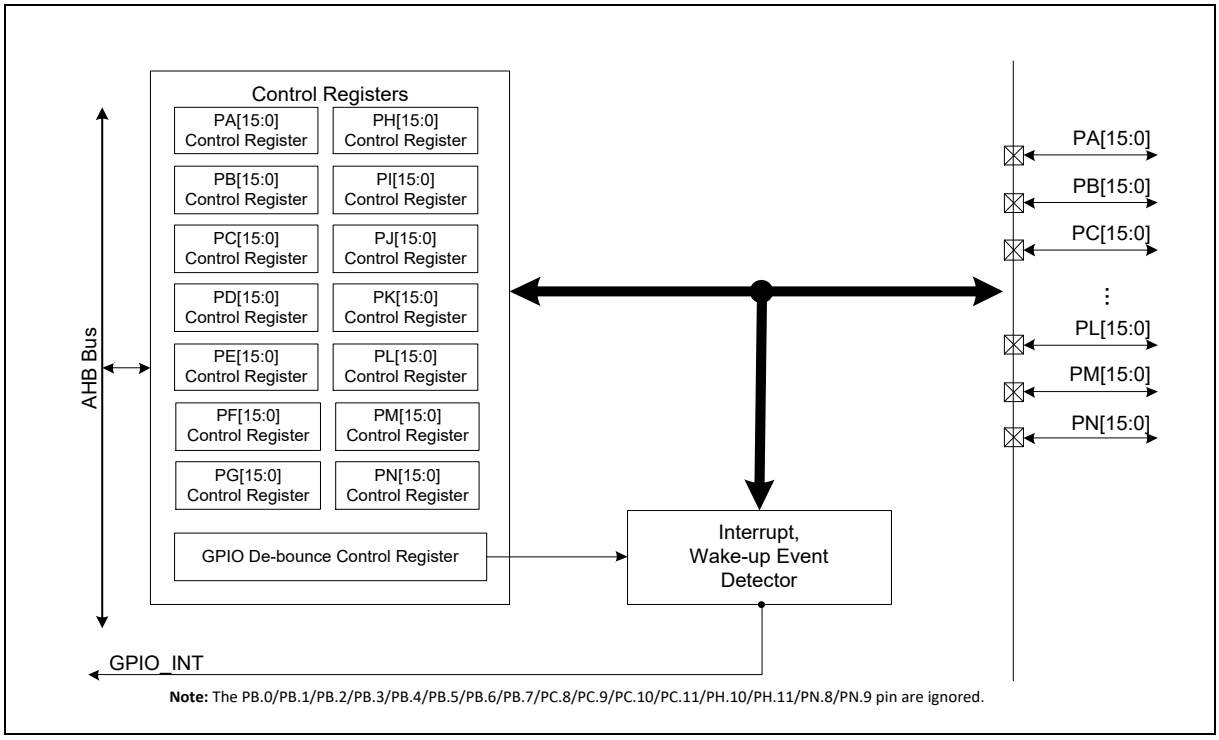


Figure 6.15-1 GPIO Block Diagram

6.15.4 Basic Configuration

- Clock Source Configuration
  - Enable GPIO clock in CLK_SYSCLK1[29:16]
- Reset Configuration
  - Reset GPIO in GPIORST (SYS_IPRST0 [27])
- Pin Configuration

Group	GPIO	MFP
EINT0	PF.14 , PL.14	MFP12
	PG.0, PI.4, PL.6	MFP13
EINT1	PB.10, PD.12, PG.2, PI.5, PJ.15, PK.5, PL.7, PM.14	MFP13
EINT2	PL.15	MFP12
	PB.11, PG.4, PI.6, PK.6, PK.12, PL.8, PM.15	MFP13
EINT3	PD.14, PG.6, PI.7, PK.7, PK.14, PL.9	MFP13

To configure pin Px.n as a General-Purpose I/O, set the corresponding field of register SYS_GPx_MFPL, SYS_GPx_MFPH to 0, x is A~N.

For example, if user wants to configure pin PA.0 as a General-Purpose I/O, it's necessary to set MFP_GPA0 (SYS_GPA_MFPL[3:0]) to 0.

6.15.5 Functional Description

6.15.5.1 Input mode

Set MODEn (Px_MODE[2n+1:2n]) to 00 as the Px.n pin is in Input mode and the I/O pin is in tri-state (high impedance) without output drive capability. The PIN (Px_PIN[n]) value reflects the status of the corresponding port pins.

6.15.5.2 Push-pull Output Mode

Figure 6.15-2 shows the diagram of Push-pull Output Mode. Set MODEn (Px_MODE[2n+1:2n]) to 01 as Px.n pin is in Push-pull Output mode and the I/O pin supports digital output function with source/sink current capability. The bit value in the corresponding DOUT (Px_DOUT[n]) is driven on the pin.

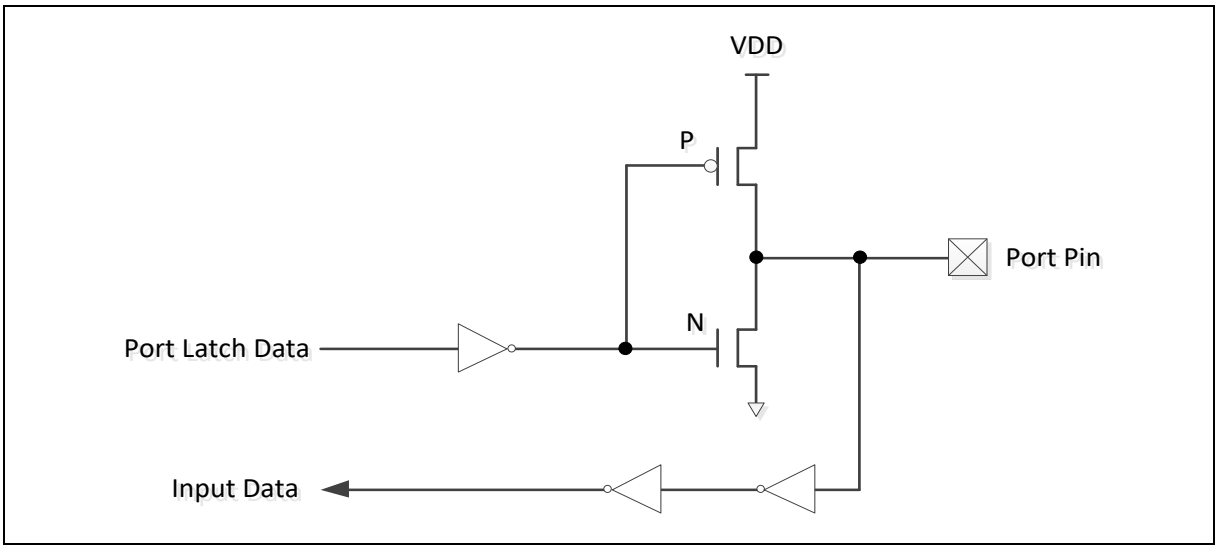


Figure 6.15-2 Push-Pull Output

6.15.5.3 Open-drain Mode

Figure 6.15-3 shows the diagram of Open-drain Mode. Set MODEn (Px_MODE[2n+1:2n]) to 10 the Px.n pin is in Open-drain mode and the digital output function of I/O pin supports only sink current capability, an external pull-up register is needed for driving high state. If the bit value in the corresponding DOUT (Px_DOUT[n]) bit is 0, the pin drive a low output on the pin. If the bit value in the corresponding DOUT (Px_DOUT[n]) bit is 1, the pin output drives high that is controlled by external pull high resistor.

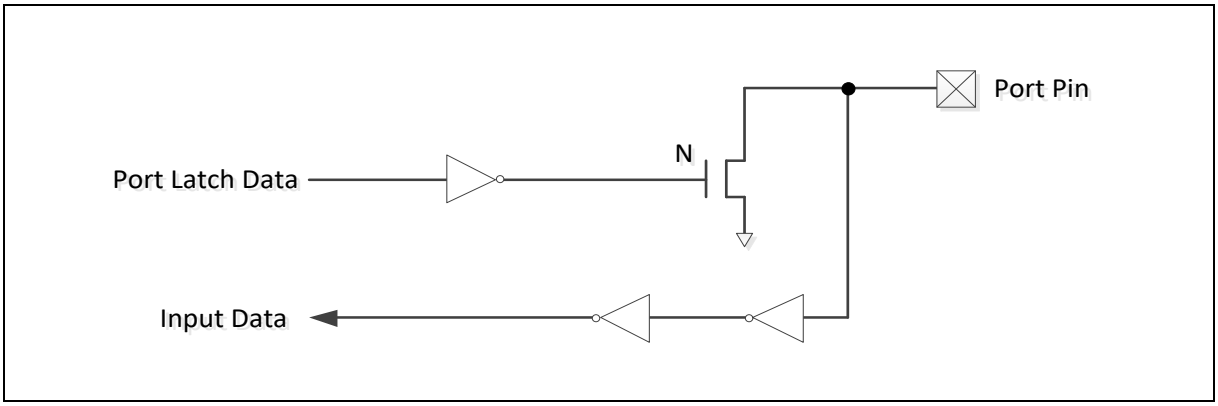


Figure 6.15-3 Open-Drain Output

6.15.5.4 GPIO Interrupt and Wake-up Function

Each GPIO pin can be set as chip interrupt source by setting correlative RHIEN (Px_INTEN[n+16])/ FLIEN (Px_INTEN[n]) bit and TYPE (Px_INTTYPE[n]). There are five types of interrupt condition can be selected: low level trigger, high level trigger, falling edge trigger, rising edge trigger and both rising and falling edge trigger. The GPIO can also be the chip wake-up source when chip enters Idle/Power-down mode. The setting of wake-up trigger condition is the same as GPIO interrupt trigger.

6.15.5.5 GPIO De-bounce Function

GPIO de-bounce function can be used to sample interrupt input for each GPIO pin and prevent unexpected interrupt happened which caused by noise. GPIO de-bounce function only support edge detection trigger type. For edge trigger condition, there are three types of interrupt condition can be selected for de-bounce function: falling edge trigger, rising edge trigger and both rising and falling edge trigger. If user wants to use de-bounce function, de-bounce enable control register Px_DBEN must be set for corresponding GPIO pin. The de-bounce clock source can be HXT or LIRC by setting DBCLKSRC (Px_DBCTL[4]) register. The DBCLKSEL (Px_DBCTL[3:0]) register can control sampling cycle period.

Figure 6.15-4 shows GPIO rising edge trigger interrupt. The interval of time between the two valid sample signal is determined by DBCLKSRC (Px_DBCTL[4]) and DBCLKSEL (Px_DBCTL[3:0]). Each valid data from GPIO pin need to be sample twice. For rising edge setting, if pin status is low before setting DBEN (Px_DBEN), interrupt will happen when generating a pin high valid data. But if the pin status is high before setting DBEN (Px_DBEN), interrupt will happen when generating a pin low valid data first, and then generating a pin high valid data. For falling edge trigger, Figure 6.15-5 shows the situation is opposite to rising edge trigger.

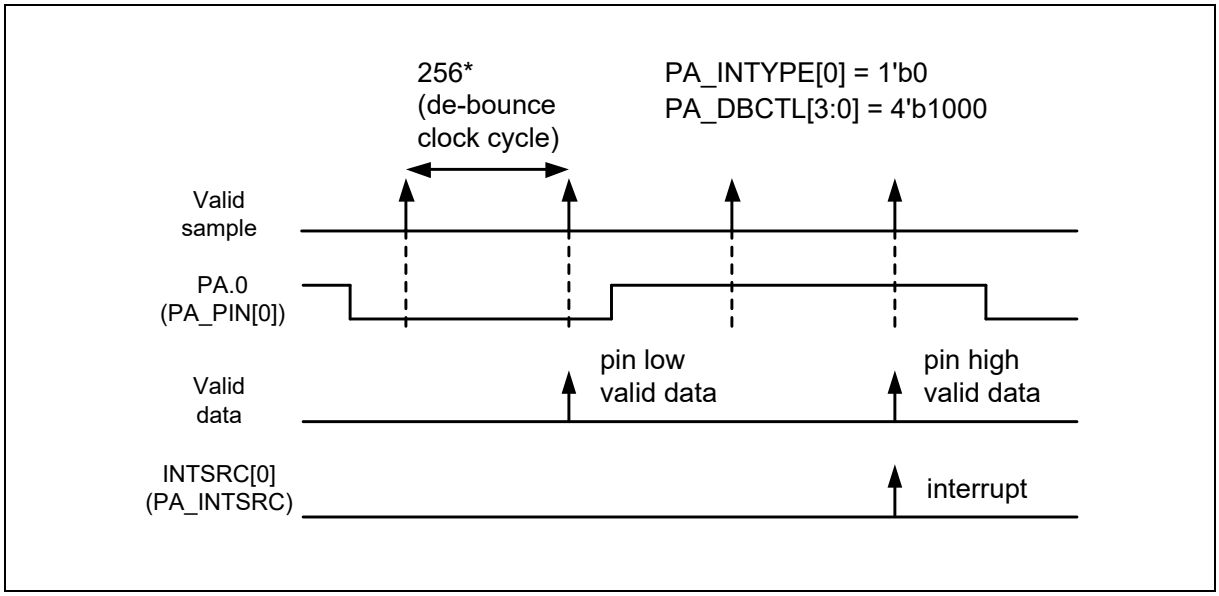


Figure 6.15-4 GPIO Rising Edge Trigger Interrupt

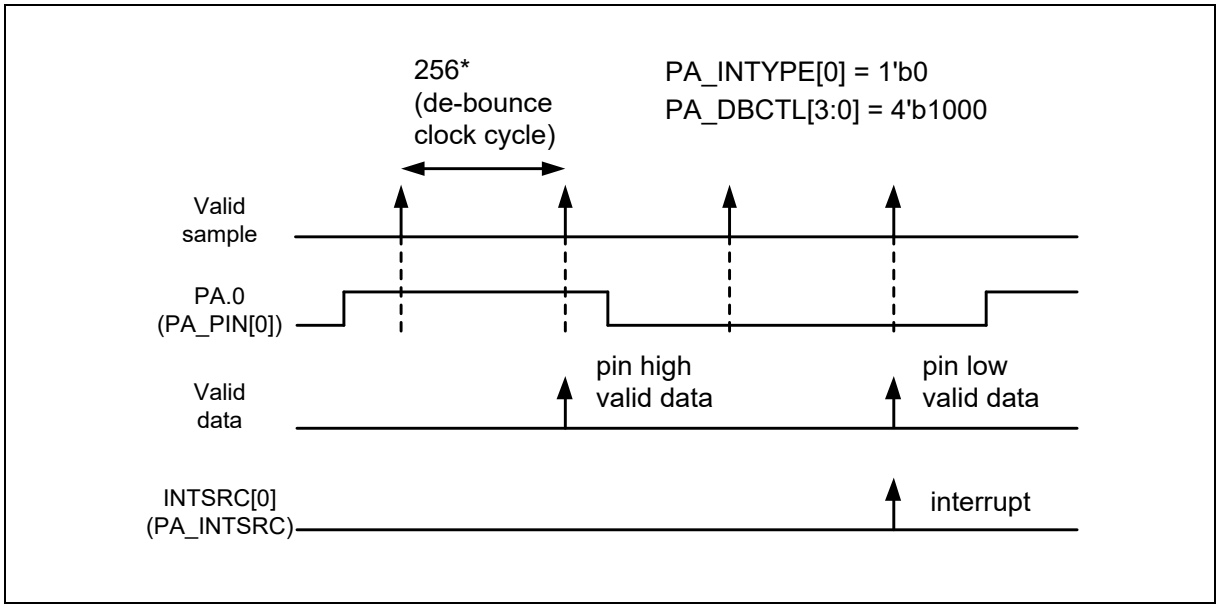


Figure 6.15-5 GPIO Falling Edge Trigger Interrupt

6.15.5.6 GPIO Digital Input Path Disable Control

User can disable GPIO digital input path by setting DINOFF (Px_DINOFF[n]). When GPIO digital input path is disabled, the digital input pin value PIN (Px_PIN[n]) is tied to low. By the way, the GPIO digital input path is force disabled by hardware and DINOFF control is useless when I/O function configure as ADC.

6.15.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
GPIO Base Address: **GPIO_BA = 0x4004_0000**				
PA_MODE	GPIO_BA+0x000	R/W	PA I/O Mode Control	0x0000_0000
PA_DINOFF	GPIO_BA+0x004	R/W	PA Digital Input Path Disable Control	0x0000_0000
PA_DOUT	GPIO_BA+0x008	R/W	PA Data Output Value	0x0000_FFFF
PA_DATMSK	GPIO_BA+0x00C	R/W	PA Data Output Write Mask	0x0000_0000
PA_PIN	GPIO_BA+0x010	R	PA Pin Value	0x0000_XXXX
PA_DBEN	GPIO_BA+0x014	R/W	PA De-bounce Enable Control Register	0x0000_0000
PA_INTTYPE	GPIO_BA+0x018	R/W	PA Interrupt Trigger Type Control	0x0000_0000
PA_INTEN	GPIO_BA+0x01C	R/W	PA Interrupt Enable Control Register	0x0000_0000
PA_INTSRC	GPIO_BA+0x020	R/W	PA Interrupt Source Flag	0x0000_XXXX
PA_SMTEN	GPIO_BA+0x024	R/W	PA Input Schmitt Trigger Enable Register	0x0000_0000
PA_SLEWCTL	GPIO_BA+0x028	R/W	PA High Slew Rate Control Register	0x0000_0000
PA_PUSEL	GPIO_BA+0x030	R/W	PA Pull-up and Pull-down Selection Register	0x0000_0000
PA_DBCTL	GPIO_BA+0x034	R/W	PA Interrupt De-bounce Control Register	0x0000_0020
PA_DSL	GPIO_BA+0x038	R/W	PA Low Byte Driver Strength Control Register	0x0000_0000
PA_DSH	GPIO_BA+0x03C	R/W	PA High Byte Driver Strength Control Register	0x0000_0000
PB_MODE	GPIO_BA+0x040	R/W	PB I/O Mode Control	0x0000_0000
PB_DINOFF	GPIO_BA+0x044	R/W	PB Digital Input Path Disable Control	0x0000_0000
PB_DOUT	GPIO_BA+0x048	R/W	PB Data Output Value	0x0000_FFFF
PB_DATMSK	GPIO_BA+0x04C	R/W	PB Data Output Write Mask	0x0000_0000
PB_PIN	GPIO_BA+0x050	R	PB Pin Value	0x0000_XXXX
PB_DBEN	GPIO_BA+0x054	R/W	PB De-bounce Enable Control Register	0x0000_0000
PB_INTTYPE	GPIO_BA+0x058	R/W	PB Interrupt Trigger Type Control	0x0000_0000
PB_INTEN	GPIO_BA+0x05C	R/W	PB Interrupt Enable Control Register	0x0000_0000
PB_INTSRC	GPIO_BA+0x060	R/W	PB Interrupt Source Flag	0x0000_XXXX
PB_SMTEN	GPIO_BA+0x064	R/W	PB Input Schmitt Trigger Enable Register	0x0000_0000
PB_SLEWCTL	GPIO_BA+0x068	R/W	PB High Slew Rate Control Register	0x0000_0000
PB_SPW	GPIO_BA+0x06C	R/W	PB Power Mode Selection Register	0x0000_FFFF

PB_PUSEL	GPIO_BA+0x070	R/W	PB Pull-up and Pull-down Selection Register	0x0000_0000
PB_DBCTL	GPIO_BA+0x074	R/W	PB Interrupt De-bounce Control Register	0x0000_0020
PB_DSL	GPIO_BA+0x078	R/W	PB Low Byte Driver Strength Control Register	0x0000_0000
PB_DSH	GPIO_BA+0x07C	R/W	PB High Byte Driver Strength Control Register	0x0000_0000
PC_MODE	GPIO_BA+0x080	R/W	PC I/O Mode Control	0x0000_0000
PC_DINOFF	GPIO_BA+0x084	R/W	PC Digital Input Path Disable Control	0x0000_0000
PC_DOUT	GPIO_BA+0x088	R/W	PC Data Output Value	0x0000_FFFF
PC_DATMSK	GPIO_BA+0x08C	R/W	PC Data Output Write Mask	0x0000_0000
PC_PIN	GPIO_BA+0x090	R	PC Pin Value	0x0000_XXXX
PC_DBEN	GPIO_BA+0x094	R/W	PC De-bounce Enable Control Register	0x0000_0000
PC_INTTYPE	GPIO_BA+0x098	R/W	PC Interrupt Trigger Type Control	0x0000_0000
PC_INTEN	GPIO_BA+0x09C	R/W	PC Interrupt Enable Control Register	0x0000_0000
PC_INTSRC	GPIO_BA+0x0A0	R/W	PC Interrupt Source Flag	0x0000_XXXX
PC_SMTEN	GPIO_BA+0x0A4	R/W	PC Input Schmitt Trigger Enable Register	0x0000_0000
PC_SLEWCTL	GPIO_BA+0x0A8	R/W	PC High Slew Rate Control Register	0x0000_0000
PC_SPW	GPIO_BA+0x0AC	R/W	PC Power Mode Selection Register	0x0000_FFFF
PC_PUSEL	GPIO_BA+0x0B0	R/W	PC Pull-up and Pull-down Selection Register	0x0000_0000
PC_DBCTL	GPIO_BA+0x0B4	R/W	PC Interrupt De-bounce Control Register	0x0000_0020
PC_DSL	GPIO_BA+0x0B8	R/W	PC Low Byte Driver Strength Control Register	0x0000_0000
PC_DSH	GPIO_BA+0x0BC	R/W	PC High Byte Driver Strength Control Register	0x0000_0000
PD_MODE	GPIO_BA+0x0C0	R/W	PD I/O Mode Control	0x0000_0000
PD_DINOFF	GPIO_BA+0x0C4	R/W	PD Digital Input Path Disable Control	0x0000_0000
PD_DOUT	GPIO_BA+0x0C8	R/W	PD Data Output Value	0x0000_FFFF
PD_DATMSK	GPIO_BA+0x0CC	R/W	PD Data Output Write Mask	0x0000_0000
PD_PIN	GPIO_BA+0x0D0	R	PD Pin Value	0x0000_XXXX
PD_DBEN	GPIO_BA+0x0D4	R/W	PD De-bounce Enable Control Register	0x0000_0000
PD_INTTYPE	GPIO_BA+0x0D8	R/W	PD Interrupt Trigger Type Control	0x0000_0000
PD_INTEN	GPIO_BA+0x0DC	R/W	PD Interrupt Enable Control Register	0x0000_0000
PD_INTSRC	GPIO_BA+0x0E0	R/W	PD Interrupt Source Flag	0x0000_XXXX
PD_SMTEN	GPIO_BA+0x0E4	R/W	PD Input Schmitt Trigger Enable Register	0x0000_0000
PD_SLEWCTL	GPIO_BA+0x0E8	R/W	PD High Slew Rate Control Register	0x0000_0000

PD_SPW	GPIO_BA+0x0EC	R/W	PD Power Mode Selection Register	0x0000_FFFF
PD_PUSEL	GPIO_BA+0x0F0	R/W	PD Pull-up and Pull-down Selection Register	0x0000_0000
PD_DBCTL	GPIO_BA+0x0F4	R/W	PD Interrupt De-bounce Control Register	0x0000_0020
PD_DSL	GPIO_BA+0x0F8	R/W	PD Low Byte Driver Strength Control Register	0x0000_0000
PD_DSH	GPIO_BA+0x0FC	R/W	PD High Byte Driver Strength Control Register	0x0000_0000
PE_MODE	GPIO_BA+0x100	R/W	PE I/O Mode Control	0x0000_0000
PE_DINOFF	GPIO_BA+0x104	R/W	PE Digital Input Path Disable Control	0x0000_0000
PE_DOUT	GPIO_BA+0x108	R/W	PE Data Output Value	0x0000_FFFF
PE_DATMSK	GPIO_BA+0x10C	R/W	PE Data Output Write Mask	0x0000_0000
PE_PIN	GPIO_BA+0x110	R	PE Pin Value	0x0000_XXXX
PE_DBEN	GPIO_BA+0x114	R/W	PE De-bounce Enable Control Register	0x0000_0000
PE_INTTYPE	GPIO_BA+0x118	R/W	PE Interrupt Trigger Type Control	0x0000_0000
PE_INTEN	GPIO_BA+0x11C	R/W	PE Interrupt Enable Control Register	0x0000_0000
PE_INTSRC	GPIO_BA+0x120	R/W	PE Interrupt Source Flag	0x0000_XXXX
PE_SMTEN	GPIO_BA+0x124	R/W	PE Input Schmitt Trigger Enable Register	0x0000_0000
PE_SLEWCTL	GPIO_BA+0x128	R/W	PE High Slew Rate Control Register	0x0000_0000
PE_SPW	GPIO_BA+0x12C	R/W	PE Power Mode Selection Register	0x0000_FFFF
PE_PUSEL	GPIO_BA+0x130	R/W	PE Pull-up and Pull-down Selection Register	0x0000_0000
PE_DBCTL	GPIO_BA+0x134	R/W	PE Interrupt De-bounce Control Register	0x0000_0020
PE_DSL	GPIO_BA+0x138	R/W	PE Low Byte Driver Strength Control Register	0x0000_0000
PE_DSH	GPIO_BA+0x13C	R/W	PE High Byte Driver Strength Control Register	0x0000_0000
PF_MODE	GPIO_BA+0x140	R/W	PF I/O Mode Control	0x0000_0000
PF_DINOFF	GPIO_BA+0x144	R/W	PF Digital Input Path Disable Control	0x0000_0000
PF_DOUT	GPIO_BA+0x148	R/W	PF Data Output Value	0x0000_FFFF
PF_DATMSK	GPIO_BA+0x14C	R/W	PF Data Output Write Mask	0x0000_0000
PF_PIN	GPIO_BA+0x150	R	PF Pin Value	0x0000_XXXX
PF_DBEN	GPIO_BA+0x154	R/W	PF De-bounce Enable Control Register	0x0000_0000
PF_INTTYPE	GPIO_BA+0x158	R/W	PF Interrupt Trigger Type Control	0x0000_0000
PF_INTEN	GPIO_BA+0x15C	R/W	PF Interrupt Enable Control Register	0x0000_0000
PF_INTSRC	GPIO_BA+0x160	R/W	PF Interrupt Source Flag	0x0000_XXXX
PF_SMTEN	GPIO_BA+0x164	R/W	PF Input Schmitt Trigger Enable Register	0x0000_0000

PF_SLEWCTL	GPIO_BA+0x168	R/W	PF High Slew Rate Control Register	0x0000_0000
PF_SPW	GPIO_BA+0x16C	R/W	PF Power Mode Selection Register	0x0000_FFFF
PF_PUSEL	GPIO_BA+0x170	R/W	PF Pull-up and Pull-down Selection Register	0x0000_0000
PF_DBCTL	GPIO_BA+0x174	R/W	PF Interrupt De-bounce Control Register	0x0000_0020
PF_DSL	GPIO_BA+0x178	R/W	PF Low Byte Driver Strength Control Register	0x0000_0000
PF_DSH	GPIO_BA+0x17C	R/W	PF High Byte Driver Strength Control Register	0x0000_0000
PG_MODE	GPIO_BA+0x180	R/W	PG I/O Mode Control	0x0000_0000
PG_DINOFF	GPIO_BA+0x184	R/W	PG Digital Input Path Disable Control	0x0000_0000
PG_DOUT	GPIO_BA+0x188	R/W	PG Data Output Value	0x0000_FFFF
PG_DATMSK	GPIO_BA+0x18C	R/W	PG Data Output Write Mask	0x0000_0000
PG_PIN	GPIO_BA+0x190	R	PG Pin Value	0x0000_XXXX
PG_DBEN	GPIO_BA+0x194	R/W	PG De-bounce Enable Control Register	0x0000_0000
PG_INTTYPE	GPIO_BA+0x198	R/W	PG Interrupt Trigger Type Control	0x0000_0000
PG_INTEN	GPIO_BA+0x19C	R/W	PG Interrupt Enable Control Register	0x0000_0000
PG_INTSRC	GPIO_BA+0x1A0	R/W	PG Interrupt Source Flag	0x0000_XXXX
PG_SMTEN	GPIO_BA+0x1A4	R/W	PG Input Schmitt Trigger Enable Register	0x0000_0000
PG_SLEWCTL	GPIO_BA+0x1A8	R/W	PG High Slew Rate Control Register	0x0000_0000
PG_SPW	GPIO_BA+0x1AC	R/W	PG Power Mode Selection Register	0x0000_FFFF
PG_PUSEL	GPIO_BA+0x1B0	R/W	PG Pull-up and Pull-down Selection Register	0x0000_0000
PG_DBCTL	GPIO_BA+0x1B4	R/W	PG Interrupt De-bounce Control Register	0x0000_0020
PG_DSL	GPIO_BA+0x1B8	R/W	PG Low Byte Driver Strength Control Register	0x0000_0000
PG_DSH	GPIO_BA+0x1BC	R/W	PG High Byte Driver Strength Control Register	0x0000_0000
PH_MODE	GPIO_BA+0x1C0	R/W	PH I/O Mode Control	0x0000_0000
PH_DINOFF	GPIO_BA+0x1C4	R/W	PH Digital Input Path Disable Control	0x0000_0000
PH_DOUT	GPIO_BA+0x1C8	R/W	PH Data Output Value	0x0000_FFFF
PH_DATMSK	GPIO_BA+0x1CC	R/W	PH Data Output Write Mask	0x0000_0000
PH_PIN	GPIO_BA+0x1D0	R	PH Pin Value	0x0000_XXXX
PH_DBEN	GPIO_BA+0x1D4	R/W	PH De-bounce Enable Control Register	0x0000_0000
PH_INTTYPE	GPIO_BA+0x1D8	R/W	PH Interrupt Trigger Type Control	0x0000_0000
PH_INTEN	GPIO_BA+0x1DC	R/W	PH Interrupt Enable Control Register	0x0000_0000
PH_INTSRC	GPIO_BA+0x1E0	R/W	PH Interrupt Source Flag	0x0000_XXXX

PH_SMTEN	GPIO_BA+0x1E4	R/W	PH Input Schmitt Trigger Enable Register	0x0000_0000
PH_SLEWCTL	GPIO_BA+0x1E8	R/W	PH High Slew Rate Control Register	0x0000_0000
PH_PUSEL	GPIO_BA+0x1F0	R/W	PH Pull-up and Pull-down Selection Register	0x0000_0000
PH_DBCTL	GPIO_BA+0x1F4	R/W	PH Interrupt De-bounce Control Register	0x0000_0020
PH_DSL	GPIO_BA+0x1F8	R/W	PH Low Byte Driver Strength Control Register	0x0000_0000
PH_DSH	GPIO_BA+0x1FC	R/W	PH High Byte Driver Strength Control Register	0x0000_0000
PI_MODE	GPIO_BA+0x200	R/W	PI I/O Mode Control	0x0000_0000
PI_DINOFF	GPIO_BA+0x204	R/W	PI Digital Input Path Disable Control	0x0000_0000
PI_DOUT	GPIO_BA+0x208	R/W	PI Data Output Value	0x0000_FFFF
PI_DATMSK	GPIO_BA+0x20C	R/W	PI Data Output Write Mask	0x0000_0000
PI_PIN	GPIO_BA+0x210	R	PI Pin Value	0x0000_XXXX
PI_DBEN	GPIO_BA+0x214	R/W	PI De-bounce Enable Control Register	0x0000_0000
PI_INTTYPE	GPIO_BA+0x218	R/W	PI Interrupt Trigger Type Control	0x0000_0000
PI_INTEN	GPIO_BA+0x21C	R/W	PI Interrupt Enable Control Register	0x0000_0000
PI_INTSRC	GPIO_BA+0x220	R/W	PI Interrupt Source Flag	0x0000_XXXX
PI_SMTEN	GPIO_BA+0x224	R/W	PI Input Schmitt Trigger Enable Register	0x0000_0000
PI_SLEWCTL	GPIO_BA+0x228	R/W	PI High Slew Rate Control Register	0x0000_0000
PI_PUSEL	GPIO_BA+0x230	R/W	PI Pull-up and Pull-down Selection Register	0x0000_0000
PI_DBCTL	GPIO_BA+0x234	R/W	PI Interrupt De-bounce Control Register	0x0000_0020
PI_DSL	GPIO_BA+0x238	R/W	PI Low Byte Driver Strength Control Register	0x0000_0000
PI_DSH	GPIO_BA+0x23C	R/W	PI High Byte Driver Strength Control Register	0x0000_0000
PJ_MODE	GPIO_BA+0x240	R/W	PJ I/O Mode Control	0x0000_0000
PJ_DINOFF	GPIO_BA+0x244	R/W	PJ Digital Input Path Disable Control	0x0000_0000
PJ_DOUT	GPIO_BA+0x248	R/W	PJ Data Output Value	0x0000_FFFF
PJ_DATMSK	GPIO_BA+0x24C	R/W	PJ Data Output Write Mask	0x0000_0000
PJ_PIN	GPIO_BA+0x250	R	PJ Pin Value	0x0000_XXXX
PJ_DBEN	GPIO_BA+0x254	R/W	PJ De-bounce Enable Control Register	0x0000_0000
PJ_INTTYPE	GPIO_BA+0x258	R/W	PJ Interrupt Trigger Type Control	0x0000_0000
PJ_INTEN	GPIO_BA+0x25C	R/W	PJ Interrupt Enable Control Register	0x0000_0000
PJ_INTSRC	GPIO_BA+0x260	R/W	PJ Interrupt Source Flag	0x0000_XXXX
PJ_SMTEN	GPIO_BA+0x264	R/W	PJ Input Schmitt Trigger Enable Register	0x0000_0000

PJ_SLEWCTL	GPIO_BA+0x268	R/W	PJ High Slew Rate Control Register	0x0000_0000
PJ_SPW	GPIO_BA+0x26C	R/W	PJ Power Mode Selection Register	0x0000_FFFF
PJ_PUSEL	GPIO_BA+0x270	R/W	PJ Pull-up and Pull-down Selection Register	0x0000_0000
PJ_DBCTL	GPIO_BA+0x274	R/W	PJ Interrupt De-bounce Control Register	0x0000_0020
PJ_DSL	GPIO_BA+0x278	R/W	PJ Low Byte Driver Strength Control Register	0x0000_0000
PJ_DSH	GPIO_BA+0x27C	R/W	PJ High Byte Driver Strength Control Register	0x0000_0000
PK_MODE	GPIO_BA+0x280	R/W	PK I/O Mode Control	0x0000_0000
PK_DINOFF	GPIO_BA+0x284	R/W	PK Digital Input Path Disable Control	0x0000_0000
PK_DOUT	GPIO_BA+0x288	R/W	PK Data Output Value	0x0000_FFFF
PK_DATMSK	GPIO_BA+0x28C	R/W	PK Data Output Write Mask	0x0000_0000
PK_PIN	GPIO_BA+0x290	R	PK Pin Value	0x0000_XXXX
PK_DBEN	GPIO_BA+0x294	R/W	PK De-bounce Enable Control Register	0x0000_0000
PK_INTTYPE	GPIO_BA+0x298	R/W	PK Interrupt Trigger Type Control	0x0000_0000
PK_INTEN	GPIO_BA+0x29C	R/W	PK Interrupt Enable Control Register	0x0000_0000
PK_INTSRC	GPIO_BA+0x2A0	R/W	PK Interrupt Source Flag	0x0000_XXXX
PK_SMTEN	GPIO_BA+0x2A4	R/W	PK Input Schmitt Trigger Enable Register	0x0000_0000
PK_SLEWCTL	GPIO_BA+0x2A8	R/W	PK High Slew Rate Control Register	0x0000_0000
PK_SPW	GPIO_BA+0x2AC	R/W	PK Power Mode Selection Register	0x0000_FFFF
PK_PUSEL	GPIO_BA+0x2B0	R/W	PK Pull-up and Pull-down Selection Register	0x0000_0000
PK_DBCTL	GPIO_BA+0x2B4	R/W	PK Interrupt De-bounce Control Register	0x0000_0020
PK_DSL	GPIO_BA+0x2B8	R/W	PK Low Byte Driver Strength Control Register	0x0000_0000
PK_DSH	GPIO_BA+0x2BC	R/W	PK High Byte Driver Strength Control Register	0x0000_0000
PL_MODE	GPIO_BA+0x2C0	R/W	PL I/O Mode Control	0x0000_0000
PL_DINOFF	GPIO_BA+0x2C4	R/W	PL Digital Input Path Disable Control	0x0000_0000
PL_DOUT	GPIO_BA+0x2C8	R/W	PL Data Output Value	0x0000_FFFF
PL_DATMSK	GPIO_BA+0x2CC	R/W	PL Data Output Write Mask	0x0000_0000
PL_PIN	GPIO_BA+0x2D0	R	PL Pin Value	0x0000_XXXX
PL_DBEN	GPIO_BA+0x2D4	R/W	PL De-bounce Enable Control Register	0x0000_0000
PL_INTTYPE	GPIO_BA+0x2D8	R/W	PL Interrupt Trigger Type Control	0x0000_0000
PL_INTEN	GPIO_BA+0x2DC	R/W	PL Interrupt Enable Control Register	0x0000_0000
PL_INTSRC	GPIO_BA+0x2E0	R/W	PL Interrupt Source Flag	0x0000_XXXX

PL_SMTEN	GPIO_BA+0x2E4	R/W	PL Input Schmitt Trigger Enable Register	0x0000_0000
PL_SLEWCTL	GPIO_BA+0x2E8	R/W	PL High Slew Rate Control Register	0x0000_0000
PL_PUSEL	GPIO_BA+0x2F0	R/W	PL Pull-up and Pull-down Selection Register	0x0000_0000
PL_DBCTL	GPIO_BA+0x2F4	R/W	PL Interrupt De-bounce Control Register	0x0000_0020
PL_DSL	GPIO_BA+0x2F8	R/W	PL Low Byte Driver Strength Control Register	0x0000_0000
PL_DSH	GPIO_BA+0x2FC	R/W	PL High Byte Driver Strength Control Register	0x0000_0000
PM_MODE	GPIO_BA+0x300	R/W	PM I/O Mode Control	0x0000_0000
PM_DINOFF	GPIO_BA+0x304	R/W	PM Digital Input Path Disable Control	0x0000_0000
PM_DOUT	GPIO_BA+0x308	R/W	PM Data Output Value	0x0000_FFFF
PM_DATMSK	GPIO_BA+0x30C	R/W	PM Data Output Write Mask	0x0000_0000
PM_PIN	GPIO_BA+0x310	R	PM Pin Value	0x0000_XXXX
PM_DBEN	GPIO_BA+0x314	R/W	PM De-bounce Enable Control Register	0x0000_0000
PM_INTTYPE	GPIO_BA+0x318	R/W	PM Interrupt Trigger Type Control	0x0000_0000
PM_INTEN	GPIO_BA+0x31C	R/W	PM Interrupt Enable Control Register	0x0000_0000
PM_INTSRC	GPIO_BA+0x320	R/W	PM Interrupt Source Flag	0x0000_XXXX
PM_SMTEN	GPIO_BA+0x324	R/W	PM Input Schmitt Trigger Enable Register	0x0000_0000
PM_SLEWCTL	GPIO_BA+0x328	R/W	PM High Slew Rate Control Register	0x0000_0000
PM_PUSEL	GPIO_BA+0x330	R/W	PM Pull-up and Pull-down Selection Register	0x0000_0000
PM_DBCTL	GPIO_BA+0x334	R/W	PM Interrupt De-bounce Control Register	0x0000_0020
PM_DSL	GPIO_BA+0x338	R/W	PM Low Byte Driver Strength Control Register	0x0000_0000
PM_DSH	GPIO_BA+0x33C	R/W	PM High Byte Driver Strength Control Register	0x0000_0000
PN_MODE	GPIO_BA+0x340	R/W	PN I/O Mode Control	0x0000_0000
PN_DINOFF	GPIO_BA+0x344	R/W	PN Digital Input Path Disable Control	0x0000_0000
PN_DOUT	GPIO_BA+0x348	R/W	PN Data Output Value	0x0000_FFFF
PN_DATMSK	GPIO_BA+0x34C	R/W	PN Data Output Write Mask	0x0000_0000
PN_PIN	GPIO_BA+0x350	R	PN Pin Value	0x0000_XXXX
PN_DBEN	GPIO_BA+0x354	R/W	PN De-bounce Enable Control Register	0x0000_0000
PN_INTTYPE	GPIO_BA+0x358	R/W	PN Interrupt Trigger Type Control	0x0000_0000
PN_INTEN	GPIO_BA+0x35C	R/W	PN Interrupt Enable Control Register	0x0000_0000
PN_INTSRC	GPIO_BA+0x360	R/W	PN Interrupt Source Flag	0x0000_XXXX
PN_SMTEN	GPIO_BA+0x364	R/W	PN Input Schmitt Trigger Enable Register	0x0000_0000

PN_SLEWCTL	GPIO_BA+0x368	R/W	PN High Slew Rate Control Register	0x0000_0000
PN_SPW	GPIO_BA+0x36C	R/W	PN Power Mode Selection Register	0x0000_FFFF
PN_PUSEL	GPIO_BA+0x370	R/W	PN Pull-up and Pull-down Selection Register	0x0000_0000
PN_DBCTL	GPIO_BA+0x374	R/W	PN Interrupt De-bounce Control Register	0x0000_0020
PN_DSL	GPIO_BA+0x378	R/W	PN Low Byte Driver Strength Control Register	0x0000_0000
PN_DSH	GPIO_BA+0x37C	R/W	PN High Byte Driver Strength Control Register	0x0000_0000
PAn_PDIO n=0,1..15	GPIO_BA+0x800+(0x04 * n)	R/W	GPIO PA.n Pin Data Input/Output Register	0x0000_000X
PBn_PDIO n=0,1..15	GPIO_BA+0x840+(0x04 * n)	R/W	GPIO PB.n Pin Data Input/Output Register	0x0000_000X
PCn_PDIO n=0,1..15	GPIO_BA+0x880+(0x04 * n)	R/W	GPIO PC.n Pin Data Input/Output Register	0x0000_000X
PDn_PDIO n=0,1..15	GPIO_BA+0x8C0+(0x04 * n)	R/W	GPIO PD.n Pin Data Input/Output Register	0x0000_000X
PEn_PDIO n=0,1..15	GPIO_BA+0x900+(0x04 * n)	R/W	GPIO PE.n Pin Data Input/Output Register	0x0000_000X
PFn_PDIO n=0,1..15	GPIO_BA+0x940+(0x04 * n)	R/W	GPIO PF.n Pin Data Input/Output Register	0x0000_000X
PGn_PDIO n=0,1..15	GPIO_BA+0x980+(0x04 * n)	R/W	GPIO PG.n Pin Data Input/Output Register	0x0000_000X
PHn_PDIO n=0,1..15	GPIO_BA+0x9C0+(0x04 * n)	R/W	GPIO PH.n Pin Data Input/Output Register	0x0000_000X
PIn_PDIO n=0,1..15	GPIO_BA+0xA00+(0x04 * n)	R/W	GPIO PI.n Pin Data Input/Output Register	0x0000_000X
PJn_PDIO n=0,1..15	GPIO_BA+0xA40+(0x04 * n)	R/W	GPIO PJ.n Pin Data Input/Output Register	0x0000_000X
PKn_PDIO n=0,1..15	GPIO_BA+0xA80+(0x04 * n)	R/W	GPIO PK.n Pin Data Input/Output Register	0x0000_000X
PLn_PDIO n=0,1..15	GPIO_BA+0xAC0+(0x04 * n)	R/W	GPIO PL.n Pin Data Input/Output Register	0x0000_000X
PMn_PDIO n=0,1..15	GPIO_BA+0xB00+(0x04 * n)	R/W	GPIO PM.n Pin Data Input/Output Register	0x0000_000X
PNn_PDIO n=0,1..15	GPIO_BA+0xB40+(0x04 * n)	R/W	GPIO PN.n Pin Data Input/Output Register	0x0000_000X

6.15.7 Register Description

Port A-N I/O Mode Control (Px_MODE)

Register	Offset	R/W	Description	Reset Value
PA_MODE	GPIO_BA+0x000	R/W	PA I/O Mode Control	0x0000_0000
PB_MODE	GPIO_BA+0x040	R/W	PB I/O Mode Control	0x0000_0000
PC_MODE	GPIO_BA+0x080	R/W	PC I/O Mode Control	0x0000_0000
PD_MODE	GPIO_BA+0x0C0	R/W	PD I/O Mode Control	0x0000_0000
PE_MODE	GPIO_BA+0x100	R/W	PE I/O Mode Control	0x0000_0000
PF_MODE	GPIO_BA+0x140	R/W	PF I/O Mode Control	0x0000_0000
PG_MODE	GPIO_BA+0x180	R/W	PG I/O Mode Control	0x0000_0000
PH_MODE	GPIO_BA+0x1C0	R/W	PH I/O Mode Control	0x0000_0000
PI_MODE	GPIO_BA+0x200	R/W	PI I/O Mode Control	0x0000_0000
PJ_MODE	GPIO_BA+0x240	R/W	PJ I/O Mode Control	0x0000_0000
PK_MODE	GPIO_BA+0x280	R/W	PK I/O Mode Control	0x0000_0000
PL_MODE	GPIO_BA+0x2C0	R/W	PL I/O Mode Control	0x0000_0000
PM_MODE	GPIO_BA+0x300	R/W	PM I/O Mode Control	0x0000_0000
PN_MODE	GPIO_BA+0x340	R/W	PN I/O Mode Control	0x0000_0000

31	30	29	28	27	26	25	24
MODE15		MODE14		MODE13		MODE12	
23	22	21	20	19	18	17	16
MODE11		MODE10		MODE9		MODE8	
15	14	13	12	11	10	9	8
MODE7		MODE6		MODE5		MODE4	
7	6	5	4	3	2	1	0
MODE3		MODE2		MODE1		MODE0	

Bits	Description	
[2n+1:2n] n=0,1..15	**MODEn**	**Port A-N I/O Pin[n] Mode Control** Determine each I/O mode of Px.n pins. 00 = Px.n is in Input mode. 01 = Px.n is in Push-pull Output mode. 10 = Px.n is in Open-drain Output mode. 11 = Reserved. **Note 1:** The default value is 0x0000_0000 and all pins will be input mode after chip powered on.

		Note 2: If MFOS is enabled then GPIO mode setting is ignored.

Port A-N Digital Input Path Disable Control (Px_DINOFF)

Register	Offset	R/W	Description	Reset Value
PA_DINOFF	GPIO_BA+0x004	R/W	PA Digital Input Path Disable Control	0x0000_0000
PB_DINOFF	GPIO_BA+0x044	R/W	PB Digital Input Path Disable Control	0x0000_0000
PC_DINOFF	GPIO_BA+0x084	R/W	PC Digital Input Path Disable Control	0x0000_0000
PD_DINOFF	GPIO_BA+0x0C4	R/W	PD Digital Input Path Disable Control	0x0000_0000
PE_DINOFF	GPIO_BA+0x104	R/W	PE Digital Input Path Disable Control	0x0000_0000
PF_DINOFF	GPIO_BA+0x144	R/W	PF Digital Input Path Disable Control	0x0000_0000
PG_DINOFF	GPIO_BA+0x184	R/W	PG Digital Input Path Disable Control	0x0000_0000
PH_DINOFF	GPIO_BA+0x1C4	R/W	PH Digital Input Path Disable Control	0x0000_0000
PI_DINOFF	GPIO_BA+0x204	R/W	PI Digital Input Path Disable Control	0x0000_0000
PJ_DINOFF	GPIO_BA+0x244	R/W	PJ Digital Input Path Disable Control	0x0000_0000
PK_DINOFF	GPIO_BA+0x284	R/W	PK Digital Input Path Disable Control	0x0000_0000
PL_DINOFF	GPIO_BA+0x2C4	R/W	PL Digital Input Path Disable Control	0x0000_0000
PM_DINOFF	GPIO_BA+0x304	R/W	PM Digital Input Path Disable Control	0x0000_0000
PN_DINOFF	GPIO_BA+0x344	R/W	PN Digital Input Path Disable Control	0x0000_0000

31	30	29	28	27	26	25	24
DINOFF							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DINOFF							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[n+16] n=0,1..15	**DINOFFn**	**Port A-N Pin[n] Digital Input Path Disable Bit** Each of these bits is used to control if the digital input path of corresponding Px.n pin is disabled. If input is analog signal, users can disable Px.n digital input path to avoid input current leakage. 0 = Px.n digital input path Enabled. 1 = Px.n digital input path Disabled (digital input tied to low).
[15:0]	**Reserved**	Reserved.

Port A-N Data Output Value (Px_DOUT)

Register	Offset	R/W	Description	Reset Value
PA_DOUT	GPIO_BA+0x008	R/W	PA Data Output Value	0x0000_FFFF
PB_DOUT	GPIO_BA+0x048	R/W	PB Data Output Value	0x0000_FFFF
PC_DOUT	GPIO_BA+0x088	R/W	PC Data Output Value	0x0000_FFFF
PD_DOUT	GPIO_BA+0x0C8	R/W	PD Data Output Value	0x0000_FFFF
PE_DOUT	GPIO_BA+0x108	R/W	PE Data Output Value	0x0000_FFFF
PF_DOUT	GPIO_BA+0x148	R/W	PF Data Output Value	0x0000_FFFF
PG_DOUT	GPIO_BA+0x188	R/W	PG Data Output Value	0x0000_FFFF
PH_DOUT	GPIO_BA+0x1C8	R/W	PH Data Output Value	0x0000_FFFF
PI_DOUT	GPIO_BA+0x208	R/W	PI Data Output Value	0x0000_FFFF
PJ_DOUT	GPIO_BA+0x248	R/W	PJ Data Output Value	0x0000_FFFF
PK_DOUT	GPIO_BA+0x288	R/W	PK Data Output Value	0x0000_FFFF
PL_DOUT	GPIO_BA+0x2C8	R/W	PL Data Output Value	0x0000_FFFF
PM_DOUT	GPIO_BA+0x308	R/W	PM Data Output Value	0x0000_FFFF
PN_DOUT	GPIO_BA+0x348	R/W	PN Data Output Value	0x0000_FFFF

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DOUT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DOUT							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[n] n=0,1..15	**DOUTn**	**Port A-N Pin[n] Output Value** Each of these bits controls the status of a Px.n pin when the Px.n is configured as Push-pull output or Open-drain output mode. 0 = Px.n will drive Low if the Px.n pin is configured as Push-pull output or Open-drain output mode. 1 = Px.n will drive High if the Px.n pin is configured as Push-pull output mode.

Port A-N Data Output Write Mask (Px_DATMSK)

Register	Offset	R/W	Description	Reset Value
PA_DATMSK	GPIO_BA+0x00C	R/W	PA Data Output Write Mask	0x0000_0000
PB_DATMSK	GPIO_BA+0x04C	R/W	PB Data Output Write Mask	0x0000_0000
PC_DATMSK	GPIO_BA+0x08C	R/W	PC Data Output Write Mask	0x0000_0000
PD_DATMSK	GPIO_BA+0x0CC	R/W	PD Data Output Write Mask	0x0000_0000
PE_DATMSK	GPIO_BA+0x10C	R/W	PE Data Output Write Mask	0x0000_0000
PF_DATMSK	GPIO_BA+0x14C	R/W	PF Data Output Write Mask	0x0000_0000
PG_DATMSK	GPIO_BA+0x18C	R/W	PG Data Output Write Mask	0x0000_0000
PH_DATMSK	GPIO_BA+0x1CC	R/W	PH Data Output Write Mask	0x0000_0000
PI_DATMSK	GPIO_BA+0x20C	R/W	PI Data Output Write Mask	0x0000_0000
PJ_DATMSK	GPIO_BA+0x24C	R/W	PJ Data Output Write Mask	0x0000_0000
PK_DATMSK	GPIO_BA+0x28C	R/W	PK Data Output Write Mask	0x0000_0000
PL_DATMSK	GPIO_BA+0x2CC	R/W	PL Data Output Write Mask	0x0000_0000
PM_DATMSK	GPIO_BA+0x30C	R/W	PM Data Output Write Mask	0x0000_0000
PN_DATMSK	GPIO_BA+0x34C	R/W	PN Data Output Write Mask	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DATMSK							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATMSK							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[n] n=0,1..15	**DATMSKn**	**Port A-N Pin[n] Data Output Write Mask** These bits are used to protect the corresponding DOUT (Px_DOUT[n]) bit. When the DATMSK (Px_DATMSK[n]) bit is set to 1, the corresponding DOUT (Px_DOUT[n]) bit is protected. If the write signal is masked, writing data to the protect bit is ignored. 0 = Corresponding DOUT (Px_DOUT[n]) bit can be updated. 1 = Corresponding DOUT (Px_DOUT[n]) bit protected.

		Note: This function only protects the corresponding DOUT (Px_DOUT[n]) bit, and will not protect the corresponding PDIO (Pxn_PDIO[0]) bit.

Port A-N Pin Value (Px_PIN)

Register	Offset	R/W	Description	Reset Value
PA_PIN	GPIO_BA+0x010	R	PA Pin Value	0x0000_XXXX
PB_PIN	GPIO_BA+0x050	R	PB Pin Value	0x0000_XXXX
PC_PIN	GPIO_BA+0x090	R	PC Pin Value	0x0000_XXXX
PD_PIN	GPIO_BA+0x0D0	R	PD Pin Value	0x0000_XXXX
PE_PIN	GPIO_BA+0x110	R	PE Pin Value	0x0000_XXXX
PF_PIN	GPIO_BA+0x150	R	PF Pin Value	0x0000_XXXX
PG_PIN	GPIO_BA+0x190	R	PG Pin Value	0x0000_XXXX
PH_PIN	GPIO_BA+0x1D0	R	PH Pin Value	0x0000_XXXX
PI_PIN	GPIO_BA+0x210	R	PI Pin Value	0x0000_XXXX
PJ_PIN	GPIO_BA+0x250	R	PJ Pin Value	0x0000_XXXX
PK_PIN	GPIO_BA+0x290	R	PK Pin Value	0x0000_XXXX
PL_PIN	GPIO_BA+0x2D0	R	PL Pin Value	0x0000_XXXX
PM_PIN	GPIO_BA+0x310	R	PM Pin Value	0x0000_XXXX
PN_PIN	GPIO_BA+0x350	R	PN Pin Value	0x0000_XXXX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PIN							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PIN							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[n] n=0,1..15	**PINn**	**Port A-N Pin[n] Pin Value** Each bit of the register reflects the actual status of the respective Px.n pin. If the bit is 1, it indicates the corresponding pin status is high; else the pin status is low.

Port A-N De-bounce Enable Control Register (Px_DBEN)

Register	Offset	R/W	Description	Reset Value
PA_DBEN	GPIO_BA+0x014	R/W	PA De-bounce Enable Control Register	0x0000_0000
PB_DBEN	GPIO_BA+0x054	R/W	PB De-bounce Enable Control Register	0x0000_0000
PC_DBEN	GPIO_BA+0x094	R/W	PC De-bounce Enable Control Register	0x0000_0000
PD_DBEN	GPIO_BA+0x0D4	R/W	PD De-bounce Enable Control Register	0x0000_0000
PE_DBEN	GPIO_BA+0x114	R/W	PE De-bounce Enable Control Register	0x0000_0000
PF_DBEN	GPIO_BA+0x154	R/W	PF De-bounce Enable Control Register	0x0000_0000
PG_DBEN	GPIO_BA+0x194	R/W	PG De-bounce Enable Control Register	0x0000_0000
PH_DBEN	GPIO_BA+0x1D4	R/W	PH De-bounce Enable Control Register	0x0000_0000
PI_DBEN	GPIO_BA+0x214	R/W	PI De-bounce Enable Control Register	0x0000_0000
PJ_DBEN	GPIO_BA+0x254	R/W	PJ De-bounce Enable Control Register	0x0000_0000
PK_DBEN	GPIO_BA+0x294	R/W	PK De-bounce Enable Control Register	0x0000_0000
PL_DBEN	GPIO_BA+0x2D4	R/W	PL De-bounce Enable Control Register	0x0000_0000
PM_DBEN	GPIO_BA+0x314	R/W	PM De-bounce Enable Control Register	0x0000_0000
PN_DBEN	GPIO_BA+0x354	R/W	PN De-bounce Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DBEN							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DBEN							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[n] n=0,1..15	**DBENn**	**Port A-N Pin[n] Input Signal De-bounce Enable Bit** The DBEN[n] bit is used to enable the de-bounce function for each corresponding bit. If the input signal pulse width cannot be sampled by continuous two de-bounce sample cycle, the input signal transition is seen as the signal bounce and will not trigger the interrupt. The de-bounce clock source is controlled by DBCLKSRC (Px_DBCTL [4]), one de-bounce sample cycle period is controlled by DBCLKSEL (Px_DBCTL [3:0]). 0 = Px.n de-bounce function Disabled.

		1 = Px.n de-bounce function Enabled. The de-bounce function is valid only for edge triggered interrupt. If the interrupt mode is level triggered, the de-bounce enable bit is ignored.

Port A-N Interrupt Type Control (Px_INTTYPE)

Register	Offset	R/W	Description	Reset Value
PA_INTTYPE	GPIO_BA+0x018	R/W	PA Interrupt Trigger Type Control	0x0000_0000
PB_INTTYPE	GPIO_BA+0x058	R/W	PB Interrupt Trigger Type Control	0x0000_0000
PC_INTTYPE	GPIO_BA+0x098	R/W	PC Interrupt Trigger Type Control	0x0000_0000
PD_INTTYPE	GPIO_BA+0x0D8	R/W	PD Interrupt Trigger Type Control	0x0000_0000
PE_INTTYPE	GPIO_BA+0x118	R/W	PE Interrupt Trigger Type Control	0x0000_0000
PF_INTTYPE	GPIO_BA+0x158	R/W	PF Interrupt Trigger Type Control	0x0000_0000
PG_INTTYPE	GPIO_BA+0x198	R/W	PG Interrupt Trigger Type Control	0x0000_0000
PH_INTTYPE	GPIO_BA+0x1D8	R/W	PH Interrupt Trigger Type Control	0x0000_0000
PI_INTTYPE	GPIO_BA+0x218	R/W	PI Interrupt Trigger Type Control	0x0000_0000
PJ_INTTYPE	GPIO_BA+0x258	R/W	PJ Interrupt Trigger Type Control	0x0000_0000
PK_INTTYPE	GPIO_BA+0x298	R/W	PK Interrupt Trigger Type Control	0x0000_0000
PL_INTTYPE	GPIO_BA+0x2D8	R/W	PL Interrupt Trigger Type Control	0x0000_0000
PM_INTTYPE	GPIO_BA+0x318	R/W	PM Interrupt Trigger Type Control	0x0000_0000
PN_INTTYPE	GPIO_BA+0x358	R/W	PN Interrupt Trigger Type Control	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TYPE							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TYPE							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[n] n=0,1..15	**TYPEn**	**Port A-N Pin[n] Edge or Level Detection Interrupt Trigger Type Control** TYPE (Px_INTTYPE[n]) bit is used to control the triggered interrupt is by level trigger or by edge trigger. If the interrupt is by edge trigger, the trigger source can be controlled by de-bounce. If the interrupt is by level trigger, the input source is sampled by one HCLK clock and generates the interrupt. 0 = Edge trigger interrupt. 1 = Level trigger interrupt.

		If the pin is set as the level trigger interrupt, only one level can be set on the registers RHIEN (Px_INTEN[n+16])/FLIEN (Px_INTEN[n]). If both levels to trigger interrupt are set, the setting is ignored and no interrupt will occur. The de-bounce function is valid only for edge triggered interrupt. If the interrupt mode is level triggered, the de-bounce enable bit is ignored.

Port A-N Interrupt Enable Control Register (Px_INTEN)

Register	Offset	R/W	Description	Reset Value
PA_INTEN	GPIO_BA+0x01C	R/W	PA Interrupt Enable Control Register	0x0000_0000
PB_INTEN	GPIO_BA+0x05C	R/W	PB Interrupt Enable Control Register	0x0000_0000
PC_INTEN	GPIO_BA+0x09C	R/W	PC Interrupt Enable Control Register	0x0000_0000
PD_INTEN	GPIO_BA+0x0DC	R/W	PD Interrupt Enable Control Register	0x0000_0000
PE_INTEN	GPIO_BA+0x11C	R/W	PE Interrupt Enable Control Register	0x0000_0000
PF_INTEN	GPIO_BA+0x15C	R/W	PF Interrupt Enable Control Register	0x0000_0000
PG_INTEN	GPIO_BA+0x19C	R/W	PG Interrupt Enable Control Register	0x0000_0000
PH_INTEN	GPIO_BA+0x1DC	R/W	PH Interrupt Enable Control Register	0x0000_0000
PI_INTEN	GPIO_BA+0x21C	R/W	PI Interrupt Enable Control Register	0x0000_0000
PJ_INTEN	GPIO_BA+0x25C	R/W	PJ Interrupt Enable Control Register	0x0000_0000
PK_INTEN	GPIO_BA+0x29C	R/W	PK Interrupt Enable Control Register	0x0000_0000
PL_INTEN	GPIO_BA+0x2DC	R/W	PL Interrupt Enable Control Register	0x0000_0000
PM_INTEN	GPIO_BA+0x31C	R/W	PM Interrupt Enable Control Register	0x0000_0000
PN_INTEN	GPIO_BA+0x35C	R/W	PN Interrupt Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
RHIEN							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RHIEN							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
FLIEN							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FLIEN							

Bits	Description	
[n+16] n=0,1..15	**RHIENn**	**Port A-N Pin[n] Rising Edge or High Level Interrupt Trigger Type Enable Bit** The RHIEN (Px_INTEN[n+16]) bit is used to enable the interrupt for each of the corresponding input Px.n pin. Set bit to 1 also enable the pin wake-up function. When setting the RHIEN (Px_INTEN[n+16]) bit to 1 : If the interrupt is level trigger (TYPE (Px_INTTYPE[n]) bit is set to 1), the input Px.n pin will generate the interrupt while this pin state is at high level. If the interrupt is edge trigger (TYPE (Px_INTTYPE[n]) bit is set to 0), the input Px.n pin will generate the interrupt while this pin state changed from low to high.

		0 = Px.n level high or low to high interrupt Disabled. 1 = Px.n level high or low to high interrupt Enabled.
[n] n=0,1..15	**FLIENn**	**Port A-N Pin[n] Falling Edge or Low Level Interrupt Trigger Type Enable Bit** The FLIEN (Px_INTEN[n]) bit is used to enable the interrupt for each of the corresponding input Px.n pin. Set bit to 1 also enable the pin wake-up function. When setting the FLIEN (Px_INTEN[n]) bit to 1 : If the interrupt is level trigger (TYPE (Px_INTTYPE[n]) bit is set to 1), the input Px.n pin will generate the interrupt while this pin state is at low level. If the interrupt is edge trigger(TYPE (Px_INTTYPE[n]) bit is set to 0), the input Px.n pin will generate the interrupt while this pin state changed from high to low. 0 = Px.n level low or high to low interrupt Disabled. 1 = Px.n level low or high to low interrupt Enabled.

Port A-N Interrupt Source Flag (Px_INTSRC)

Register	Offset	R/W	Description	Reset Value
PA_INTSRC	GPIO_BA+0x020	R/W	PA Interrupt Source Flag	0x0000_XXXX
PB_INTSRC	GPIO_BA+0x060	R/W	PB Interrupt Source Flag	0x0000_XXXX
PC_INTSRC	GPIO_BA+0x0A0	R/W	PC Interrupt Source Flag	0x0000_XXXX
PD_INTSRC	GPIO_BA+0x0E0	R/W	PD Interrupt Source Flag	0x0000_XXXX
PE_INTSRC	GPIO_BA+0x120	R/W	PE Interrupt Source Flag	0x0000_XXXX
PF_INTSRC	GPIO_BA+0x160	R/W	PF Interrupt Source Flag	0x0000_XXXX
PG_INTSRC	GPIO_BA+0x1A0	R/W	PG Interrupt Source Flag	0x0000_XXXX
PH_INTSRC	GPIO_BA+0x1E0	R/W	PH Interrupt Source Flag	0x0000_XXXX
PI_INTSRC	GPIO_BA+0x220	R/W	PI Interrupt Source Flag	0x0000_XXXX
PJ_INTSRC	GPIO_BA+0x260	R/W	PJ Interrupt Source Flag	0x0000_XXXX
PK_INTSRC	GPIO_BA+0x2A0	R/W	PK Interrupt Source Flag	0x0000_XXXX
PL_INTSRC	GPIO_BA+0x2E0	R/W	PL Interrupt Source Flag	0x0000_XXXX
PM_INTSRC	GPIO_BA+0x320	R/W	PM Interrupt Source Flag	0x0000_XXXX
PN_INTSRC	GPIO_BA+0x360	R/W	PN Interrupt Source Flag	0x0000_XXXX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
INTSRC							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INTSRC							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[n] n=0,1..15	**INTSRCn**	**Port A-N Pin[n] Interrupt Source Flag** Write Operation: 0 = No action. 1 = Clear the corresponding pending interrupt. Read Operation: 0 = No interrupt at Px.n.

		1 = Px.n generates an interrupt.

Port A-N Input Schmitt Trigger Enable Register (Px_SMTEN)

Register	Offset	R/W	Description	Reset Value
PA_SMTEN	GPIO_BA+0x024	R/W	PA Input Schmitt Trigger Enable Register	0x0000_0000
PB_SMTEN	GPIO_BA+0x064	R/W	PB Input Schmitt Trigger Enable Register	0x0000_0000
PC_SMTEN	GPIO_BA+0x0A4	R/W	PC Input Schmitt Trigger Enable Register	0x0000_0000
PD_SMTEN	GPIO_BA+0x0E4	R/W	PD Input Schmitt Trigger Enable Register	0x0000_0000
PE_SMTEN	GPIO_BA+0x124	R/W	PE Input Schmitt Trigger Enable Register	0x0000_0000
PF_SMTEN	GPIO_BA+0x164	R/W	PF Input Schmitt Trigger Enable Register	0x0000_0000
PG_SMTEN	GPIO_BA+0x1A4	R/W	PG Input Schmitt Trigger Enable Register	0x0000_0000
PH_SMTEN	GPIO_BA+0x1E4	R/W	PH Input Schmitt Trigger Enable Register	0x0000_0000
PI_SMTEN	GPIO_BA+0x224	R/W	PI Input Schmitt Trigger Enable Register	0x0000_0000
PJ_SMTEN	GPIO_BA+0x264	R/W	PJ Input Schmitt Trigger Enable Register	0x0000_0000
PK_SMTEN	GPIO_BA+0x2A4	R/W	PK Input Schmitt Trigger Enable Register	0x0000_0000
PL_SMTEN	GPIO_BA+0x2E4	R/W	PL Input Schmitt Trigger Enable Register	0x0000_0000
PM_SMTEN	GPIO_BA+0x324	R/W	PM Input Schmitt Trigger Enable Register	0x0000_0000
PN_SMTEN	GPIO_BA+0x364	R/W	PN Input Schmitt Trigger Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SMTEN							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SMTEN							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[n] n=0,1..15	**SMTENn**	**Port A-N Pin[n] Input Schmitt Trigger Enable Bit** 0 = Px.n input schmitt trigger function Disabled. 1 = Px.n input schmitt trigger function Enabled.

Port A-N High Slew Rate Control Register (Px_SLEWCTL)

Register	Offset	R/W	Description	Reset Value
PA_SLEWCTL	GPIO_BA+0x028	R/W	PA High Slew Rate Control Register	0x0000_0000
PB_SLEWCTL	GPIO_BA+0x068	R/W	PB High Slew Rate Control Register	0x0000_0000
PC_SLEWCTL	GPIO_BA+0x0A8	R/W	PC High Slew Rate Control Register	0x0000_0000
PD_SLEWCTL	GPIO_BA+0x0E8	R/W	PD High Slew Rate Control Register	0x0000_0000
PE_SLEWCTL	GPIO_BA+0x128	R/W	PE High Slew Rate Control Register	0x0000_0000
PF_SLEWCTL	GPIO_BA+0x168	R/W	PF High Slew Rate Control Register	0x0000_0000
PG_SLEWCTL	GPIO_BA+0x1A8	R/W	PG High Slew Rate Control Register	0x0000_0000
PH_SLEWCTL	GPIO_BA+0x1E8	R/W	PH High Slew Rate Control Register	0x0000_0000
PI_SLEWCTL	GPIO_BA+0x228	R/W	PI High Slew Rate Control Register	0x0000_0000
PJ_SLEWCTL	GPIO_BA+0x268	R/W	PJ High Slew Rate Control Register	0x0000_0000
PK_SLEWCTL	GPIO_BA+0x2A8	R/W	PK High Slew Rate Control Register	0x0000_0000
PL_SLEWCTL	GPIO_BA+0x2E8	R/W	PL High Slew Rate Control Register	0x0000_0000
PM_SLEWCTL	GPIO_BA+0x328	R/W	PM High Slew Rate Control Register	0x0000_0000
PN_SLEWCTL	GPIO_BA+0x368	R/W	PN High Slew Rate Control Register	0x0000_0000

31	30	29	28	27	26	25	24
HSREN15		HSREN14		HSREN13		HSREN12	
23	22	21	20	19	18	17	16
HSREN11		HSREN10		HSREN9		HSREN8	
15	14	13	12	11	10	9	8
HSREN07		HSREN6		HSREN5		HSREN4	
7	6	5	4	3	2	1	0
HSREN03		HSREN2		HSREN1		HSREN0	

Bits	Description	
[2n+1:2n] n=0,1..15	**HSRENn**	**Port A-N Pin[n] High Slew Rate Control** 00 = Px.n output with normal slew rate mode. 01 = Px.n output with high slew rate mode . 10 = Reserved. 11 = Reserved.

Port B~G and J and K and N Power Mode Selection Register (Px_SPW)

Register	Offset	R/W	Description	Reset Value
PB_SPW	GPIO_BA+0x06C	R/W	PB Power Mode Selection Register	0x0000_FFFF
PC_SPW	GPIO_BA+0x0AC	R/W	PC Power Mode Selection Register	0x0000_FFFF
PD_SPW	GPIO_BA+0x0EC	R/W	PD Power Mode Selection Register	0x0000_FFFF
PE_SPW	GPIO_BA+0x12C	R/W	PE Power Mode Selection Register	0x0000_FFFF
PF_SPW	GPIO_BA+0x16C	R/W	PF Power Mode Selection Register	0x0000_FFFF
PG_SPW	GPIO_BA+0x1AC	R/W	PG Power Mode Selection Register	0x0000_FFFF
PJ_SPW	GPIO_BA+0x26C	R/W	PJ Power Mode Selection Register	0x0000_FFFF
PK_SPW	GPIO_BA+0x2AC	R/W	PK Power Mode Selection Register	0x0000_FFFF
PN_SPW	GPIO_BA+0x36C	R/W	PN Power Mode Selection Register	0x0000_FFFF

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SPW							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SPW							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[n] n=0,1..15	**SPWn**	**Port B-G,J,K,N Pin[n] Power Mode Selection Bit** 0 = Px.n pad power voltage selection is 1.8V. 1 = Px.n pad power voltage selection is 3.3V. **Note :** The Power Mode Selection bit is for different application. Only PB.8~15, PD.1, PE.2~13, PF.0~13, PG.10, PJ.0~11, PK.9~10, PN.10~11 pad voltage can be selected.

Port A-N Pull-up and Pull-down Selection Register (Px_PUSEL)

Register	Offset	R/W	Description	Reset Value
PA_PUSEL	GPIO_BA+0x030	R/W	PA Pull-up and Pull-down Selection Register	0x0000_0000
PB_PUSEL	GPIO_BA+0x070	R/W	PB Pull-up and Pull-down Selection Register	0x0000_0000
PC_PUSEL	GPIO_BA+0x0B0	R/W	PC Pull-up and Pull-down Selection Register	0x0000_0000
PD_PUSEL	GPIO_BA+0x0F0	R/W	PD Pull-up and Pull-down Selection Register	0x0000_0000
PE_PUSEL	GPIO_BA+0x130	R/W	PE Pull-up and Pull-down Selection Register	0x0000_0000
PF_PUSEL	GPIO_BA+0x170	R/W	PF Pull-up and Pull-down Selection Register	0x0000_0000
PG_PUSEL	GPIO_BA+0x1B0	R/W	PG Pull-up and Pull-down Selection Register	0x0000_0000
PH_PUSEL	GPIO_BA+0x1F0	R/W	PH Pull-up and Pull-down Selection Register	0x0000_0000
PI_PUSEL	GPIO_BA+0x230	R/W	PI Pull-up and Pull-down Selection Register	0x0000_0000
PJ_PUSEL	GPIO_BA+0x270	R/W	PJ Pull-up and Pull-down Selection Register	0x0000_0000
PK_PUSEL	GPIO_BA+0x2B0	R/W	PK Pull-up and Pull-down Selection Register	0x0000_0000
PL_PUSEL	GPIO_BA+0x2F0	R/W	PL Pull-up and Pull-down Selection Register	0x0000_0000
PM_PUSEL	GPIO_BA+0x330	R/W	PM Pull-up and Pull-down Selection Register	0x0000_0000
PN_PUSEL	GPIO_BA+0x370	R/W	PN Pull-up and Pull-down Selection Register	0x0000_0000

31	30	29	28	27	26	25	24
PUSEL15		PUSEL14		PUSEL13		PUSEL12	
23	22	21	20	19	18	17	16
PUSEL11		PUSEL10		PUSEL9		PUSEL8	
15	14	13	12	11	10	9	8
PUSEL7		PUSEL6		PUSEL5		PUSEL4	
7	6	5	4	3	2	1	0
PUSEL3		PUSEL2		PUSEL1		PUSEL0	

Bits	Description	
[2n+1:2n] n=0,1..15	**PUSELn**	**Port A-N Pin[n] Pull-up and Pull-down Enable Register** Determine each I/O Pull-up/pull-down of Px.n pins. 00 = Px.n pull-up and pull-down disable. 01 = Px.n pull-up enable. 10 = Px.n pull-down enable. 11 = Px.n pull-up and pull-down disable. **Note:**

		Basically, the pull-up control and pull-down control has following behavior limitation. The independent pull-up control register only valid when MODEn is set as input and open-drain mode even if I/O function is switched to multi-function pin, e.g. UARTx_RXD. The independent pull-down control register is only valid when MODEn set as tri-state mode. When both pull-up pull-down is set as 1 at “tri-state” mode, keep I/O in tri-state mode.

Port A-N Interrupt De-bounce Control Register (Px_DBCTL)

Register	Offset	R/W	Description	Reset Value
PA_DBCTL	GPIO_BA+0x034	R/W	PA Interrupt De-bounce Control Register	0x0000_0020
PB_DBCTL	GPIO_BA+0x074	R/W	PB Interrupt De-bounce Control Register	0x0000_0020
PC_DBCTL	GPIO_BA+0x0B4	R/W	PC Interrupt De-bounce Control Register	0x0000_0020
PD_DBCTL	GPIO_BA+0x0F4	R/W	PD Interrupt De-bounce Control Register	0x0000_0020
PE_DBCTL	GPIO_BA+0x134	R/W	PE Interrupt De-bounce Control Register	0x0000_0020
PF_DBCTL	GPIO_BA+0x174	R/W	PF Interrupt De-bounce Control Register	0x0000_0020
PG_DBCTL	GPIO_BA+0x1B4	R/W	PG Interrupt De-bounce Control Register	0x0000_0020
PH_DBCTL	GPIO_BA+0x1F4	R/W	PH Interrupt De-bounce Control Register	0x0000_0020
PI_DBCTL	GPIO_BA+0x234	R/W	PI Interrupt De-bounce Control Register	0x0000_0020
PJ_DBCTL	GPIO_BA+0x274	R/W	PJ Interrupt De-bounce Control Register	0x0000_0020
PK_DBCTL	GPIO_BA+0x2B4	R/W	PK Interrupt De-bounce Control Register	0x0000_0020
PL_DBCTL	GPIO_BA+0x2F4	R/W	PL Interrupt De-bounce Control Register	0x0000_0020
PM_DBCTL	GPIO_BA+0x334	R/W	PM Interrupt De-bounce Control Register	0x0000_0020
PN_DBCTL	GPIO_BA+0x374	R/W	PN Interrupt De-bounce Control Register	0x0000_0020

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		ICLKON	DBCLKSRC	DBCLKSEL			

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**ICLKON**	**Interrupt Clock on Mode** 0 = Edge detection circuit is active only if I/O pin corresponding RHIEN (Px_INTEN[n+16])/FLIEN (Px_INTEN[n]) bit is set to 1. 1 = All I/O pins edge detection circuit is always active after reset. **Note:** It is recommended to disable this bit to save system power if no special application concern.

Bits	Description	
[4]	**DBCLKSRC**	**De-bounce Counter Clock Source Selection** 0 = De-bounce counter clock source is the HXT. 1 = De-bounce counter clock source is the LIRC. **Note:** This bit is reserved if the chip package without LIRC. The de-bounce counter clock source is only from HXT. And setting this bit does not guarantee what will occur.
[3:0]	**DBCLKSEL**	**De-bounce Sampling Cycle Selection** 0000 = Sample interrupt input once per 1 clocks. 0001 = Sample interrupt input once per 2 clocks. 0010 = Sample interrupt input once per 4 clocks. 0011 = Sample interrupt input once per 8 clocks. 0100 = Sample interrupt input once per 16 clocks. 0101 = Sample interrupt input once per 32 clocks. 0110 = Sample interrupt input once per 64 clocks. 0111 = Sample interrupt input once per 128 clocks. 1000 = Sample interrupt input once per 256 clocks. 1001 = Sample interrupt input once per 2*256 clocks. 1010 = Sample interrupt input once per 4*256 clocks. 1011 = Sample interrupt input once per 8*256 clocks. 1100 = Sample interrupt input once per 16*256 clocks. 1101 = Sample interrupt input once per 32*256 clocks. 1110 = Sample interrupt input once per 64*256 clocks. 1111 = Sample interrupt input once per 128*256 clocks.

Port A-N Low Byte Driver Strength Control Register (Px_DSL)

Register	Offset	R/W	Description	Reset Value
PA_DSL	GPIO_BA+0x038	R/W	PA Low Byte Driver Strength Control Register	0x0000_0000
PB_DSL	GPIO_BA+0x078	R/W	PB Low Byte Driver Strength Control Register	0x0000_0000
PC_DSL	GPIO_BA+0x0B8	R/W	PC Low Byte Driver Strength Control Register	0x0000_0000
PD_DSL	GPIO_BA+0x0F8	R/W	PD Low Byte Driver Strength Control Register	0x0000_0000
PE_DSL	GPIO_BA+0x138	R/W	PE Low Byte Driver Strength Control Register	0x0000_0000
PF_DSL	GPIO_BA+0x178	R/W	PF Low Byte Driver Strength Control Register	0x0000_0000
PG_DSL	GPIO_BA+0x1B8	R/W	PG Low Byte Driver Strength Control Register	0x0000_0000
PH_DSL	GPIO_BA+0x1F8	R/W	PH Low Byte Driver Strength Control Register	0x0000_0000
PI_DSL	GPIO_BA+0x238	R/W	PI Low Byte Driver Strength Control Register	0x0000_0000
PJ_DSL	GPIO_BA+0x278	R/W	PJ Low Byte Driver Strength Control Register	0x0000_0000
PK_DSL	GPIO_BA+0x2B8	R/W	PK Low Byte Driver Strength Control Register	0x0000_0000
PL_DSL	GPIO_BA+0x2F8	R/W	PL Low Byte Driver Strength Control Register	0x0000_0000
PM_DSL	GPIO_BA+0x338	R/W	PM Low Byte Driver Strength Control Register	0x0000_0000
PN_DSL	GPIO_BA+0x378	R/W	PN Low Byte Driver Strength Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	**DS7**			**Reserved**	**DS6**		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved	**DS5**			**Reserved**	**DS4**		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**DS3**			**Reserved**	**DS2**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**DS1**			**Reserved**	**DS0**		

Bits	Description	
[n] n=0,1..7	**DSn**	**Port A-N Low Byte pin[n] Driver Strength Control Register** Determine each driver strength of Px.n low byte pins. 000 = Px.n is minimum deiver strength. 111 = Px.n is maximum deiver strength.

Port A-N High Byte Strength Control Register (Px_DSH)

Register	Offset	R/W	Description	Reset Value
PA_DSH	GPIO_BA+0x03C	R/W	PA High Byte Driver Strength Control Register	0x0000_0000
PB_DSH	GPIO_BA+0x07C	R/W	PB High Byte Driver Strength Control Register	0x0000_0000
PC_DSH	GPIO_BA+0x0BC	R/W	PC High Byte Driver Strength Control Register	0x0000_0000
PD_DSH	GPIO_BA+0x0FC	R/W	PD High Byte Driver Strength Control Register	0x0000_0000
PE_DSH	GPIO_BA+0x13C	R/W	PE High Byte Driver Strength Control Register	0x0000_0000
PF_DSH	GPIO_BA+0x17C	R/W	PF High Byte Driver Strength Control Register	0x0000_0000
PG_DSH	GPIO_BA+0x1BC	R/W	PG High Byte Driver Strength Control Register	0x0000_0000
PH_DSH	GPIO_BA+0x1FC	R/W	PH High Byte Driver Strength Control Register	0x0000_0000
PI_DSH	GPIO_BA+0x23C	R/W	PI High Byte Driver Strength Control Register	0x0000_0000
PJ_DSH	GPIO_BA+0x27C	R/W	PJ High Byte Driver Strength Control Register	0x0000_0000
PK_DSH	GPIO_BA+0x2BC	R/W	PK High Byte Driver Strength Control Register	0x0000_0000
PL_DSH	GPIO_BA+0x2FC	R/W	PL High Byte Driver Strength Control Register	0x0000_0000
PM_DSH	GPIO_BA+0x33C	R/W	PM High Byte Driver Strength Control Register	0x0000_0000
PN_DSH	GPIO_BA+0x37C	R/W	PN High Byte Driver Strength Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	**DS15**			**Reserved**	**DS14**		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved	**DS13**			**Reserved**	**DS12**		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**DS11**			**Reserved**	**DS10**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**DS9**			**Reserved**	**DS8**		

Bits	Description	
[n] n=8,9..15	**DSn**	**Port A-N High Byte pin[n] Driver Strength Control Register** Determine each driver strength of Px.n high byte pins. 000 = Px.n is minimum deiver strength. 111 = Px.n is maximum deiver strength.

GPIO Px.n Pin Data Input/Outut Register (Pxn_PDIO)

Register	Offset	R/W	Description	Reset Value
PAn_PDIO n=0,1..15	GPIO_BA+0x800+(0x04 * n)	R/W	GPIO PA.n Pin Data Input/Output Register	0x0000_000X
PBn_PDIO n=0,1..15	GPIO_BA+0x840+(0x04 * n)	R/W	GPIO PB.n Pin Data Input/Output Register	0x0000_000X
PCn_PDIO n=0,1..15	GPIO_BA+0x880+(0x04 * n)	R/W	GPIO PC.n Pin Data Input/Output Register	0x0000_000X
PDn_PDIO n=0,1..15	GPIO_BA+0x8C0+(0x04 * n)	R/W	GPIO PD.n Pin Data Input/Output Register	0x0000_000X
PEn_PDIO n=0,1..15	GPIO_BA+0x900+(0x04 * n)	R/W	GPIO PE.n Pin Data Input/Output Register	0x0000_000X
PFn_PDIO n=0,1..15	GPIO_BA+0x940+(0x04 * n)	R/W	GPIO PF.n Pin Data Input/Output Register	0x0000_000X
PGn_PDIO n=0,1..15	GPIO_BA+0x980+(0x04 * n)	R/W	GPIO PG.n Pin Data Input/Output Register	0x0000_000X
PHn_PDIO n=0,1..15	GPIO_BA+0x9C0+(0x04 * n)	R/W	GPIO PH.n Pin Data Input/Output Register	0x0000_000X
PIn_PDIO n=0,1..15	GPIO_BA+0xA00+(0x04 * n)	R/W	GPIO PI.n Pin Data Input/Output Register	0x0000_000X
PJn_PDIO n=0,1..15	GPIO_BA+0xA40+(0x04 * n)	R/W	GPIO PJ.n Pin Data Input/Output Register	0x0000_000X
PKn_PDIO n=0,1..15	GPIO_BA+0xA80+(0x04 * n)	R/W	GPIO PK.n Pin Data Input/Output Register	0x0000_000X
PLn_PDIO n=0,1..15	GPIO_BA+0xAC0+(0x04 * n)	R/W	GPIO PL.n Pin Data Input/Output Register	0x0000_000X
PMn_PDIO n=0,1..15	GPIO_BA+0xB00+(0x04 * n)	R/W	GPIO PM.n Pin Data Input/Output Register	0x0000_000X
PNn_PDIO n=0,1..15	GPIO_BA+0xB40+(0x04 * n)	R/W	GPIO PN.n Pin Data Input/Output Register	0x0000_000X

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							PDIO

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**PDIO**	**GPIO Px.n Pin Data Input/Output** Writing this bit can control one GPIO pin output value. 0 = Corresponding GPIO pin set to low. 1 = Corresponding GPIO pin set to high. Read this register to get GPIO pin status. For example, writing PA0_PDIO will reflect the written value to bit DOUT (Px_DOUT[0]), reading PA0_PDIO will return the value of PIN (PA_PIN[0]). **Note:** The writing operation will not be affected by register DATMSK (Px_DATMSK[n]).

6.16 PDMA Controller (PDMA)

6.16.1 Overview

The peripheral direct memory access (PDMA) controller is used to provide high-speed data transfer. The PDMA controller can transfer data from one address to another without CPU intervention. This has the benefit of reducing the workload of CPU and keeps CPU resources free for other applications. There are four PDMA controller, PDMA0 ~ PDMA3. Both PDMA0 and PDMA1 can be set as secure or non-secure. On the other side, both PDMA2 and PDMA3 can be configured as non-secure or SubM domain. Each PDMA controller has a total of 10 channels and each channel can perform transfer between memory and peripherals or between memory and memory.

6.16.2 Features

- Supports 10 independently configurable channels
- Supports 2 selectable levels of priority
  - Fixed priority
  - Round-robin priority
- Supports 4 PDMA controllers:
  - PDMA0 and PDMA1 can be configured as secure or non-secure PDMA
  - PDMA2 and PDMA3 can be configured as non-secure or SubM PDMA
- Supports transfer data width of 8, 16, and 32 bits
- Supports source and destination address increment size can be byte, half-word, word or no increment
- Supports software and UART, QSPI, SPI, EPWM, I^2C, I^2S, Timer and EADC request
- Supports Scatter-gather mode to perform sophisticated transfer through the use of the descriptor link list table
- Supports single and burst transfer type
- Supports time-out function
- Supports stride function from channel 0 to channel 5
- Supports enhanced stride function on channel 0 and channel 1

6.16.3 Block Diagram

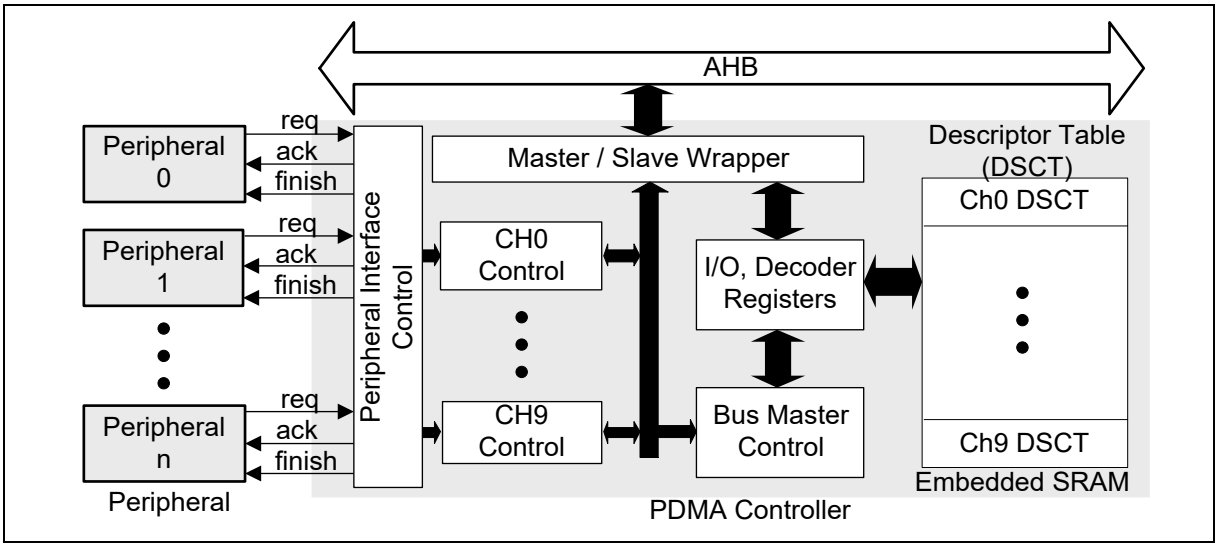


Figure 6.16-1 PDMA Controller Block Diagram

6.16.4 Basic Configuration

6.16.4.1 PDMA0 Basic Configuration

- Clock Source Configuration
  - Enable PDMA0 controller clock in PDMA0EN(CLK_SYSCLK1[0]).
- Reset Configuration
  - Reset PDMA0 controller in PDMA0RST (SYS_IPRST0[4]).

6.16.4.2 PDMA1 Basic Configuration

- Clock Source Configuration
  - Enable PDMA1 controller clock in PDMA1EN (CLK_SYSCLK1[1]).
- Reset Configuration
  - Reset PDMA1 controller in PDMA1RST (SYS_IPRST0[5]).

6.16.4.3 PDMA2 Basic Configuration

- Clock Source Configuration
  - Enable PDMA2 controller clock in PDMA2EN (CLK_SYSCLK1[2]).
- Reset Configuration
  - Reset PDMA2 controller in PDMA2RST (SYS_IPRST0[6]).

6.16.4.4 PDMA3 Basic Configuration

- Clock Source Configuration
  - Enable PDMA3 controller clock in PDMA3EN (CLK_SYSCLK1[3]).
- Reset Configuration
  - Reset PDMA3 controller in PDMA3RST (SYS_IPRST0[7]).

6.16.5 Functional Description

The PDMA controller transfers data from one address to another without CPU intervention. The PDMA controller supports 10 independent channels and serves only one channel at one time, as the result, PDMA controller supports two level channel priorities: fixed and round-robin priority, PDMA controller serves channel in order from highest to lowest priority channel. The PDMA controller supports two operation modes: Basic mode and Scatter-gather mode. Basic mode is used to perform one descriptor table transfer. Scatter-gather mode has more entries for each PDMA channel, and thus the PDMA controller supports sophisticated transfer through the entries. The descriptor table entry data structure contains many transfer information including the transfer source address, transfer destination address, transfer count, burst size, transfer type and operation mode. Figure 6.16-2 shows the diagram of descriptor table (DSCT) data structure.

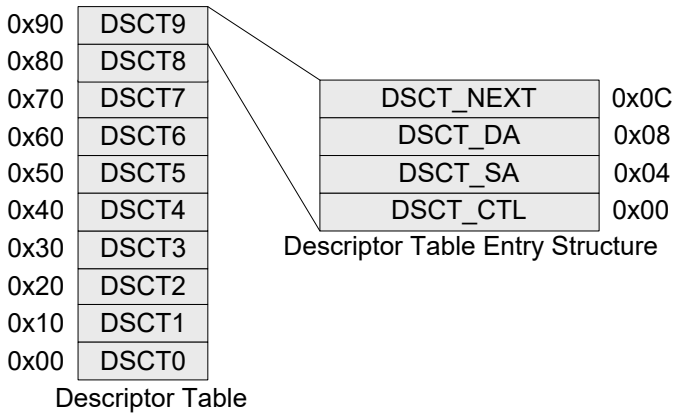


Figure 6.16-2 Descriptor Table Entry Structure

The PDMA controller also supports single and burst transfer type and the request source can be from software or peripheral request, transfer between memory to memory using software request. A single transfer means that software or peripheral is ready to transfer one data (every data needs one request), and the burst transfer means that software or peripherals will transfer multiple data (multiple data only need one request).

6.16.5.1 Channel Priority

The PDMA controller supports two level channel priorities including fixed and round-robin priority. The fixed priority channel has higher priority than round-robin priority channel. If multiple channels are set as fixed or round-robin priority, the higher channel will have higher priority. The priority order is listed in Table 6.16-1.

PDMA_PRISET	Channel Number	Priority Setting	Arbitration Priority In Descending Order
1	9	Channel9, Fixed Priority	Highest
1	8	Channel8, Fixed Priority	---
---	---	---	---
1	0	Channel0, Fixed Priority	---
0	9	Channel9, Round-Robin Priority	---
0	8	Channel8, Round-Robin Priority	---

---	---	---	---
0	0	Channel0, Round-Robin Priority	Lowest

Table 6.16-1 Channel Priority Table

6.16.5.2 PDMA Operation Mode

The PDMA controller supports two operation modes including Basic mode and Scatter-gather mode.

Basic Mode

Basic mode is used to perform one descriptor table transfer mode. This mode can be used to transfer data between memory and memory, peripherals and memory or peripherals and peripherals, but if user want to transfer data between peripherals and peripherals, one thing must be sure is that the request from peripherals knows that the data is ready for transfer or not. PDMA controller operation mode can be set from OPMODE (PDMA_DSCTn_CTL[1:0], n denotes PDMA channel), the default setting is in idle state (OPMODE (PDMA_DSCTn_CTL[1:0]) = 0x0) and recommend user configure the descriptor table in idle state. If operation mode is not in idle state, user re-configure channel setting may make some operation error.

User must fill the transfer count TXCNT (PDMA_DSCTn_CTL[31:16]) register and select transfer width TXWIDTH (PDMA_DSCTn_CTL[13:12]), destination address increment size DAINC (PDMA_DSCTn_CTL[11:10]), source address increment size SAINC (PDMA_DSCTn_CTL[9:8]), burst size BURSIZE (PDMA_DSCTn_CTL[6:4]) and transfer type TXTYPE (PDMA_DSCTn_CTL[2]), then the PDMA controller will perform transfer operation in transfer state after receiving request signal. Finishing this task will generate an interrupt to CPU if corresponding PDMA interrupt bit INTENn (PDMA_INTEN[7:0]) is enabled and the operation mode will be updated to idle state as shown in Figure 6.16-3. If software configures the operation mode to idle state, the PDMA controller will not perform any transfer and then clear this operation request. Finishing this task will also generate an interrupt to CPU if corresponding PDMA interrupt bit is enabled.

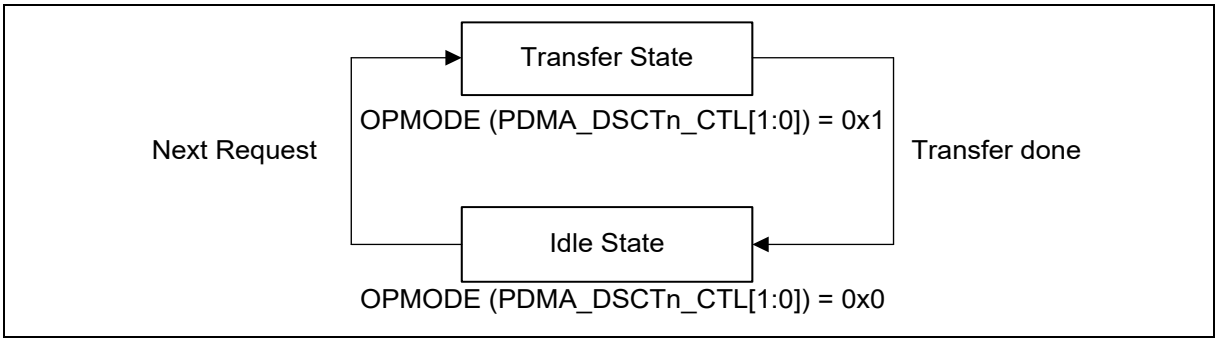


Figure 6.16-3 Basic Mode Finite State Machine

Scatter-gather Mode

Scatter-gather mode is a complex mode and can perform sophisticated transfer through the use of the description link list table as shown in Figure 6.16-4. Through operation mode user can perform peripheral wrapper-around, and multiple PDMA task can be used for data transfer between varied locations in system memory instead of a set of contiguous locations. Scatter-gather mode only needs a request to finish all table entries task till the last task with OPMODE (PDMA_DSCTn_CTL[1:0]) is idle state without ack. It also means Scatter-gather mode can only be used to transfer data between memory to memory without handshaking.

In Scatter-gather mode, the table is just used for jumping to the next table entry. The first task will not perform any operation transfer. Finishing each task will generate an interrupt to CPU if corresponding PDMA interrupt bit is enabled and TBINTDIS (PDMA_DSCTn_CTL[7]) bit is “0” (when finishing task and TBINTDIS bit is “0”, corresponding TDIFn (PDMA_TDSTS[7:0]) flag will be asserted and if this bit is “1”

TDIFn will not be active).

If channel 9 has been triggered, and the operation mode is in Scatter-Gather mode (OPMODE (PDMA_DSCTn_CTL[1:0]) = 0x2), the hardware will load the real PDMA information task from the address generated by PDMA_DSCTn_NEXT (link address). For example, the address in PDMA_DSCTn_NEXT is 0x2000_0100, and then the next DSCT entry start address is 0x2000_0100.

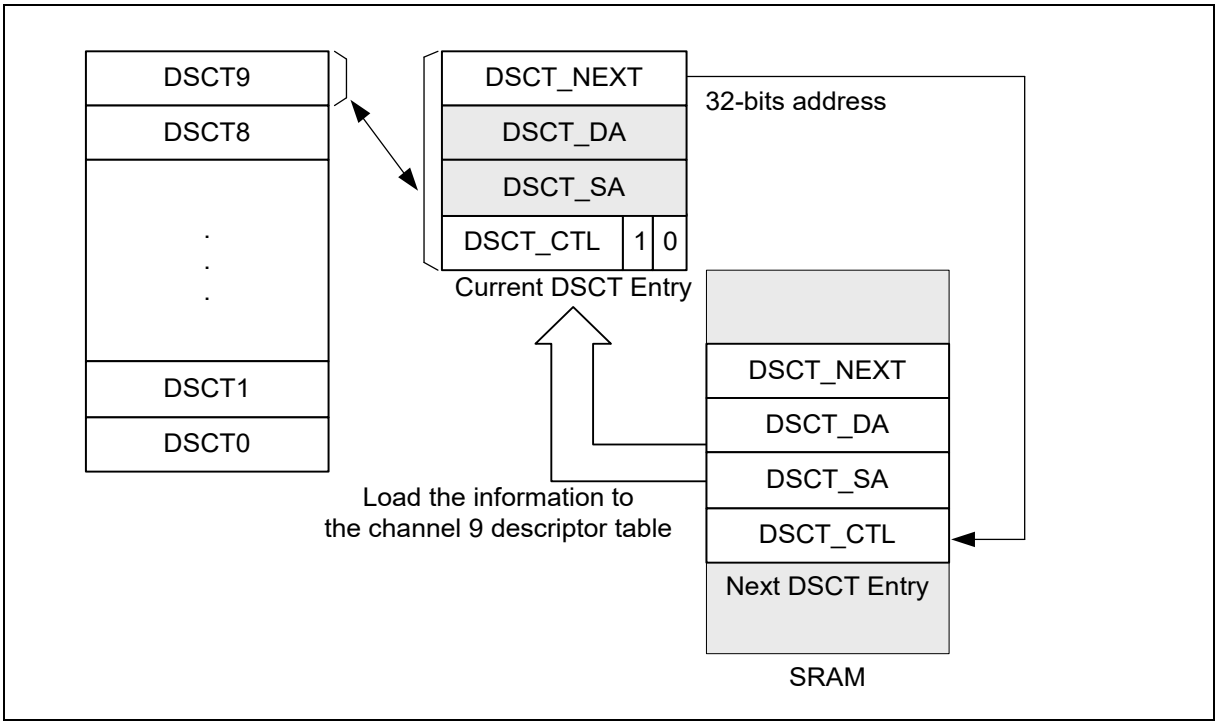


Figure 6.16-4 Descriptor Table Link List Structure

The above link list table operation is DSCT state in Scatter-gather Mode as shown in Figure 6.16-5. When loading the information is finished, it will go to transfer state and start transfer by this information automatically. However, if the next PDMA information is also set to Scatter-gather mode, the hardware will catch the next PDMA information block when the current task is finished. The Scatter-gather mode switches to basic mode when doing the next task. Then, the basic mode switches to Idle state when the last task is finished.

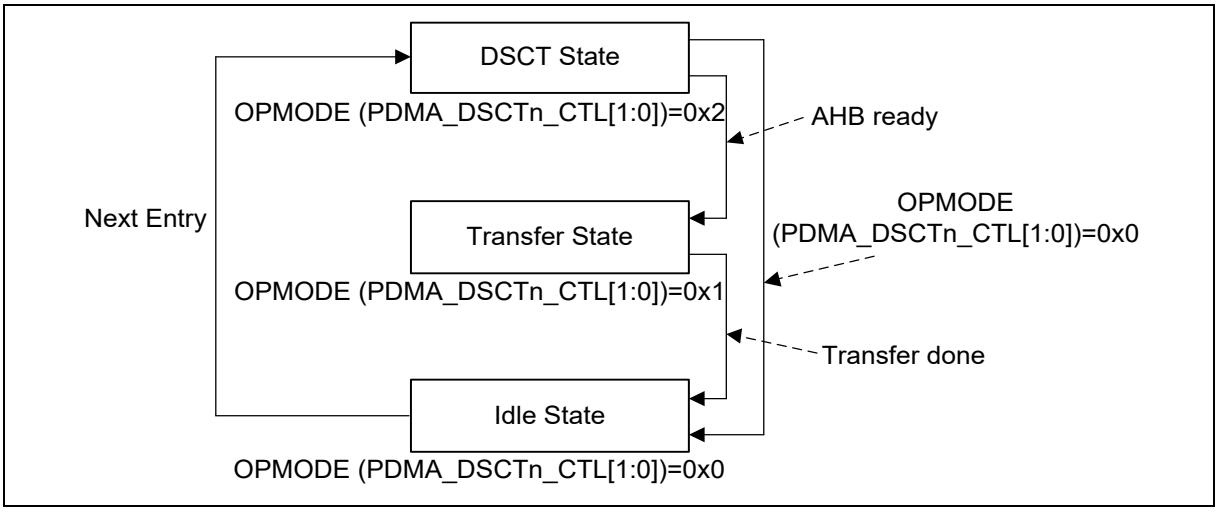


Figure 6.16-5 Scatter-gather Mode Finite State Machine

6.16.5.3 Transfer Type

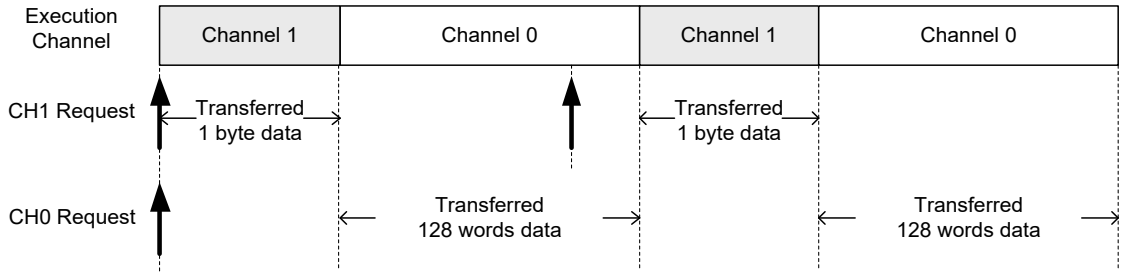
The PDMA controller supports two transfer types: single transfer type and burst transfer type, configure by setting TXTYPE (PDMA_DSCTn_CTL[2]).

When the PDMA controller is operated in single transfer type, each transfer data needs one request signal for one transfer, after transferred data, TXCNT (PDMA_DSCTn_CTL[31:16]) will decrease 1. Transfer will be finished after the TXCNT (PDMA_DSCTn_CTL[31:16]) decreases to 0. In this mode, the BURSIZE (PDMA_DSCTn_CTL[6:4]) is not useful to control the transfer size. The BURSIZE (PDMA_DSCTn_CTL[6:4]) will be fixed as one.

For the burst transfer type, the PDMA controller transfers TXCNT (PDMA_DSCTn_CTL[31:16]) of data and need only one request signal. After transferred BURSIZE (PDMA_DSCTn_CTL[6:4]) of data, TXCNT (PDMA_DSCTn_CTL[31:16]) will decrease BURSIZE number. Transfer will be done after the transfer count TXCNT (PDMA_DSCTn_CTL[31:16]) decreases to 0. Note that burst transfer type can only be used for PDMA controller to do burst transfer between memory and memory. User must use single request type for memory-to-peripheral and peripheral-to-memory transfers. Please note that, PDMA transfer data between Flash and memory should finish before MCU enter idle mode or power done mode to prevent access wrong data.

Figure 6.16-6 shows an example about single and burst transfer type in basic mode. In this example, channel 1 uses single transfer type and TXCNT (PDMA_DSCTn_CTL[31:16]) = 127. Channel 0 uses burst transfer type, BURSIZE (PDMA_DSCTn_CTL[6:4]) = 128 and TXCNT (PDMA_DSCTn_CTL[31:16]) = 255. The operation sequence is described below:

1. Channel 0 and channel 1 get the trigger signal at the same time.
2. Channel 1 has higher priority than channel 0 by default; the PDMA controller will load the channel 1 descriptor table first and executing. But channel 1 is single transfer type, and thus the PDMA controller will only transfer one transfer data.
3. Then, the PDMA controller turns to the channel 0 and loads channel 0's descriptor table. The channel 0 is burst transfer type and the burst size selected to 128. Therefore, the PDMA controller will transfer 128 transfer data.
4. When channel 0 transfers 128 data, channel 1 gets another request signal, then after channel 0 finishes 128 transfer data, the PDMA controller will turn to channel 1 and transfer next one data.
5. After channel 1 transfers data, the PDMA controller switches to low priority channel 0 to continuous next 128 data transfer. If no channel 1 request receives, PDMA will start next channel 0, 128 data transfer.
6. The PDMA controller will complete transfer when channel 0 finishes data transfer 256 times, and channel 1 finishes transferring 128 times.



	CH1	CH0
TXCNT (PDMA_DSCTn_CTL[31:16])	127	255
BURSIZE (PDMA_DSCTn_CTL[6:4])	0 (1 byte)	2 (1 word)
BURSIZE (PDMA_DSCTn_CTL[6:4])	Non-useful	0 (128 transfers)
TXTYPE (PDMA_DSCTn_CTL[2])	1 (Single Transfer)	0 (Burst Transfer)

Figure 6.16-6 Example of Single Transfer Type and Burst Transfer Type in Basic Mode

6.16.5.4 Advanced Peripheral to Memory Transfer

For advanced IPs, UARTs, QSPIs and SPIs, the PDMA provides a more efficient way to transfer data, called advanced mode. PDMA channels enter advanced mode automatically when user sets certain channel to those advanced IPs. In order to make sure PDMA detects advanced mode correctly, user must set enabling PDMA channels firstly, followed by setting PDMA Request Source Select Register. Within this mode, in general case the PDMA will transfer 2-words data to/from peripheral for every request. Besides, this mode does not support stride mode and enhanced stride mode. The source address and destination address must be word-aligned. When all advanced transfer missions were completed, the PDMA Request Source Select Register must be reset to 0.

With those advanced IPs in scatter-gather mode, PDMA will pre-fetch table in the memory according to the internal descriptor that user had defined. In order to load table clearly, user must to make sure that action of filling the internal descriptor will not start until all linked-tables in the memory are done and stable. There are 2 rows data, control and next address, will be used in internal descriptor, user must to make sure that control setting will be the last one to finish.

For example, there are 2 transfer tasks which are await, the flow of software setting is

1. Fill table 1 in the memory
2. Fill table 2 in the memory
3. Enabling PDMA channel
4. Set PDMA request source select register
5. Fill the next address in internal descriptor
6. Fill the control setting in internal descriptor

6.16.5.5 Channel Time-out

The PDMA supports time-out function. When the transfer channel is enabled and selected to the peripheral, corresponding channel time-out TOUTENn (PDMA_TOUTEN [n], n=0~9) is enabled, then

channel's corresponding time-out counter will start counting up from 0 while the channel has received trigger signal from the peripheral. The basic transfer unit for advanced peripheral IP is 2-words. On the other hand, the time-out function will alarm due to the absence of 2-words data from SPI or UART.

The time-out counter is based on output of HCLK prescaler, which is setting by corresponding channel's TOUTPSCn (PDMA_TOUTPSC [2+4n:4n], n=0~9). If time-out counter counts up from 0 to corresponding channel's TOCn (PDMA_TOC0_1 [16(n+1)-1):16n], n=0~9), the PDMA controller will generate interrupt signal when corresponding TOUTIENn (PDMA_TOUTIEN [n], n=0~9) is enabled. When time-out occurred, corresponding channel's REQTOFn (PDMA_INTSTS [n+8], n=0~9) will be set to indicate channel time-out is happened.

Time-out counter will restart from 0 while counter counts to TOCn (PDMA_TOC0_1 [16(n+1)-1:16n], n=0~9), received trigger signal, time-out function is disabled or chip enters Power-down mode. The time-out counter will keep counting until time-out function is disabled.

Figure 6.16-7 shows an example about time-out counter operation. The operation sequence is described below:

1. The channel 0 time-out counter is not counting when time-out function is enabled by setting TOUTEN0(PDMA_TOUTEN[0]) bit to 1.
2. Time-out counter starts counting from 0 to the value of TOC0(PDMA_TOC0_1[15:0]) bits when receiving the first peripheral request.
3. Time-out counter is reset to 0 by received second peripheral request.
4. Channel 0 request time-out flag(REQTOF0(PDMA_INTSTS[8])) is set to high when time-out counter counts to 5. The counter will keep counting.
5. Time-out counter is reset to 0 when time-out function is disabled.

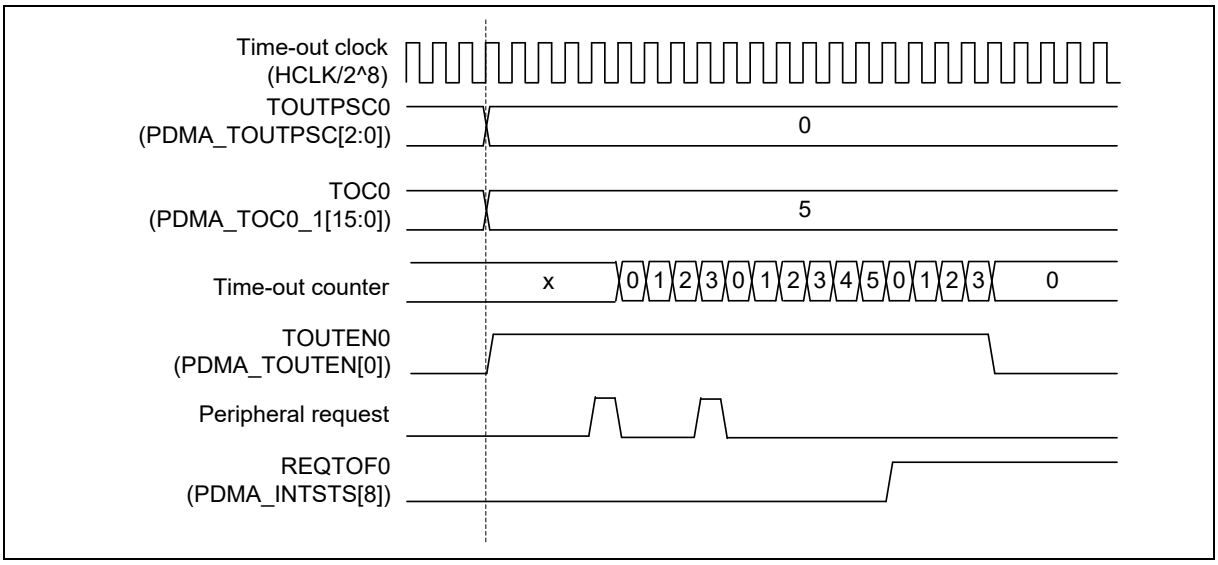


Figure 6.16-7 Example of PDMA Channel 0 Time-out Counter Operation

6.16.5.6 Stride Function

The PDMA supports six channels channel 0 to channel 5 with stride function. The stride function can transfer data from one address to another address and support block transfer with stride. When operating with stride function, the transfer address can be fixed or incremented successively.

Set STRIDEEN (PDMA_DSCTn_CTL[15]) to enable the stride function, and then write a valid source address to the PDMA_DSCTn_SA register and a source address offset count to SASOL (PDMA_ASOCRn[15:0]) register, a destination address to the PDMA_DSCTn_DA register and a destination address offset count to DASOL (PDMA_ASOCRn[31:16]), and a transfer count to the

TXCNT (PDMA_DSCTn_CTL[31:16]) and a stride transfer count to STC (PDMA_STCRn[15:0]). Next, trigger the SWREQn (PDMA_SWREQ[5:0]). The PDMA will start and then stop the transfer after TXCNT (PDMA_DSCTn_CTL[31:16]) counts down to 0.

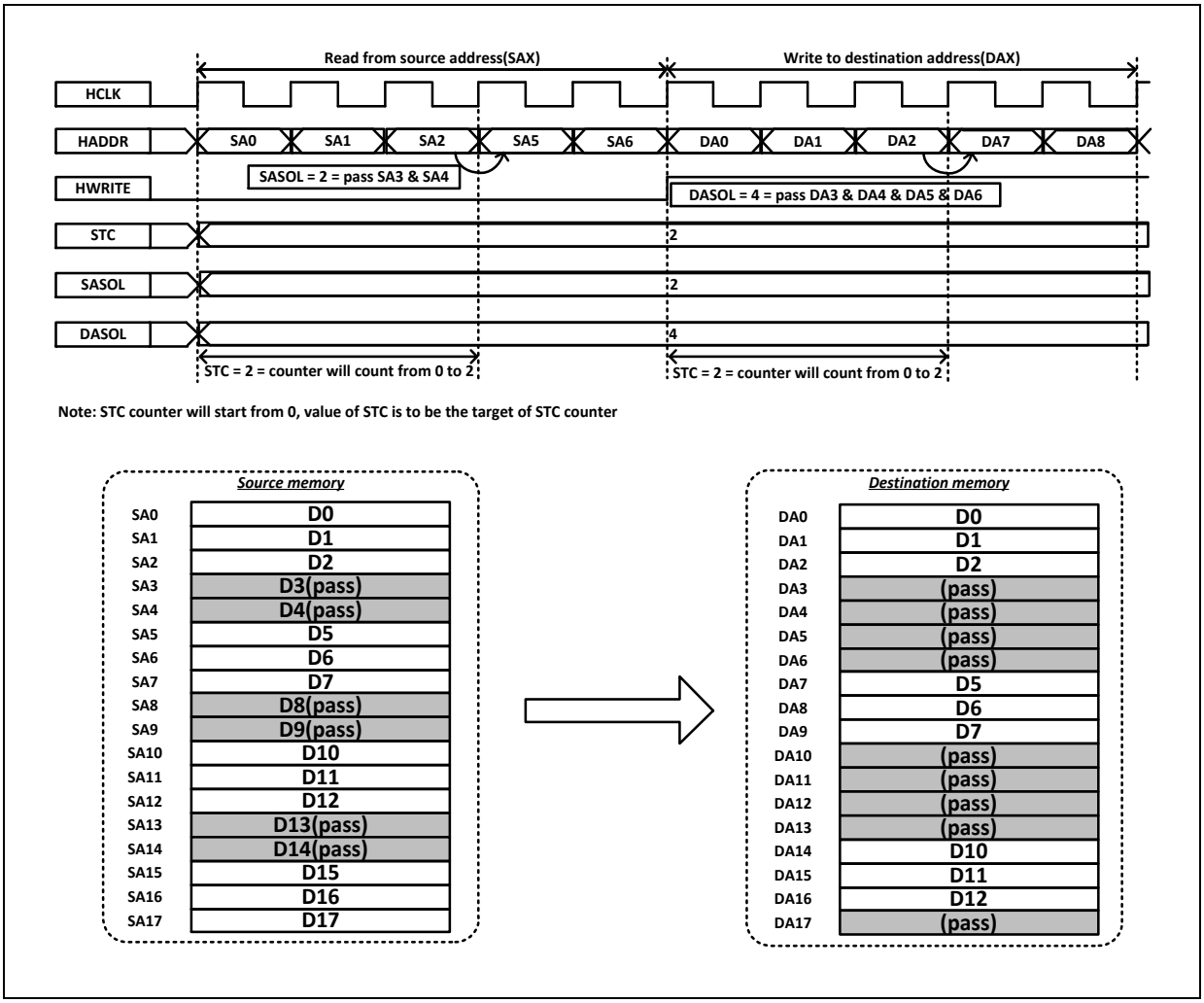


Figure 6.16-8 Stride Function Block Transfer

Figure 6.16-8 shows the block transfer relationship between source memory and destination memory. The stride function only supports memory to memory transfer.

6.16.5.7 Enhanced Stride Function

To improve the stride function for DSP, image processing and other applications, channel 0 and channel 1 support interval count and repeat count. Interval count is used to decide the interval between each block transfer. SAICNT (PDMA_AICTLn[15:0]) is for source address interval count, and DAICNT (PDMA_AICTLn[31:16]) is for destination address interval count. Repeat count is used to decide repeat times of block transfer with only one trigger. If block transfer repeats RCNT (PDMA_RCNTn[15:0]) times, transfer is stopped. Figure 6.16-9 shows two cases of block transfer: Case one with SAICNT = 3, DAICNT = 9 and repeat count = 3. Case two with SAICNT = 3, DAICNT = 6 and repeat count = 3. The enhanced stride function only supports memory to memory transfer.

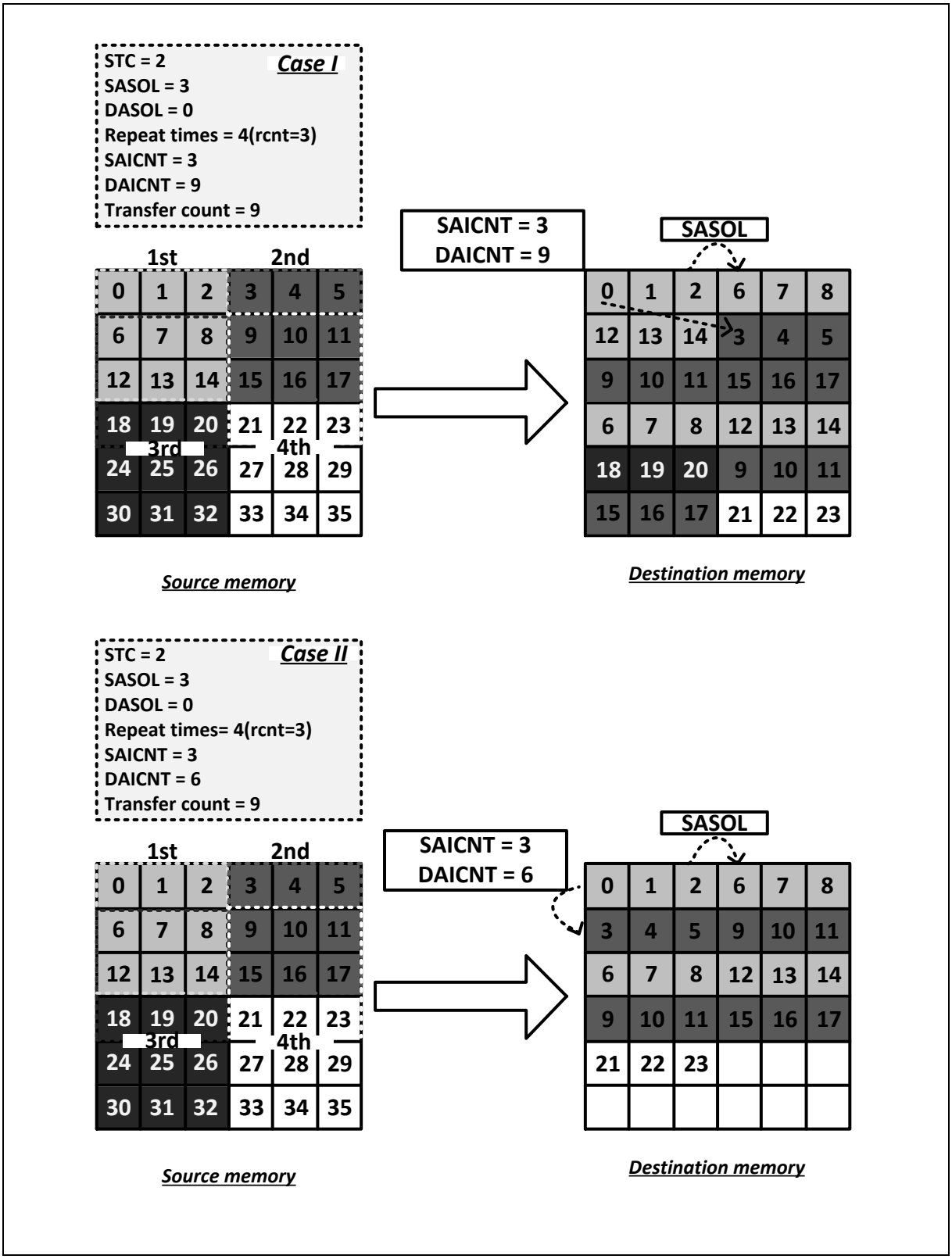


Figure 6.16-9 Block Transfer with Interval =3 and Repeat Count =2

6.16.5.8 Security Policy

There are four PDMA controllers: Both PDMA0 and PDMA1 can be configured as secure or non-secure PDMA. Both PDMA2 and PDMA3 can be configured as non-secure or SubM PDMA. The Secure PDMA is only served as secure master to access secure or non-secure region. If non-secure master requests secure PDMA to access secure region, the PDMA will return fault signal. Non-secure PDMA can be served as both secure and non-secure master, but it can only access non-secure region. For memory access policy, please refer to the Secure Configuration Unit (SCU) section.

6.16.6 Register Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
PDMA Base Address: **PDMAx_BA = 0x4008_0000+(0x1_0000*x)** **x=0,1,2,3**				
PDMAx_DSCTn_CTL **n = 0,1..9**	PDMAx_BA+0x10*n	R/W	Descriptor Table Control Register of PDMA Channel n	0xXXXX_XXXX
PDMAx_DSCTn_SA **n = 0,1..9**	PDMAx_BA+0x0004+0x10*n	R/W	Source Address Register of PDMA Channel n	0xXXXX_XXXX
PDMAx_DSCTn_DA **n = 0,1..9**	PDMAx_BA+0x0008+0x10*n	R/W	Destination Address Register of PDMA Channel n	0xXXXX_XXXX
PDMAx_DSCTn_NEXT **n = 0,1..9**	PDMAx_BA+0x000c+0x10*n	R/W	Next Scatter-gather Descriptor Table Offset Address of PDMA Channel n	0xXXXX_XXXX
PDMAx_CURSCATn **n = 0,1..9**	PDMAx_BA+0x00A0+0x004*n	R	Current Scatter-gather Descriptor Table Address of PDMA Channel n	0xXXXX_XXXX
PDMAx_CHCTL	PDMAx_BA+0x400	R/W	PDMA Channel Control Register	0x0000_0000
PDMAx_PAUSE	PDMAx_BA+0x404	W	PDMA Transfer Pause Control Register	0x0000_0000
PDMAx_SWREQ	PDMAx_BA+0x408	W	PDMA Software Request Register	0x0000_0000
PDMAx_TRGSTS	PDMAx_BA+0x40C	R	PDMA Channel Request Status Register	0x0000_0000
PDMAx_PRISET	PDMAx_BA+0x410	R/W	PDMA Fixed Priority Setting Register	0x0000_0000
PDMAx_PRICLR	PDMAx_BA+0x414	W	PDMA Fixed Priority Clear Register	0x0000_0000
PDMAx_INTEN	PDMAx_BA+0x418	R/W	PDMA Interrupt Enable Register	0x0000_0000
PDMAx_INTSTS	PDMAx_BA+0x41C	R/W	PDMA Interrupt Status Register	0x0000_0000
PDMAx_ABTSTS	PDMAx_BA+0x420	R/W	PDMA Channel Read/Write Target Abort Flag Register	0x0000_0000
PDMAx_TDSTS	PDMAx_BA+0x424	R/W	PDMA Channel Transfer Done Flag Register	0x0000_0000
PDMAx_ALIGN	PDMAx_BA+0x428	R/W	PDMA Transfer Alignment Status Register	0x0000_0000
PDMAx_TACTSTS	PDMAx_BA+0x42C	R	PDMA Transfer Active Flag Register	0x0000_0000
PDMAx_TOUTEN	PDMAx_BA+0x434	R/W	PDMA Time-out Enable Register	0x0000_0000
PDMAx_TOUTIEN	PDMAx_BA+0x438	R/W	PDMA Time-out Interrupt Enable Register	0x0000_0000
PDMAx_TOC0_1	PDMAx_BA+0x440	R/W	PDMA Time-out Counter Ch1 and Ch0 Register	0x0000_0000
PDMAx_TOC2_3	PDMAx_BA+0x444	R/W	PDMA Time-out Counter Ch3 and Ch2 Register	0x0000_0000
PDMAx_TOC4_5	PDMAx_BA+0x448	R/W	PDMA Time-out Counter Ch5 and Ch4 Register	0x0000_0000
PDMAx_TOC6_7	PDMAx_BA+0x44C	R/W	PDMA Time-out Counter Ch7 and Ch6 Register	0x0000_0000
PDMAx_TOC8_9	PDMAx_BA+0x450	R/W	PDMA Time-out Counter Ch9 and Ch8 Register	0x0000_0000
PDMAx_CHRST	PDMAx_BA+0x460	R/W	PDMA Channel Reset Register	0x0000_0000
PDMAx_TOUTPSC	PDMAx_BA+0x470	R/W	PDMA Time-out Prescaler Register	0x0000_0000

PDMAx_TOUTPSC1	PDMAx_BA+0x474	R/W	PDMA Time-out Prescaler Register1	0x0000_0000
PDMAx_REQSEL0_3	PDMAx_BA+0x480	R/W	PDMA Request Source Select Register 0	0x0000_0000
PDMAx_REQSEL4_7	PDMAx_BA+0x484	R/W	PDMA Request Source Select Register 1	0x0000_0000
PDMAx_REQSEL8_11	PDMAx_BA+0x488	R/W	PDMA Request Source Select Register 2	0x0000_0000
PDMAx_STCR0	PDMAx_BA+0x500	R/W	Stride Transfer Count Register of PDMA Channel 0	0x0000_0000
PDMAx_ASOCR0	PDMAx_BA+0x504	R/W	Address Stride Offset Register of PDMA Channel 0	0x0000_0000
PDMAx_STCR1	PDMAx_BA+0x508	R/W	Stride Transfer Count Register of PDMA Channel 1	0x0000_0000
PDMAx_ASOCR1	PDMAx_BA+0x50C	R/W	Address Stride Offset Register of PDMA Channel 1	0x0000_0000
PDMAx_STCR2	PDMAx_BA+0x510	R/W	Stride Transfer Count Register of PDMA Channel 2	0x0000_0000
PDMAx_ASOCR2	PDMAx_BA+0x514	R/W	Address Stride Offset Register of PDMA Channel 2	0x0000_0000
PDMAx_STCR3	PDMAx_BA+0x518	R/W	Stride Transfer Count Register of PDMA Channel 3	0x0000_0000
PDMAx_ASOCR3	PDMAx_BA+0x51C	R/W	Address Stride Offset Register of PDMA Channel 3	0x0000_0000
PDMAx_STCR4	PDMAx_BA+0x520	R/W	Stride Transfer Count Register of PDMA Channel 4	0x0000_0000
PDMAx_ASOCR4	PDMAx_BA+0x524	R/W	Address Stride Offset Register of PDMA Channel 4	0x0000_0000
PDMAx_STCR5	PDMAx_BA+0x528	R/W	Stride Transfer Count Register of PDMA Channel 5	0x0000_0000
PDMAx_ASOCR5	PDMAx_BA+0x52C	R/W	Address Stride Offset Register of PDMA Channel 5	0x0000_0000
PDMAx_AICTL0	PDMAx_BA + 0x600	R/W	Address Interval Control Register of PDMA Channel 0	0x0000_0000
PDMAx_RCNT0	PDMAx_BA + 0x604	R/W	Repeat Count Register of PDMA Channel 0	0x0000_0000
PDMAx_AICTL1	PDMAx_BA + 0x608	R/W	Address Interval Control Register of PDMA Channel 1	0x0000_0000
PDMAx_RCNT1	PDMAx_BA + 0x60C	R/W	Repeat Count Register of PDMA Channel 1	0x0000_0000

6.16.7 Register Description

Descriptor Table Control Register (PDMAx_DSCTn_CTL)

Register	Offset	R/W	Description	Reset Value
PDMAx_DSCTn_CTL	PDMAx_BA+0x10*n	R/W	Descriptor Table Control Register of PDMA Channel n	0xXXXX_XXXX

31	30	29	28	27	26	25	24
TXCNT							
23	22	21	20	19	18	17	16
TXCNT							
15	14	13	12	11	10	9	8
STRIDEEN	Reserved	TXWIDTH		DAINC		SAINC	
7	6	5	4	3	2	1	0
TBINTDIS	BURSIZE			Reserved	TXTYPE	OPMODE	

Bits	Description	
[31:16]	**TXCNT**	**Transfer Count** The TXCNT represents the required number of PDMA transfer, the real transfer count is (TXCNT + 1); the maximum transfer count is 65536, every transfer may be byte, half-word or word that is dependent on TXWIDTH field. **Note:** When PDMA finishes each transfer data, this field will be decreased immediately.
[15]	**STRIDEEN**	**Stride Mode Enable Bit** 0 = Stride transfer mode Disabled. 1 = Stride transfer mode Enabled.
[14]	**Reserved**	Reserved.
[13:12]	**TXWIDTH**	**Transfer Width Selection** This field is used for transfer width. 00 = One byte (8 bit) is transferred for every operation. 01= One half-word (16 bit) is transferred for every operation. 10 = One word (32-bit) is transferred for every operation. 11 = Reserved. **Note:** The PDMA transfer source address (PDMA_DSCTn_SA) and PDMA transfer destination address (PDMA_DSCTn_DA) should be alignment under the TXWIDTH selection
[11:10]	**DAINC**	**Destination Address Increment** This field is used to set the destination address increment size. 11 = No increment (fixed address). Others = Increment and size is depended on TXWIDTH selection. **Note:** The fixed address function does not support in memory to memory transfer type.
[9:8]	**SAINC**	**Source Address Increment** This field is used to set the source address increment size. 11 = No increment (fixed address). Others = Increment and size is depended on TXWIDTH selection.

Bits	Description	
		Note: The fixed address function does not support in memory to memory transfer type.
[7]	**TBINTDIS**	**Table Interrupt Disable Bit** This field can be used to decide whether to enable table interrupt or not. If the TBINTDIS bit is enabled it will not generates TDIFn(PDMA_TDSTS[7:0]) when PDMA controller finishes transfer task. 0 = Table interrupt Enabled. 1 = Table interrupt Disabled. **Note:** This function is only for Scatter-gather mode.
[6:4]	**BURSIZE**	**Burst Size** This field is used for peripheral to determine the burst size or used for determine the re-arbitration size. 000 = 128 Transfers. 001 = 64 Transfers. 010 = 32 Transfers. 011 = 16 Transfers. 100 = 8 Transfers. 101 = 4 Transfers. 110 = 2 Transfers. 111 = 1 Transfers. **Note:** This field is only useful in burst transfer type.
[3]	**Reserved**	Reserved.
[2]	**TXTYPE**	**Transfer Type** 0 = Burst transfer type. 1 = Single transfer type.
[1:0]	**OPMODE**	**PDMA Operation Mode Selection** 00 = Idle state: Channel is stopped or this table is complete, when PDMA finishes channel table task, OPMODE will be cleared to idle state automatically. 01 = Basic mode: The descriptor table only has one task. When this task is finished, the PDMA_INTSTS[1] will be asserted. 10 = Scatter-gather mode: When operating in this mode, user must give the next descriptor table address in PDMA_DSCTn_NEXT register; PDMA controller will ignore this task, then load the next task to execute. 11 = Reserved. **Note:** Before filling new transfer task in the Descriptor Table, user must check the PDMA_INTSTS[1] to make sure the current task is complete.

Start Source Address Register (PDMAx_DSCTn_SA)

Register	Offset	R/W	Description	Reset Value
PDMAx_DSCTn_SA	PDMAx_BA+0x0004+0x10*n	R/W	Source Address Register of PDMA Channel n	0xXXXX_XXXX

31	30	29	28	27	26	25	24
SA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SA							

Bits	Description	
[31:0]	**SA**	**PDMA Transfer Source Address** This field indicates a 32-bit source address of PDMA controller.

Destination Address Register (PDMAx_DSCTn_DA)

Register	Offset	R/W	Description	Reset Value
PDMAx_DSCTn_DA	PDMAx_BA+0x0008+0x10*n	R/W	Destination Address Register of PDMA Channel n	0xXXXX_XXXX

31	30	29	28	27	26	25	24
DA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DA							

Bits	Description	
[31:0]	**DA**	**PDMA Transfer Destination Address** This field indicates a 32-bit destination address of PDMA controller.

Next Scatter-gather Descriptor Table Offset Address (PDMAx_DSCTn_NEXT)

Register	Offset	R/W	Description	Reset Value
PDMAx_DSCTn_NEXT	PDMAx_BA+0x000c+0x10*n	R/W	Next Scatter-gather Descriptor Table Offset Address of PDMA Channel n	0xXXXX_XXXX

31	30	29	28	27	26	25	24
EXENEXT							
23	22	21	20	19	18	17	16
EXENEXT							
15	14	13	12	11	10	9	8
NEXT							
7	6	5	4	3	2	1	0
NEXT							

Bits	Description	
[31:16]	**EXENEXT**	**PDMA Execution Next Descriptor Table Offset** This field indicates the offset of next descriptor table address of current execution descriptor table in system memory. **Note:** Write operation is useless in this field.
[15:0]	**NEXT**	**PDMA Next Descriptor Table Offset** This field indicates the offset of the next descriptor table address in system memory. **Read Operation:** When operating in Scatter-gather mode, the last two bits NEXT[1:0] will become reserved, and indicate the first next address of system memory. **Note 1:** The descriptor table address must be word boundary. **Note 2:** Before filling transfer task in the descriptor table, user must check if the descriptor table is complete.

Current Scatter-gather Descriptor Table Address (PDMAx_CURSCATn)

Register	Offset	R/W	Description	Reset Value
PDMAx_CURSCATn	PDMAx_BA+0x00A0+0x004*n	R	Current Scatter-gather Descriptor Table Address of PDMA Channel n	0xXXXX_XXXX

31	30	29	28	27	26	25	24
CURADDR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CURADDR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CURADDR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CURADDR							

Bits	Description	
[31:0]	**CURADDR**	**PDMA Current Description Address (Read Only)** This field indicates a 32-bit current external description address of PDMA controller. **Note:** This field is read only and used for Scatter-gather mode only to indicate the current external description address.

Channel Control Register (PDMAx_CHCTL)

Register	Offset	R/W	Description	Reset Value
PDMAx_CHCTL	PDMAx_BA+0x400	R/W	PDMA Channel Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						CHEN9	CHEN8
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CHEN7	CHEN6	CHEN5	CHEN4	CHEN3	CHEN2	CHEN1	CHEN0

Bits	Description	
[31:10]	**Reserved**	Reserved.
[n] n=0,1..9	**CHENn**	**PDMA Channel Enable Bits** Set this bit to 1 to enable PDMAn operation. Channel cannot be active if it is not set as enabled. 0 = PDMA channel [n] Disabled. 1 = PDMA channel [n] Enabled. **Note:** Setting the corresponding bit of PDMA_PAUSE or PDMA_CHRST register will also clear this bit.

PDMA Transfer Pause Control Register (PDMAx_PAUSE)

Register	Offset	R/W	Description	Reset Value
PDMAx_PAUSE	PDMAx_BA+0x404	W	PDMA Transfer Pause Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						PAUSE9	PAUSE8
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PAUSE7	PAUSE6	PAUSE5	PAUSE4	PAUSE3	PAUSE2	PAUSE1	PAUSE0

Bits	Description	
[31:10]	**Reserved**	Reserved.
[n] n=0,1..9	**PAUSEn**	**PDMA Channel n Transfer Pause Control (Write Only)** User can set PAUSEn bit field to pause the PDMA transfer. When user sets PAUSEn bit, the PDMA controller will pause the on-going transfer, then clear the channel enable bit CHENn(PDMA_CHCTL [n], n=0,1.. 7) and clear request active flag(PDMA_TRGSTS[n:0], n=0,1.. 7). If the paused channel is re-enabled again, the remaining transfers will be processed. 0 = No effect. 1 = Pause PDMA channel n transfer. **Note:** Only for memory to memory transfer.

PDMA Software Request Register (PDMAx_SWREQ)

Register	Offset	R/W	Description	Reset Value
PDMAx_SWREQ	PDMAx_BA+0x408	W	PDMA Software Request Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						SWREQ9	SWREQ8
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SWREQ7	SWREQ6	SWREQ5	SWREQ4	SWREQ3	SWREQ2	SWREQ1	SWREQ0

Bits	Description	
[31:10]	**Reserved**	Reserved.
[n] n=0,1..9	**SWREQn**	**PDMA Software Request (Write Only)** Set this bit to 1 to generate a software request to PDMA [n]. 0 = No effect. 1 = Generate a software request. **Note 1:** User can read PDMA_TRGSTS register to know which channel is on active. Active flag may be triggered by software request or peripheral request. **Note 2:** If user does not enable corresponding PDMA channel, the software request will be ignored.

PDMA Channel Request Status Register (PDMAx_TRGSTS)

Register	Offset	R/W	Description	Reset Value
PDMAx_TRGSTS	PDMAx_BA+0x40C	R	PDMA Channel Request Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						REQSTS9	REQSTS8
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
REQSTS7	REQSTS6	REQSTS5	REQSTS4	REQSTS3	REQSTS2	REQSTS1	REQSTS0

Bits	Description	
[31:10]	**Reserved**	Reserved.
[n] n=0,1..9	**REQSTSn**	**PDMA Channel Request Status (Read Only)** This flag indicates whether channel[n] have a request or not, no matter request from software or peripheral. When the PDMA controller finishes channel transfer, this bit will be cleared automatically. 0 = PDMA Channel n has no request. 1 = PDMA Channel n has a request. **Note:** If user pauses or resets each PDMA transfer by setting PDMA_PAUSE or PDMA_CHRST register respectively, this bit will be cleared automatically after finishing the current transfer.

PDMA Fixed Priority Setting Register (PDMAx_PRISET)

Register	Offset	R/W	Description	Reset Value
PDMAx_PRISET	PDMAx_BA+0x410	R/W	PDMA Fixed Priority Setting Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						FPRISET9	FPRISET8
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FPRISET7	FPRISET6	FPRISET5	FPRISET4	FPRISET3	FPRISET2	FPRISET1	FPRISET0

Bits	Description	
[31:10]	**Reserved**	Reserved.
[n] n=0,1..9	**FPRISETn**	**PDMA Fixed Priority Setting** Set this bit to 1 to enable fixed priority level. **Write Operation:** 0 = No effect. 1 = Set PDMA channel [n] to fixed priority channel. **Read Operation:** 0 = Corresponding PDMA channel is round-robin priority. 1 = Corresponding PDMA channel is fixed priority. **Note:** This field is only set to fixed priority. To clear fixed priority, use PDMA_PRICLR register.

PDMA Fix Priority Clear Register (PDMAx_PRICLR)

Register	Offset	R/W	Description	Reset Value
PDMAx_PRICLR	PDMAx_BA+0x414	W	PDMA Fixed Priority Clear Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						FPRICLR9	FPRICLR8
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FPRICLR7	FPRICLR6	FPRICLR5	FPRICLR4	FPRICLR3	FPRICLR2	FPRICLR1	FPRICLR0

Bits	Description	
[31:10]	**Reserved**	Reserved.
[n] n=0,1..9	**FPRICLRn**	**PDMA Fixed Priority Clear Bits (Write Only)** Set this bit to 1 to clear fixed priority level. 0 = No effect. 1 = Clear PDMA channel [n] fixed priority setting. **Note:** User can read PDMA_PRISET register to know the channel priority.

PDMA Interrupt Enable Register (PDMAx_INTEN)

Register	Offset	R/W	Description	Reset Value
PDMAx_INTEN	PDMAx_BA+0x418	R/W	PDMA Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						INTEN9	INTEN8
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INTEN7	INTEN6	INTEN5	INTEN4	INTEN3	INTEN2	INTEN1	INTEN0

Bits	Description	
[31:10]	**Reserved**	Reserved.
[n] n=0,1..9	**INTENn**	**PDMA Interrupt Enable Bits** This field is used to enable PDMA channel[n] interrupt. 0 = PDMA channel n interrupt Disabled. 1 = PDMA channel n interrupt Enabled. **Note:** The interrupt flag is time-out, abort, transfer done and align.

PDMA Interrupt Status Register (PDMAx_INTSTS)

Register	Offset	R/W	Description	Reset Value
PDMAx_INTSTS	PDMAx_BA+0x41C	R/W	PDMA Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						REQTOF9	REQTOF8
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
REQTOF7	REQTOF6	REQTOF5	REQTOF4	REQTOF3	REQTOF2	REQTOF1	REQTOF0
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					ALIGNF	TDIF	ABTIF

Bits	Description	
[31:18]	**Reserved**	Reserved.
[n+8] n=0,1..9	**REQTOFn**	**Request Time-out Flag for Channel n** This flag indicates that PDMA controller has waited peripheral request for a period defined by TOCn(PDMA_TOCn_n+1[n15:0]. 0 = No request time-out. 1 = Peripheral request time-out. **Note 1:** Please disable time-out function before clearing this bit. **Note 2:** User can write 1 to clear this bit.
[7:3]	**Reserved**	Reserved.
[2]	**ALIGNF**	**Transfer Alignment Interrupt Flag (Read Only)** 0 = PDMA channel source address and destination address both follow transfer width setting. 1 = PDMA channel source address or destination address is not follow transfer width setting.
[1]	**TDIF**	**Transfer Done Interrupt Flag (Read Only)** This bit indicates that PDMA controller has finished transmission; User can read PDMA_TDSTS register to indicate which channel finished transfer. 0 = Not finished yet. 1 = PDMA channel has finished transmission.
[0]	**ABTIF**	**PDMA Read/Write Target Abort Interrupt Flag (Read Only)** This bit indicates that PDMA has target abort error; Software can read PDMA_ABTSTS register to find which channel has target abort error. 0 = No AHB bus ERROR response received. 1 = AHB bus ERROR response received.

PDMA Channel Read/Write Target Abort Flag Register (PDMAx_ABTSTS)

Register	Offset	R/W	Description	Reset Value
PDMAx_ABTSTS	PDMAx_BA+0x420	R/W	PDMA Channel Read/Write Target Abort Flag Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						ABTIF9	ABTIF8
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ABTIF7	ABTIF6	ABTIF5	ABTIF4	ABTIF3	ABTIF2	ABTIF1	ABTIF0

Bits	Description	
[31:10]	**Reserved**	Reserved.
[n] n=0,1..9	**ABTIFn**	**PDMA Read/Write Target Abort Interrupt Status Flag** This bit indicates which PDMA controller has target abort error. 0 = No AHB bus ERROR response received when channel n transfer. 1 = AHB bus ERROR response received when channel n transfer. **Note 1:** If channel n target abort, REQSRCn should be set 0 to disable peripheral request. **Note 2:** User can write 1 to clear this bit.

PDMA Channel Transfer Done Flag Register (PDMAx_TDSTS)

Register	Offset	R/W	Description	Reset Value
PDMAx_TDSTS	PDMAx_BA+0x424	R/W	PDMA Channel Transfer Done Flag Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						TDIF9	TDIF8
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TDIF7	TDIF6	TDIF5	TDIF4	TDIF3	TDIF2	TDIF1	TDIF0

Bits	Description	
[31:10]	**Reserved**	Reserved.
[n] n=0,1..9	**TDIFn**	**Transfer Done Flag** This bit indicates whether PDMA controller channel transfer has been finished or not. 0 = PDMA channel transfer has not finished. 1 = PDMA channel has finished transmission. **Note:** User can write 1 to clear these bits.

PDMA Transfer Alignment Status Register (PDMAx_ALIGN)

Register	Offset	R/W	Description	Reset Value
PDMAx_ALIGN	PDMAx_BA+0x428	R/W	PDMA Transfer Alignment Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						ALIGN9	ALIGN8
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ALIGN7	ALIGN6	ALIGN5	ALIGN4	ALIGN3	ALIGN2	ALIGN1	ALIGN0

Bits	Description	
[31:10]	**Reserved**	Reserved.
[n] n=0,1..9	**ALIGNn**	**Transfer Alignment Flag** This bit indicates whether source and destination address both follow transfer width setting. 0 = PDMA channel source address and destination address both follow transfer width setting. 1 = PDMA channel source address or destination address is not follow transfer width setting. **Note:** User can write 1 to clear these bits.

PDMA Transfer Active Flag Register (PDMAx_TACTSTS)

Register	Offset	R/W	Description	Reset Value
PDMAx_TACTSTS	PDMAx_BA+0x42C	R	PDMA Transfer Active Flag Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						TXACTF9	TXACTF8
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TXACTF7	TXACTF6	TXACTF5	TXACTF4	TXACTF3	TXACTF2	TXACTF1	TXACTF0

Bits	Description	
[31:10]	**Reserved**	Reserved.
[n] n=0,1..9	**TXACTFn**	**Transfer on Active Flag (Read Only)** This bit indicates which PDMA channel is in active. 0 = PDMA channel is finished. 1 = PDMA channel is active.

PDMA Time-out Enable Register (PDMAx_TOUTEN)

Register	Offset	R/W	Description	Reset Value
PDMAx_TOUTEN	PDMAx_BA+0x434	R/W	PDMA Time-out Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						TOUTEN9	TOUTEN8
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TOUTEN7	TOUTEN6	TOUTEN5	TOUTEN4	TOUTEN3	TOUTEN2	TOUTEN1	TOUTEN0

Bits	Description	
[31:10]	**Reserved**	Reserved.
[n] n=0,1..9	**TOUTENn**	**PDMA Time-out Enable Bits** 0 = PDMA Channel n time-out function Disabled. 1 = PDMA Channel n time-out function Enabled.

PDMA Time-out Interrupt Enable Register (PDMAx_TOUTIEN)

Register	Offset	R/W	Description	Reset Value
PDMAx_TOUTIEN	PDMAx_BA+0x438	R/W	PDMA Time-out Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						TOUTIEN9	TOUTIEN8
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TOUTIEN7	TOUTIEN6	TOUTIEN5	TOUTIEN4	TOUTIEN3	TOUTIEN2	TOUTIEN1	TOUTIEN0

Bits	Description	
[31:10]	**Reserved**	Reserved.
[n] n=0,1..9	**TOUTIENn**	**PDMA Time-out Interrupt Enable Bits** 0 = PDMA Channel n time-out interrupt Disabled. 1 = PDMA Channel n time-out interrupt Enabled.

PDMA Time-out Period Counter Register 0 (PDMAx_TOC0_1)

Register	Offset	R/W	Description	Reset Value
PDMAx_TOC0_1	PDMAx_BA+0x440	R/W	PDMA Time-out Counter Ch1 and Ch0 Register	0x0000_0000

31	30	29	28	27	26	25	24
TOC1							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TOC1							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TOC0							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TOC0							

Bits	Description	
[31:16]	**TOC1**	**Time-out Counter for Channel 1** This controls the period of time-out function for channel 1. The calculation unit is based on TOUTPSC1 (PDMA_TOUTPSC[6:4]) clock. The example of time-out period can refer TOC0 bit description.
[15:0]	**TOC0**	**Time-out Counter for Channel 0** This controls the period of time-out function for channel 0. The calculation unit is based on TOUTPSC0 (PDMA_TOUTPSC[2:0]) clock. Time-out period = (Period of time-out clock) * (16-bit TOCn), n = 0,1..9.

PDMA Time-out Period Counter Register 0 (PDMAx_TOC2_3)

Register	Offset	R/W	Description	Reset Value
PDMAx_TOC2_3	PDMAx_BA+0x444	R/W	PDMA Time-out Counter Ch3 and Ch2 Register	0x0000_0000

31	30	29	28	27	26	25	24
TOC3							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TOC3							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TOC2							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TOC2							

Bits	Description	
[31:16]	**TOC3**	**Time-out Counter for Channel 3** This controls the period of time-out function for channel 3. The calculation unit is based on TOUTPSC3 (PDMA_TOUTPSC[14:12]) clock. The example of time-out period can refer TOC2 bit description.
[15:0]	**TOC2**	**Time-out Counter for Channel 2** This controls the period of time-out function for channel 2. The calculation unit is based on TOUTPSC2 (PDMA_TOUTPSC[10:8]) clock. Time-out period = (Period of time-out clock) * (16-bit TOCn), n = 0,1..9.

PDMA Time-out Period Counter Register 0 (PDMAx_TOC4_5)

Register	Offset	R/W	Description	Reset Value
PDMAx_TOC4_5	PDMAx_BA+0x448	R/W	PDMA Time-out Counter Ch5 and Ch4 Register	0x0000_0000

31	30	29	28	27	26	25	24
TOC5							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TOC5							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TOC4							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TOC4							

Bits	Description	
[31:16]	**TOC5**	**Time-out Counter for Channel 5** This controls the period of time-out function for channel 5. The calculation unit is based on TOUTPSC5 (PDMA_TOUTPSC[22:20]) clock. The example of time-out period can refer TOC4 bit description.
[15:0]	**TOC4**	**Time-out Counter for Channel 4** This controls the period of time-out function for channel 4. The calculation unit is based on TOUTPSC4 (PDMA_TOUTPSC[18:16]) clock. Time-out period = (Period of time-out clock) * (16-bit TOCn), n = 0,1..9.

PDMA Time-out Period Counter Register 0 (PDMA6_TOC6_7)

Register	Offset	R/W	Description	Reset Value
PDMAx_TOC6_7	PDMAx_BA+0x44C	R/W	PDMA Time-out Counter Ch7 and Ch6 Register	0x0000_0000

31	30	29	28	27	26	25	24
TOC7							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TOC7							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TOC6							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TOC6							

Bits	Description	
[31:16]	**TOC7**	**Time-out Counter for Channel 7** This controls the period of time-out function for channel 7. The calculation unit is based on TOUTPSC7 (PDMA_TOUTPSC[30:28]) clock. The example of time-out period can refer TOC6 bit description.
[15:0]	**TOC6**	**Time-out Counter for Channel 6** This controls the period of time-out function for channel 6. The calculation unit is based on TOUTPSC6 (PDMA_TOUTPSC[26:24]) clock. Time-out period = (Period of time-out clock) * (16-bit TOCn), n = 0,1..9.

PDMA Time-out Period Counter Register 0 (PDMAx_TOC8_9)

Register	Offset	R/W	Description	Reset Value
PDMAx_TOC8_9	PDMAx_BA+0x450	R/W	PDMA Time-out Counter Ch9 and Ch8 Register	0x0000_0000

31	30	29	28	27	26	25	24
TOC9							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TOC9							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TOC8							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TOC8							

Bits	Description	
[31:16]	**TOC9**	**Time-out Counter for Channel 9** This controls the period of time-out function for channel 9. The calculation unit is based on TOUTPSC9 (PDMA_TOUTPSC1[6:4]) clock. The example of time-out period can refer TOC8 bit description.
[15:0]	**TOC8**	**Time-out Counter for Channel 8** This controls the period of time-out function for channel 8. The calculation unit is based on TOUTPSC8 (PDMA_TOUTPSC1[2:0]) clock. Time-out period = (Period of time-out clock) * (16-bit TOCn), n = 0,1..9.

PDMA Channel Reset Register (PDMAx_CHRST)

Register	Offset	R/W	Description	Reset Value
PDMAx_CHRST	PDMAx_BA+0x460	R/W	PDMA Channel Reset Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						CH9RST	CH8RST
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CH7RST	CH6RST	CH5RST	CH4RST	CH3RST	CH2RST	CH1RST	CH0RST

Bits	Description	
[31:10]	**Reserved**	Reserved.
[n] n=0,1..9	**CHnRST**	**Channel n Reset** 0 = Corresponding channel n is not reset. 1 = Corresponding channel n is reset.

PDMA Time-out Prescaler Register (PDMAx_TOUTPSC)

Register	Offset	R/W	Description	Reset Value
PDMAx_TOUTPSC	PDMAx_BA+0x470	R/W	PDMA Time-out Prescaler Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	TOUTPSC7			Reserved	TOUTPSC6		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved	TOUTPSC5			Reserved	TOUTPSC4		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	TOUTPSC3			Reserved	TOUTPSC2		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	TOUTPSC1			Reserved	TOUTPSC0		

Bits	Description	
[3+n*4] n=0,1..7	**Reserved**	Reserved.
[2+n*4:0+n*4] n=0,1..7	**TOUTPSCn**	**PDMA Channel n Time-out Clock Source Prescaler Bits** 000 = PDMA channel n time-out clock source is HCLK/2^8. 001 = PDMA channel n time-out clock source is HCLK/2^9. 010 = PDMA channel n time-out clock source is HCLK/2^{10}. 011 = PDMA channel n time-out clock source is HCLK/2^{11}. 100 = PDMA channel n time-out clock source is HCLK/2^{12}. 101 = PDMA channel n time-out clock source is HCLK/2^{13}. 110 = PDMA channel n time-out clock source is HCLK/2^{14}. 111 = PDMA channel n time-out clock source is HCLK/2^{15}.

PDMA Time-out Prescaler Register1 (PDMAx_TOUTPSC1)

Register	Offset	R/W	Description	Reset Value
PDMAx_TOUTPSC1	PDMAx_BA+0x474	R/W	PDMA Time-out Prescaler Register1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	TOUTPSC9			Reserved	TOUTPSC8		

Bits	Description	
[31:7]	**Reserved**	Reserved.
[3]	**Reserved**	Reserved.
[2+4*(n-8):0+4*(n-8)] n=8,9	**TOUTPSCn**	**PDMA Channel n Time-out Clock Source Prescaler Bits** 000 = PDMA channel n time-out clock source is HCLK/2^8. 001 = PDMA channel n time-out clock source is HCLK/2^9. 010 = PDMA channel n time-out clock source is HCLK/2^{10}. 011 = PDMA channel n time-out clock source is HCLK/2^{11}. 100 = PDMA channel n time-out clock source is HCLK/2^{12}. 101 = PDMA channel n time-out clock source is HCLK/2^{13}. 110 = PDMA channel n time-out clock source is HCLK/2^{14}. 111 = PDMA channel n time-out clock source is HCLK/2^{15}.

PDMA Request Source Select Register 0 (PDMAx_REQSEL0_3)

Register	Offset	R/W	Description	Reset Value
PDMAx_REQSEL0_3	PDMAx_BA+0x480	R/W	PDMA Request Source Select Register 0	0x0000_0000

31	30	29	28	27	26	25	24
REQSRC3							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
REQSRC2							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
REQSRC1							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
REQSRC0							

Bits	Description	
[31:24]	**REQSRC3**	**Channel 3 Request Source Selection** This filed defines which peripheral is connected to PDMA channel 3. User can configure the peripheral setting by REQSRC3. **Note:** The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.
[23:16]	**REQSRC2**	**Channel 2 Request Source Selection** This filed defines which peripheral is connected to PDMA channel 2. User can configure the peripheral setting by REQSRC2. **Note:** The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.
[15:8]	**REQSRC1**	**Channel 1 Request Source Selection** This filed defines which peripheral is connected to PDMA channel 1. User can configure the peripheral setting by REQSRC1. **Note:** The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.
[7:0]	**REQSRC0**	**Channel 0 Request Source Selection** This filed defines which peripheral is connected to PDMA channel 0. User can configure the peripheral by setting REQSRC0. 00 = Disable PDMA peripheral request. 01 = Reserved. 02 = Reserved. 03 = Reserved. 04 = Channel connects to UART0_TX. 05 = Channel connects to UART0_RX. 06 = Channel connects to UART1_TX. 07 = Channel connects to UART1_RX. 08 = Channel connects to UART2_TX. 09 = Channel connects to UART2_RX. 10 = Channel connects to UART3_TX. 11 = Channel connects to UART3_RX.

Bits	Description	
		12 = Channel connects to UART4_TX. 13 = Channel connects to UART4_RX. 14 = Channel connects to UART5_TX. 15 = Channel connects to UART5_RX. 16 = Channel connects to UART6_TX. 17 = Channel connects to UART6_RX. 18 = Channel connects to UART7_TX. 19 = Channel connects to UART7_RX. 20 = Channel connects to UART8_TX. 21 = Channel connects to UART8_RX. 22 = Channel connects to UART9_TX. 23 = Channel connects to UART9_RX. 24 = Channel connects to UART10_TX. 25 = Channel connects to UART10_RX. 26 = Channel connects to UART11_TX. 27 = Channel connects to UART11_RX. 28 = Channel connects to UART12_TX. 29 = Channel connects to UART12_RX. 30 = Channel connects to UART13_TX. 31 = Channel connects to UART13_RX. 32 = Channel connects to UART14_TX. 33 = Channel connects to UART14_RX. 34 = Channel connects to UART15_TX. 35 = Channel connects to UART15_RX. 36 = Channel connects to UART16_TX. 37 = Channel connects to UART16_RX. 54 = Channel connects to QSPI0_TX. 55 = Channel connects to QSPI0_RX. 56 = Channel connects to QSPI1_TX. 57 = Channel connects to QSPI1_RX. 60 = Channel connects to SPI0_TX. 61 = Channel connects to SPI0_RX. 62 = Channel connects to SPI1_TX. 63 = Channel connects to SPI1_RX. 64 = Channel connects to SPI2_TX. 65 = Channel connects to SPI2_RX. 66 = Channel connects to SPI3_TX. 67 = Channel connects to SPI3_RX. 70 = Channel connects to I2C0_TX. 71 = Channel connects to I2C0_RX. 72 = Channel connects to I2C1_TX. 73 = Channel connects to I2C1_RX. 74 = Channel connects to I2C2_TX. 75 = Channel connects to I2C2_RX. 76 = Channel connects to I2C3_TX. 77 = Channel connects to I2C3_RX. 78 = Channel connects to I2C4_TX. 79 = Channel connects to I2C4_RX.

Bits	Description	
		80 = Channel connects to I2C5_TX. 81 = Channel connects to I2C5_RX. 86 = Channel connects to I2S0_TX. 87 = Channel connects to I2S0_RX. 88= Channel connects to I2S1_TX. 89= Channel connects to I2S1_RX. 94 = Channel connects to TMR0. 95 = Channel connects to TMR1. 96 = Channel connects to TMR2. 97 = Channel connects to TMR3. 98 = Channel connects to TMR4. 99 = Channel connects to TMR5. 100 = Channel connects to TMR6. 101 = Channel connects to TMR7. 102 = Channel connects to TMR8. 103 = Channel connects to TMR9. 104 = Channel connects to TMR10. 105 = Channel connects to TMR11. 110 = Channel connects to EPWM0_P1_RX. 111 = Channel connects to EPWM0_P2_RX. 112 = Channel connects to EPWM0_P3_RX. 113 = Channel connects to EPWM1_P1_RX. 114 = Channel connects to EPWM1_P2_RX. 115 = Channel connects to EPWM1_P3_RX. 116 = Channel connects to EPWM2_P1_RX. 117 = Channel connects to EPWM2_P2_RX. 118 = Channel connects to EPWM2_P3_RX. 122 = Channel connects to EPWM0_CH0_TX. 123 = Channel connects to EPWM0_CH1_TX. 124 = Channel connects to EPWM0_CH2_TX. 125 = Channel connects to EPWM0_CH3_TX. 126 = Channel connects to EPWM0_CH4_TX. 127 = Channel connects to EPWM0_CH5_TX. 128 = Channel connects to EPWM1_CH0_TX. 129 = Channel connects to EPWM1_CH1_TX. 130 = Channel connects to EPWM1_CH2_TX. 131 = Channel connects to EPWM1_CH3_TX. 132 = Channel connects to EPWM1_CH4_TX. 133 = Channel connects to EPWM1_CH5_TX. 134 = Channel connects to EPWM2_CH0_TX. 135 = Channel connects to EPWM2_CH1_TX. 136 = Channel connects to EPWM2_CH2_TX. 137 = Channel connects to EPWM2_CH3_TX. 138 = Channel connects to EPWM2_CH4_TX. 139 = Channel connects to EPWM2_CH5_TX. 146 = Channel connects to ADC_RX. Others = Reserved.

Bits	Description	
		Note 1: A peripheral cannot be assigned to two channels at the same time. **Note 2:** This field is useless when transfer between memory and memory.

PDMA Request Source Select Register 1 (PDMAx_REQSEL4_7)

Register	Offset	R/W	Description	Reset Value
PDMAx_REQSEL4_7	PDMAx_BA+0x484	R/W	PDMA Request Source Select Register 1	0x0000_0000

31	30	29	28	27	26	25	24
REQSRC7							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
REQSRC6							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
REQSRC5							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
REQSRC4							

Bits	Description	
[31:24]	**REQSRC7**	**Channel 7 Request Source Selection** This filed defines which peripheral is connected to PDMA channel 7. User can configure the peripheral setting by REQSRC7. **Note:** The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.
[23:16]	**REQSRC6**	**Channel 6 Request Source Selection** This filed defines which peripheral is connected to PDMA channel 6. User can configure the peripheral setting by REQSRC6. **Note:** The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.
[15:8]	**REQSRC5**	**Channel 5 Request Source Selection** This filed defines which peripheral is connected to PDMA channel 5. User can configure the peripheral setting by REQSRC5. **Note:** The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.
[7:0]	**REQSRC4**	**Channel 4 Request Source Selection** This filed defines which peripheral is connected to PDMA channel 4. User can configure the peripheral setting by REQSRC4. **Note:** The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.

PDMA Request Source Select Register 2 (PDMAx_REQSEL8_11)

Register	Offset	R/W	Description	Reset Value
PDMAx_REQSEL8_11	PDMAx_BA+0x488	R/W	PDMA Request Source Select Register 2	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
REQSRC9							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
REQSRC8							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:8]	**REQSRC9**	**Channel 9 Request Source Selection** This filed defines which peripheral is connected to PDMA channel 9. User can configure the peripheral setting by REQSRC9. **Note:** The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.
[7:0]	**REQSRC8**	**Channel 8 Request Source Selection** This filed defines which peripheral is connected to PDMA channel 8. User can configure the peripheral setting by REQSRC8. **Note:** The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.

PDMA Stride Transfer Count Register n (PDMAx_STCRn)

Register	Offset	R/W	Description	Reset Value
PDMAx_STCR0	PDMAx_BA+0x500	R/W	Stride Transfer Count Register of PDMA Channel 0	0x0000_0000
PDMAx_STCR1	PDMAx_BA+0x508	R/W	Stride Transfer Count Register of PDMA Channel 1	0x0000_0000
PDMAx_STCR2	PDMAx_BA+0x510	R/W	Stride Transfer Count Register of PDMA Channel 2	0x0000_0000
PDMAx_STCR3	PDMAx_BA+0x518	R/W	Stride Transfer Count Register of PDMA Channel 3	0x0000_0000
PDMAx_STCR4	PDMAx_BA+0x520	R/W	Stride Transfer Count Register of PDMA Channel 4	0x0000_0000
PDMAx_STCR5	PDMAx_BA+0x528	R/W	Stride Transfer Count Register of PDMA Channel 5	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
11	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
STC							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
STC							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**STC**	**PDMA Stride Transfer Count** The 16-bit register defines the stride transfer count of each row.

PDMA Address Stride Offset Control Register n (PDMAx_ASOCRn)

Register	Offset	R/W	Description	Reset Value
PDMAx_ASOCR0	PDMAx_BA+0x504	R/W	Address Stride Offset Register of PDMA Channel 0	0x0000_0000
PDMAx_ASOCR1	PDMAx_BA+0x50C	R/W	Address Stride Offset Register of PDMA Channel 1	0x0000_0000
PDMAx_ASOCR2	PDMAx_BA+0x514	R/W	Address Stride Offset Register of PDMA Channel 2	0x0000_0000
PDMAx_ASOCR3	PDMAx_BA+0x51C	R/W	Address Stride Offset Register of PDMA Channel 3	0x0000_0000
PDMAx_ASOCR4	PDMAx_BA+0x524	R/W	Address Stride Offset Register of PDMA Channel 4	0x0000_0000
PDMAx_ASOCR5	PDMAx_BA+0x52C	R/W	Address Stride Offset Register of PDMA Channel 5	0x0000_0000

31	30	29	28	27	26	25	24
DASOL							
11	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DASOL							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SASOL							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SASOL							

Bits	Description	
[31:16]	**DASOL**	**VDMA Destination Address Stride Offset Length** The 16-bit register defines the destination address stride transfer offset count of each row.
[15:0]	**SASOL**	**VDMA Source Address Stride Offset Length** The 16-bit register defines the source address stride transfer offset count of each row.

PDMA Address Interval Control Register n (PDMAx_AICTLn)

Register	Offset	R/W	Description	Reset Value
PDMAx_AICTL0	PDMAx_BA + 0x600	R/W	Address Interval Control Register of PDMA Channel 0	0x0000_0000
PDMAx_AICTL1	PDMAx_BA + 0x608	R/W	Address Interval Control Register of PDMA Channel 1	0x0000_0000

31	30	29	28	27	26	25	24
DAICNT							
11	22	21	20	19	18	17	16
DAICNT							
15	14	13	12	11	10	9	8
SAICNT							
7	6	5	4	3	2	1	0
SAICNT							

Bits	Description	
[31:16]	**DAICNT**	**PDMA Destination Address Interval Count** The 16-bit register defines the destination address interval count of each row.
[15:0]	**SAICNT**	**PDMA Source Address Interval Count** The 16-bit register defines the source address interval count of each row.

PDMA Repeat Count Register n (PDMAx_RCNTn)

Register	Offset	R/W	Description	Reset Value
PDMAx_RCNT0	PDMAx_BA + 0x604	R/W	Repeat Count Register of PDMA Channel 0	0x0000_0000
PDMAx_RCNT1	PDMAx_BA + 0x60C	R/W	Repeat Count Register of PDMA Channel 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
11	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RCNT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RCNT							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**RCNT**	**PDMA Repeat Count** The 16-bit register defines the repeat times of block transfer.

6.17 Timer Controller (TMR)

6.17.1 Overview

The timer controller includes 12 sets of 32-bit timers, Timer0 ~ Timer11, allowing user to easily implement a timer control for applications. The timer can perform functions, such as frequency measurement, delay timing, clock generation, and event counting by external input pins, and interval measurement by external capture pins.

The timer controller also provides twelve PWM generators. Each PWM generator supports two PWM output channels in independent mode and complementary mode. The output state of PWM output pin can be control by pin mask, polarity and break control, and dead-time generator.

6.17.2 Features

6.17.2.1 Timer Function Features

- 12 sets of 32-bit timers, each timer having one 24-bit up counter and one 8-bit prescale counter
- Independent clock source for each timer
- Provides one-shot, periodic, toggle-output and continuous counting operation modes
- 24-bit up counter value is readable through CNT (TIMERx_CNT[23:0])
- Supports event counting function
- 24-bit capture value is readable through CAPDAT (TIMERx_CAP[23:0])
- Supports external capture pin event for interval measurement
- Supports external capture pin event to reset 24-bit up counter
- Supports chip wake-up from Idle/Power-down mode if a timer interrupt signal is generated
- Supports Timer0 ~ Timer11 time-out interrupt signal or capture interrupt signal to trigger EPWM, EADC and PDMA function
- Supports internal capture triggered from internal clock (HIRC, LIRC) or external clock (HXT, LXT)
- Supports Inter-Timer trigger mode

6.17.2.2 PWM Function Features

- Supports maximum clock frequency up to maximum PCLK
- Supports independent mode for PWM generator with two output channels
- Supports complementary mode for PWM generator with paired PWM output channel
  - 12-bit dead-time insertion with 12-bit prescale
- Supports 12-bit prescale from 1 to 4096
- Supports 16-bit PWM counter
  - Up, down and up-down count operation type
  - One-shot or auto-reload counter operation mode
- Supports mask function and tri-state enable for each PWM output pin
- Supports brake function
  - Brake source from pin and system safety events (clock failed, LVD interrupt flag

detection and CPU lockup)

- Brake pin noise filter control for brake source
- Edge detect brake source to control brake state until brake status cleared
- Level detect brake source to auto recover function after brake condition removed

- Supports interrupt on the following events:
  - PWM zero point, period point, up-count compared or down-count compared point events
  - Brake condition happened
- Supports trigger EADC on the following events:
  - PWM zero point, period, zero or period point, up-count compared or down-count compared point events

6.17.3 Block Diagram

The timer controller block diagram and clock control are shown as follows.

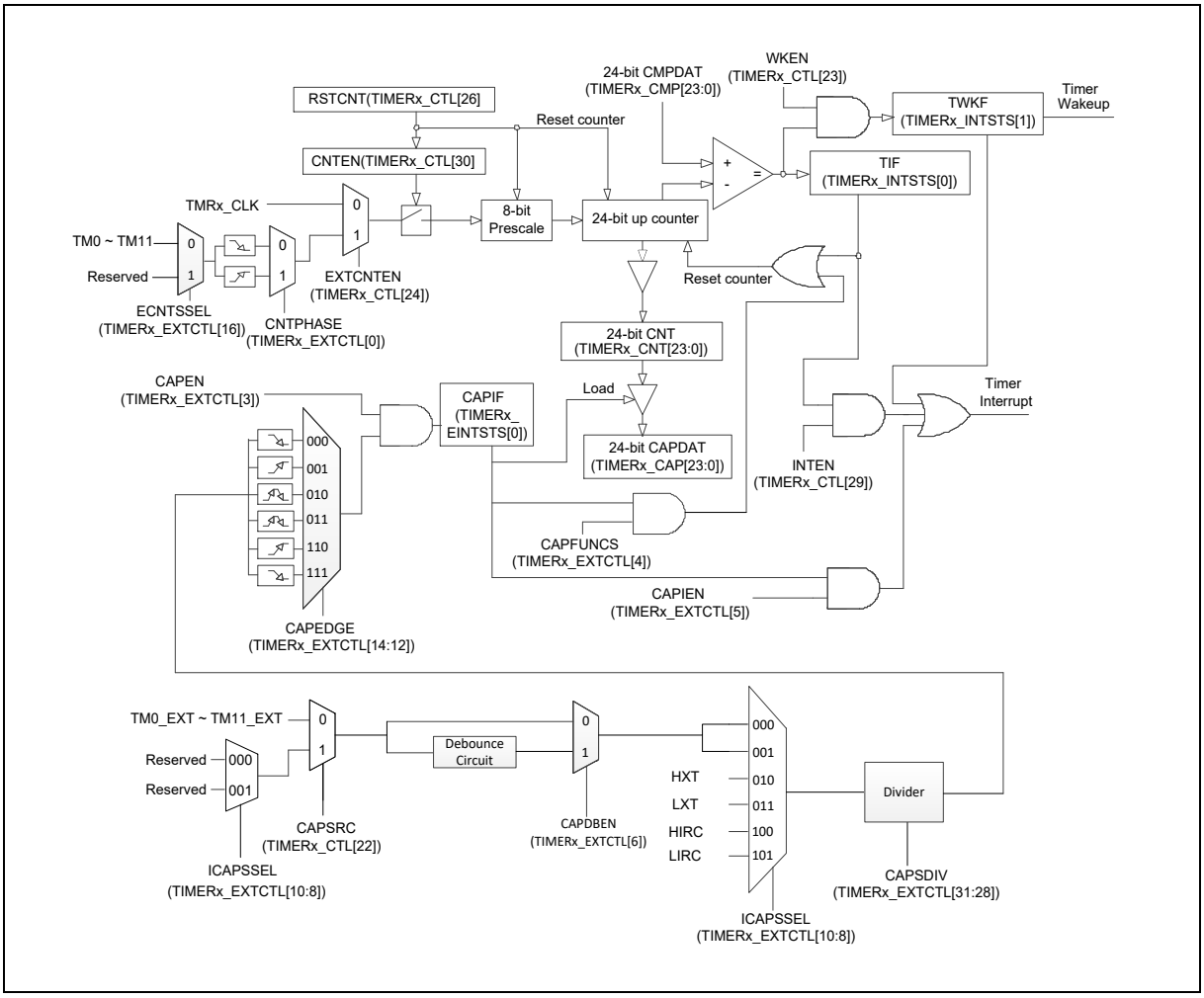
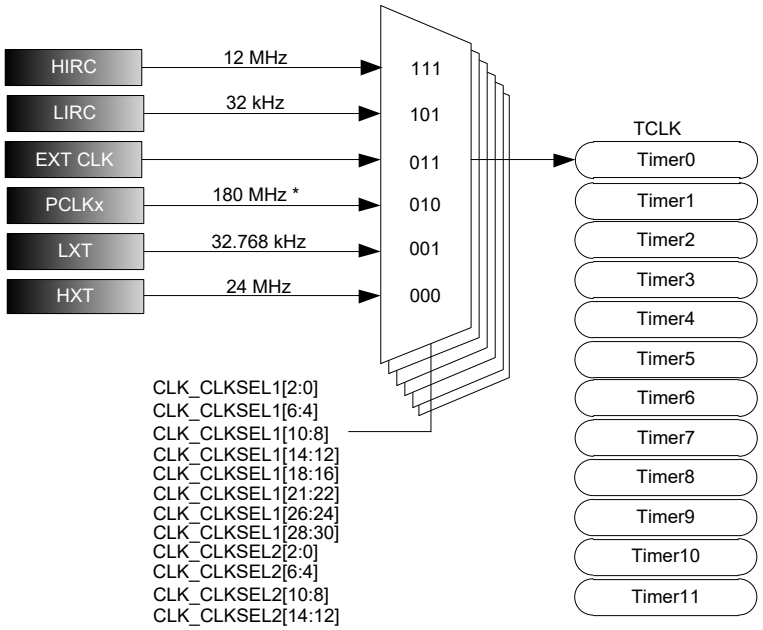


Figure 6.17-1 Timer Controller Block Diagram

Set FUNCSEL (TIMERx_ALTCTL[0]) 0 to enable timer mode. The clock source of Timer0 ~ Timer11 in timer mode can be enabled in TMRxCKEN (CLK_APBCLK0[11:0]) and selected as different frequency

in CLK_CLKSEL1[31:0] for Timer0 ~ Timer7, CLK_CLKSEL2[15:0]) for Timer8 ~ Timer11 as Figure 6.17-2.



Legend:

HXT = High Speed External clock signal
LXT = Low Speed External clock signal
HIRC = High Speed Internal clock signal
LIRC = Low Speed Internal clock signal

Note *: SYS CLK1 clock source is from SYS-PLL in this condition

Figure 6.17-2 Clock Source of Timer Controller

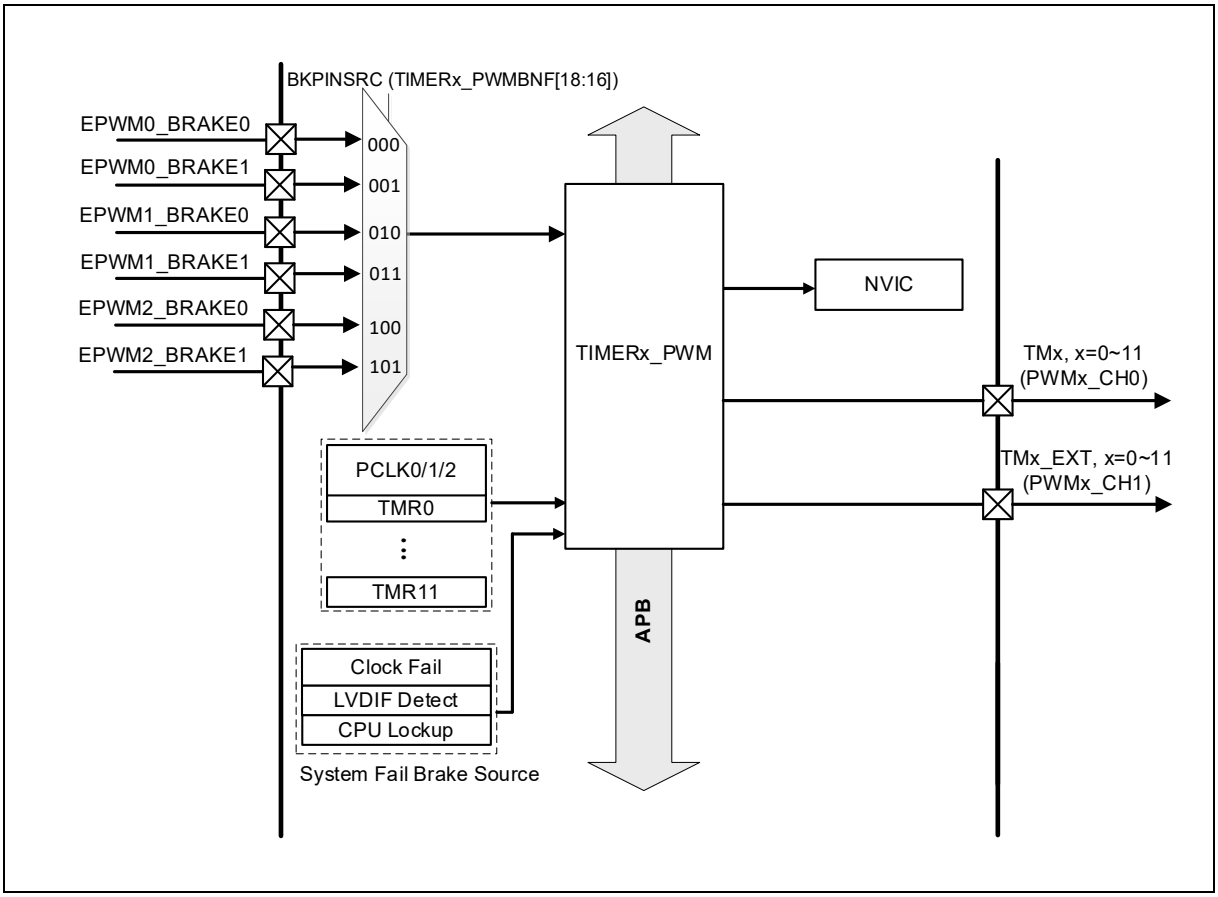


Figure 6.17-3 PWM Generator Overview Block Diagram

Set FUNCSEL (TIMERx_ALTCTL[0]) 1 to enable PWM mode. The clock source of Timer0 ~ Timer11 in PWM mode can be enabled in TMRxCKEN (CLK_APBCLK0[11:0]). TMR0/1/6/7_CLK clock sources are fixed to PCLK0. TMR2/3/8/9_CLK clock sources are fixed to PCLK1. TMR4/5/10/11_CLK clock sources are fixed to PCLK2. PWM system clock frequency will be PCLKx frequency as Figure 6.17-4 .

The clock source of PWM counter (TIMERx_PWMCLK) can be selected from PWM system clock (TMRx_CLK) or Timer interrupt events (TMRx_INT) as Figure 6.17-5.

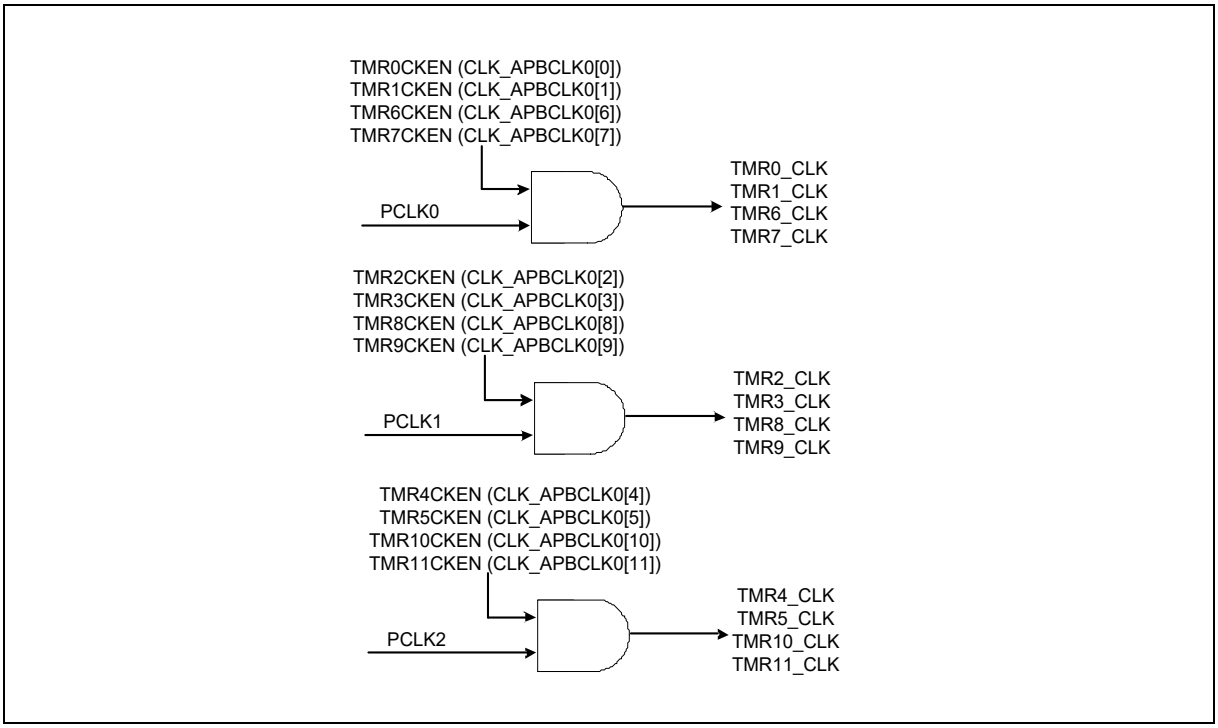


Figure 6.17-4 PWM System Clock Source Control

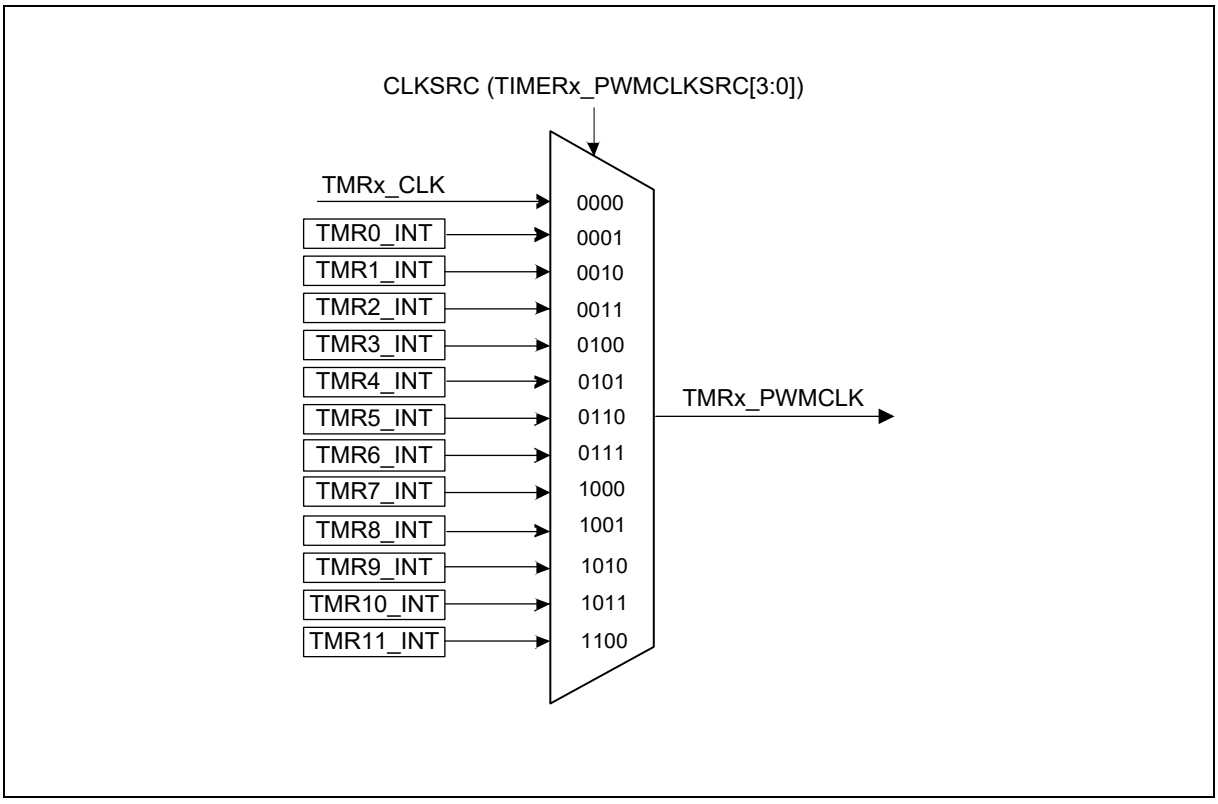


Figure 6.17-5 PWM Counter Clock Source Control

Figure 6.17-6 and Figure 6.17-7 illustrate the architecture of PWM independent mode and

complementary mode. Both independent mode and complementary mode supports PWMx_CH0 and PWMx_CH1 output channels in each PWM generator.

When counter counts to 0, PERIOD (TIMERx_PWMPERIOD[15:0]) or equal to CMP (TIMERx_PWMCMPDAT[15:0]), relative events will be generated. These events are passed to corresponding generators to generate PWM pulse (Pulse Generator), interrupt signal (Interrupt Generator) and trigger signal (Trigger Generator) for EADC to start conversion. Output Control block is used to decide PWM pulse output; brake function in Output Control block also generates interrupt events. And Dead-Time Control is available only in PWM complementary mode.

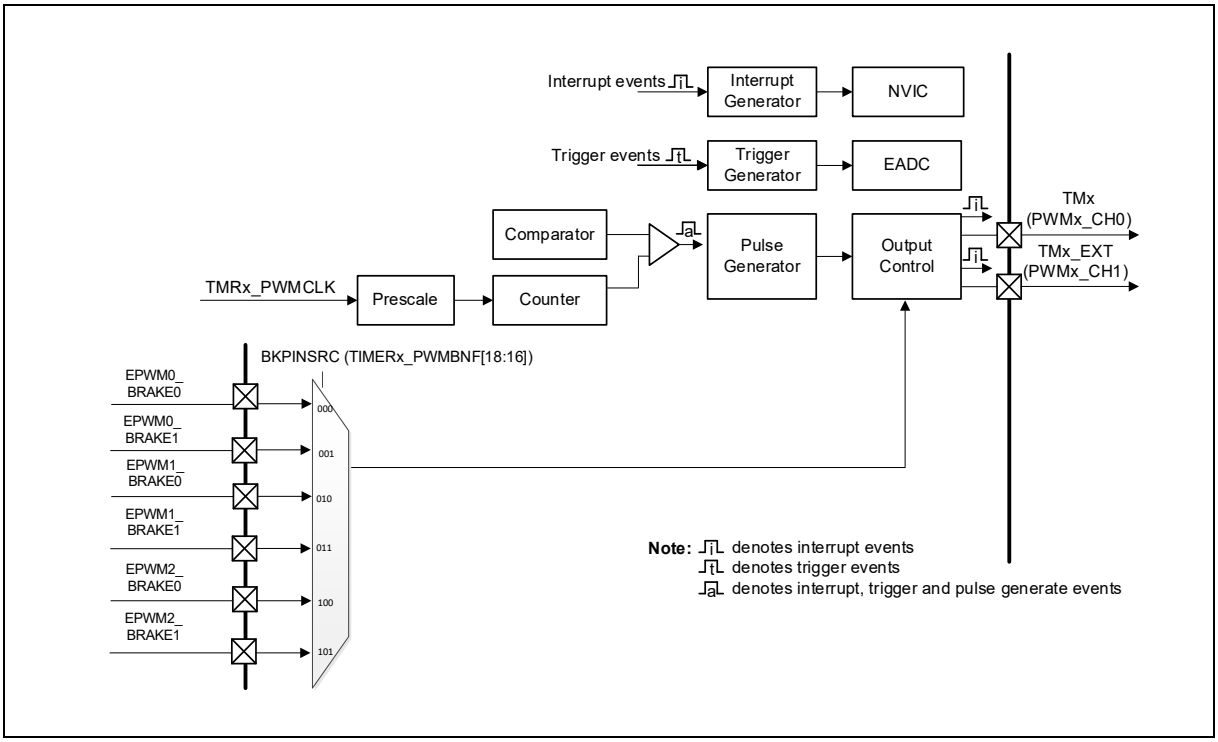


Figure 6.17-6 PWM Independent Mode Architecture Diagram

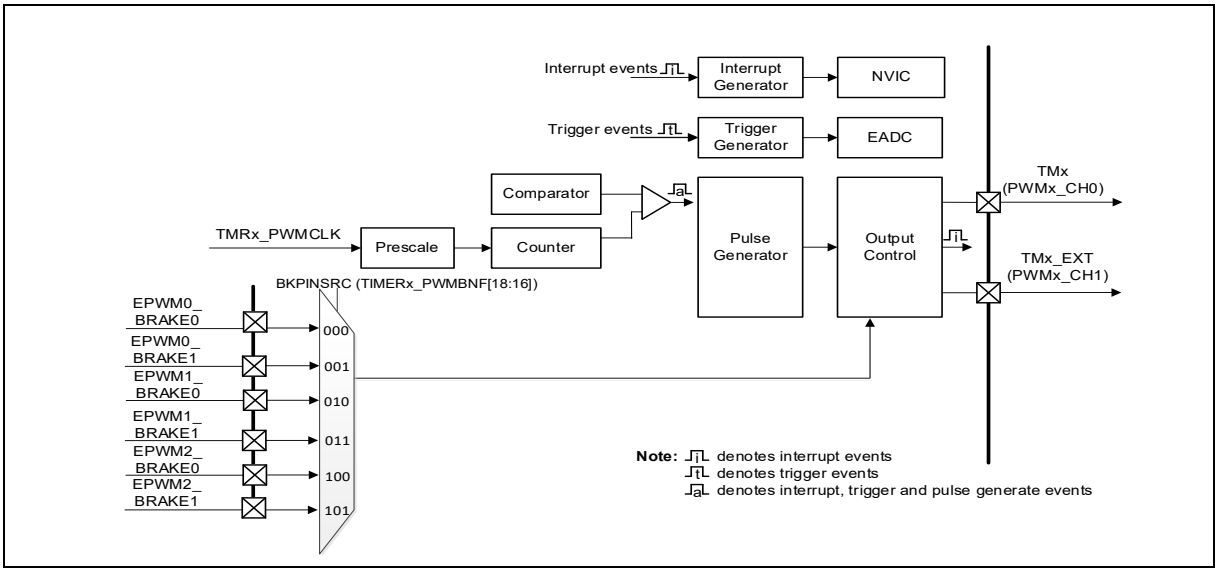


Figure 6.17-7 PWM Complementary Mode Architecture Diagram

6.17.4 Basic Configuration

Set FUNCSEL (TIMERx_ALTCTL[0]) 0 to enable timer mode. The clock source of Timer0 ~ Timer11 in timer mode can be enabled in TMRxCKEN (CLK_APBCLK0[11:0]) and selected as different frequency in TMR0SEL (CLK_CLKSEL1[2:0]) for Timer0, TMR1SEL (CLK_CLKSEL1[6:4]) for Timer1, TMR2SEL (CLK_CLKSEL1[10:8]) for Timer2, TMR3SEL (CLK_CLKSEL1[14:12]) for Timer3, TMR4SEL (CLK_CLKSEL1[18:16]) for Timer4, TMR5SEL (CLK_CLKSEL1[22:20]) for Timer5, TMR6SEL (CLK_CLKSEL1[26:24]) for Timer6, TMR7SEL (CLK_CLKSEL1[30:28]) for Timer7, TMR8SEL (CLK_CLKSEL2[2:0]) for Timer8, TMR9SEL (CLK_CLKSEL2[6:4]) for Timer9, TMR10SEL (CLK_CLKSEL2[10:8]) for Timer10 and TMR11SEL (CLK_CLKSEL2[14:12]) for Timer11.

Set FUNCSEL(TIMERx_ALTCTL[0]) 1 to enable PWM mode. The clock source of Timer0 ~ Timer11 in PWM mode can be enabled in TMRxCKEN (CLK_APBCLK0[11:0]). TMR0/1/6/7_CLK clock sources are fixed to PCLK0. TMR2/3/8/9_CLK clock sources are fixed to PCLK1. TMR4/5/10/11_CLK clock sources are fixed to PCLK2.

6.17.4.1 TIMER01 Basic Configuration

- Clock Source Configuration
  - Enable TIMER0 peripheral clock in TMR0CKEN (CLK_APBCLK0[0]).
  - Enable TIMER1 peripheral clock in TMR1CKEN (CLK_APBCLK0[1]).
- Reset Configuration
  - Reset TIMER0 controller in TMR0RST (SYS_IPRST1[2]).
  - Reset TIMER1 controller in TMR1RST (SYS_IPRST1[3]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
TMR0	TM0	PL.12, PI.0, PD.8, PF.14	MFP11
	TM0_EXT	PL.13, PI.1, PD.9, PB.9	MFP11
TMR1	TM1	PG.0, PA.15, PI.2	MFP11
	TM1_EXT	PG.1, PI.3	MFP11

Table 6.17-1 TIMER01 Pin Configuration

6.17.4.2 TIMER23 Basic Configuration

- Clock Source Configuration
  - Enable TIMER2 peripheral clock in TMR2CKEN (CLK_APBCLK0[2]).
  - Enable TIMER3 peripheral clock in TMR2CKEN (CLK_APBCLK0[3]).
- Reset Configuration
  - Reset TIMER2 controller in TMR2RST (SYS_IPRST1[4]).
  - Reset TIMER3 controller in TMR3RST (SYS_IPRST1[5]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
TM2	TM2	PL.14, PJ.12, PD.10	MFP11
	TM2_EXT	PL.15, PJ.13, PD.11	MFP11
TM3	TM3	PG.2, PJ.14, PL.6	MFP11

	TM3_EXT	PG.3, PJ.15, PL.7	MFP11

Table 6.17-2 TIMER23 Pin Configuration

6.17.4.3 TIMER45 Basic Configuration

- Clock Source Configuration
  - Enable TIMER4 peripheral clock in TMR4CKEN (CLK_APBCLK0[4]).
  - Enable TIMER5 peripheral clock in TMR5CKEN (CLK_APBCLK0[5]).
- Reset Configuration
  - Reset TIMER4 controller in TMR4RST (SYS_IPRST3[0]).
  - Reset TIMER5 controller in TMR5RST (SYS_IPRST3[1]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
TM4	TM4	PG.4, PL.8, PB.8	MFP11
	TM4_EXT	PG.5, PL.9	MFP11
TM5	TM5	PD.12, PL.0, PB.10	MFP11
	TM5_EXT	PD.13, PL.1, PB.11	MFP11

Table 6.17-3 TIMER45 Pin Configuration

6.17.4.4 TIMER67 Basic Configuration

- Clock Source Configuration
  - Enable TIMER6 peripheral clock in TMR6CKEN (CLK_APBCLK0[6]).
  - Enable TIMER7 peripheral clock in TMR7CKEN (CLK_APBCLK0[7]).
- Reset Configuration
  - Reset TIMER6 controller in TMR6RST (SYS_IPRST3[2]).
  - Reset TIMER7 controller in TMR7RST (SYS_IPRST3[3]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
TM6	TM6	PD.14, PK.6, PN.15	MFP11
	TM6_EXT	PD.15, PK.7	MFP11
TM7	TM7	PG.6, PK.0, PL.2	MFP11
	TM7_EXT	PG.7, PK.1, PL.3	MFP11

Table 6.17-4 TIMER67 Pin Configuration

6.17.4.5 TIMER89 Basic Configuration

- Clock Source Configuration
  - Enable TIMER8 peripheral clock in TMR8CKEN (CLK_APBCLK0[8]).
  - Enable TIMER9 peripheral clock in TMR9CKEN (CLK_APBCLK0[9]).

- Reset Configuration
  - Reset TIMER8 controller in TMR8RST (SYS_IPRST3[4]).
  - Reset TIMER9 controller in TMR9RST (SYS_IPRST3[5]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
TM8	TM8	PM.12, PK.8	MFP11
	TM8_EXT	PM.13, PK.5	MFP11
TM9	TM9	PL.4	MFP11
	TM9_EXT	PL.5	MFP11

Table 6.17-5 TIMER89 Pin Configuration

6.17.4.6 TIMER1011 Basic Configuration

- Clock Source Configuration
  - Enable TIMER10 peripheral clock in TMR10CKEN (CLK_APBCLK0[10]).
  - Enable TIMER11 peripheral clock in TMR11CKEN (CLK_APBCLK0[11]).
- Reset Configuration
  - Reset TIMER10 controller in TMR10RST (SYS_IPRST3[6]).
  - Reset TIMER11 controller in TMR11RST (SYS_IPRST3[7]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
TM10	TM10	PM.14, PK.12	MFP11
	TM10_EXT	PM.15, PK.13	MFP11
TM11	TM11	PK.14	MFP11
	TM11_EXT	PK.15	MFP11

Table 6.17-6 TIMER1011 Pin Configuration

6.17.5 Timer Functional Description

6.17.5.1 Timer Interrupt Flag

The timer controller supports the following interrupt flags; one is TIF (TIMERx_INTSTS[0]), which is set while timer counter value CNT (TIMERx_CNT[23:0]) matches the timer compared value CMPDAT (TIMERx_CMP[23:0]), and CAPIF (TIMERx_EINTSTS[0]) is set means when the transition on the TMx_EXT pin, or internal clock source (HXT, LXT, HIRC, LIRC) associated CAPEDGE (TIMERx_EXTCTL[14:12]) setting. The TWKF (TIMERx_INTSTS[1]) bit indicates the interrupt wake-up flag status of timer. Setting WKEN (TIMERx_CTL[23] to 1 can use wake-up function.

6.17.5.2 Timer Counting Mode

The timer controller provides four timer counting modes: one-shot, periodic, toggle-output and continuous counting operation modes:

6.17.5.3 One–shot Mode

If the timer controller is configured at one-shot mode (TIMERx_CTL[28:27] is 00) and CNTEN (TIMERx_CTL[30]) is set, the timer counter starts up counting. Once the CNT (TIMERx_CNT[23:0]) value reaches CMPDAT (TIMERx_CMP[23:0]) value, the TIF (TIMERx_INTSTS[0]) will be set to 1, CNT value and CNTEN bit is cleared automatically by timer controller then timer counting operation stops. In the meantime, if the INTEN (TIMERx_CTL[29]) is enabled, the timer interrupt signal is generated and sent to NVIC to inform CPU also.

User can monitor the counter reset operation active by RSTACT (TIMERx_CNT[31]). And set ICEDEBUG (TIMERx_CTL[31]) to 1 that disable ICE debug mode acknowledgement effects TIMER counting.

6.17.5.4 Periodic Mode

If the timer controller is configured at periodic mode (TIMERx_CTL[28:27] is 01) and CNTEN (TIMERx_CTL[30]) is set, the timer counter starts up counting. Once the CNT (TIMERx_CNT[23:0]) value reaches CMPDAT (TIMERx_CMP[23:0]) value, the TIF (TIMERx_INTSTS[0]) will be set to 1, CNT value will be cleared automatically by timer controller and timer counter operates counting again. In the meantime, if the INTEN (TIMERx_CTL[29]) bit is enabled, the timer interrupt signal is generated and sent to NVIC to inform CPU also. In this mode, the timer controller operates counting and compares with CMPDAT value periodically until the CNTEN bit is cleared by user.

User can set PERIOSEL (TIMERx_CTL[20]) to select Timer behavior at periodic mode.

6.17.5.5 Toggle-Output Mode

If the timer controller is configured at toggle-output mode (TIMERx_CTL[28:27] is 10) and CNTEN (TIMERx_CTL[30]) is set, the timer counter starts up counting. The counting operation of toggle-output mode is almost the same as periodic mode, except toggle-output mode has associated TM0 ~ TM3 or TM0_EXT ~ TM3_EXT pin to output signal while specify TIF (TIMERx_INTSTS[0]) is set. Use can set TGLPINSEL (TIMERx_CTL[21]) to choose Tx or Tx_EXT as toggle-output pin. Thus, the toggle-output signal is 50% duty cycle.

6.17.5.6 Continuous Counting Mode

If the timer controller is configured at continuous counting mode (TIMERx_CTL[28:27] is 11) and CNTEN (TIMERx_CTL[30]) is set, the timer counter starts up counting. Once the CNT (TIMERx_CNT[23:0]) value reaches CMPDAT (TIMERx_CMP[23:0]) value, the TIF (TIMERx_INTSTS[0]) will be set to 1 and CNT value keeps up counting. In the meantime, if the INTEN (TIMERx_CTL[29]) is enabled, the timer interrupt signal is generated and sent to NVIC to inform CPU also. User can change different CMPDAT value immediately without disabling timer counting and restarting timer counting in this mode.

For example, CMPDAT value is set as 80, first. The TIF will set to 1 when CNT value is equal to 80, timer counter is kept counting and CNT value will not goes back to 0, it continues to count 81, 82, 83,⋯ to 224 -1, 0, 1, 2, 3, ⋯ to 224 -1 again and again. Next, if user programs CMPDAT value as 200 and clears TIF, the TIF will set to 1 again when CNT value reaches to 200. At last, user programs CMPDAT as 500 and clears TIF, the TIF will set to 1 again when CNT value reaches to 500.

In this mode, the timer counting is continuous. Thus, this operation mode is called as continuous counting mode.

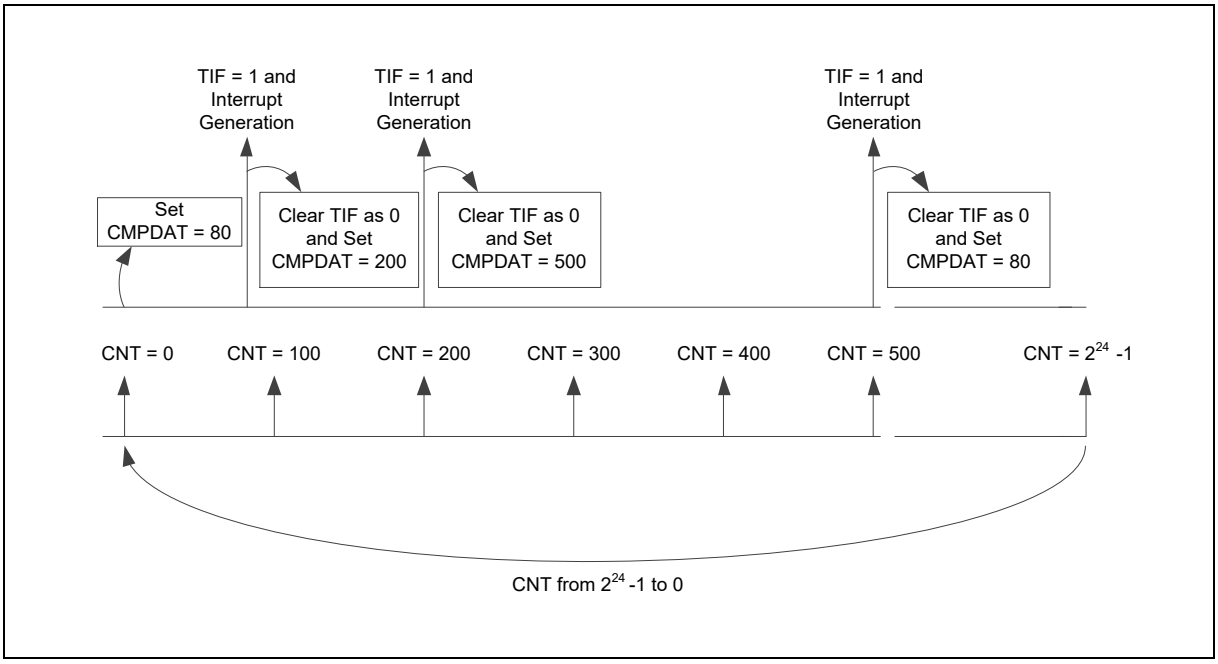


Figure 6.17-8 Continuous Counting Mode

6.17.5.7 Event Counting Mode

The timer controller also provides an application which can count the input event from TMx (x= 0~11) pin and the number of event will reflect to CNT (TIMERx_CNT[23:0]) value. It is also called as event counting function. In this function, EXTCNTEN (TIMERx_CTL[24]) should be set and the timer peripheral clock source should be set as PCLK.

If ECNTSSEL (TIMERx_EXTCTL[16]) is 0, the event counter source is from external TMx pin. User can enable or disable TMx pin de-bounce circuit by setting CNTDBEN (TIMERx_EXTCTL[7]). The input event frequency should be less than 1/3 PCLK if TMx pin de-bounce disabled or less than 1/8 PCLK if TMx pin de-bounce enabled to assure the returned CNT value is correct, and user can also select edge detection phase of TMx pin by setting CNTPHASE (TIMERx_EXTCTL[0]) bit.

In event counting mode, the timer counting operation mode can be selected as one-shot, periodic and continuous counting mode to counts the counter value CNT (TIMERx_CNT[23:0]) for TMx pin.

6.17.5.8 External Capture Mode

The event capture function is used to load CNT (TIMERx_CNT[23:0]) value to CAPDAT (TIMERx_CAP[23:0]) value while edge transition detected on capture source. In this mode, CAPEN (TIMERx_EXTCTL[3]) should be as 1 and CAPFUNCS (TIMERx_EXTCTL[4]) should be as 0 to trigger event capture function and the timer peripheral clock source should be set as PCLK.

If CAPSRC (TIMERx_CTL[22]) is 0, the capture event is triggered by TMx_EXT pin transition. User can enable or disable TMx_EXT pin de-bounce circuit by setting CAPDBEN (TIMERx_EXTCTL[6]). The transition frequency of TMx_EXT pin should be less than 1/3 PCLK if TMx_EXT pin de-bounce disabled or less than 1/8 PCLK if TMx_EXT pin de-bounce enabled to assure the capture function can be work normally, and user can also select edge transition detection of TMx_EXT pin by setting CAPEDGE (TIMERx_EXTCTL[14:12]).

If CAPSRC (TIMERx_CTL[22]) is 1, the capture event can be triggered by internal output signal transition on capture sources are HXT, LXT, HIRC, LIRC if ICAPSSEL is 010, 011, 100, 101 respectively. User can switch capture source only when original source and target source is all off. For example, if user wants to switch LXT or LIRC, both of capture source should be set off first.

In event capture mode, user does not consider what timer counting operation mode is selected, and the capture event occurred only if edge transition on capture source is detected.

User can enable CAPIEN (TIMERx_EXTCTL[5]) to use capture interrupt fuction. When the capture source edge transition meets setting, CAPIF is high.

User must consider the Timer will keep register TIMERx_CAP unchanged and drop the new capture value, if the CPU does not clear the CAPIF status.

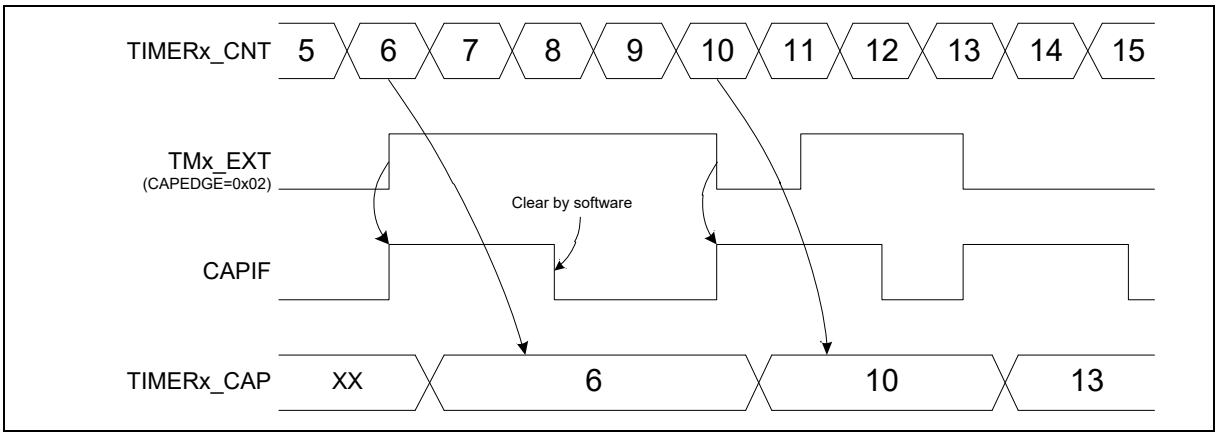


Figure 6.17-9 External Capture Mode on TMx_EXT Pin

6.17.5.9 External Reset Counter Mode

The timer controller also provides reset counter function to load CNT (TIMERx_CNT[23:0]) value to CAPDAT (TIMERx_CAP[23:0]) then reset CNT value while capture event is generated. In this mode, CAPFUNCS (TIMERx_EXTCTL[4]) should be as 1 for select TMx_EXT transition or internal clock source (HXT, LXT, HIRC, LIRC) to trigger reset counter value.

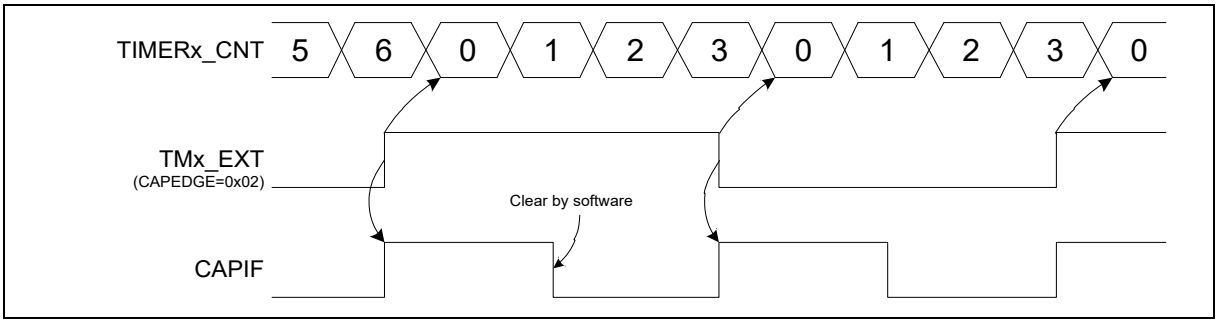


Figure 6.17-10 External Reset Counter Mode

6.17.5.10 Timer Trigger Function

The timer controller provides timer time-out interrupt or capture interrupt to trigger EPWM, EADC and PDMA. If TRGSSEL (TIMERx_TRGCTL[0]) is 0, time-out interrupt signal is used to trigger EPWM, EADC and PDMA. If TRGSSEL (TIMERx_TRGCTL[0]) is 1, capture interrupt signal is used to trigger EPWM, EADC and PDMA.

When the TRGEADC (TIMERx_TRGCTL[2]) is set, if the timer interrupt signal is generated, the timer controller will trigger EADC to start converter.

When the TRGPDMA (TIMERx_TRGCTL[4]) is set, if the timer interrupt signal is generated, the timer controller will trigger PDMA.

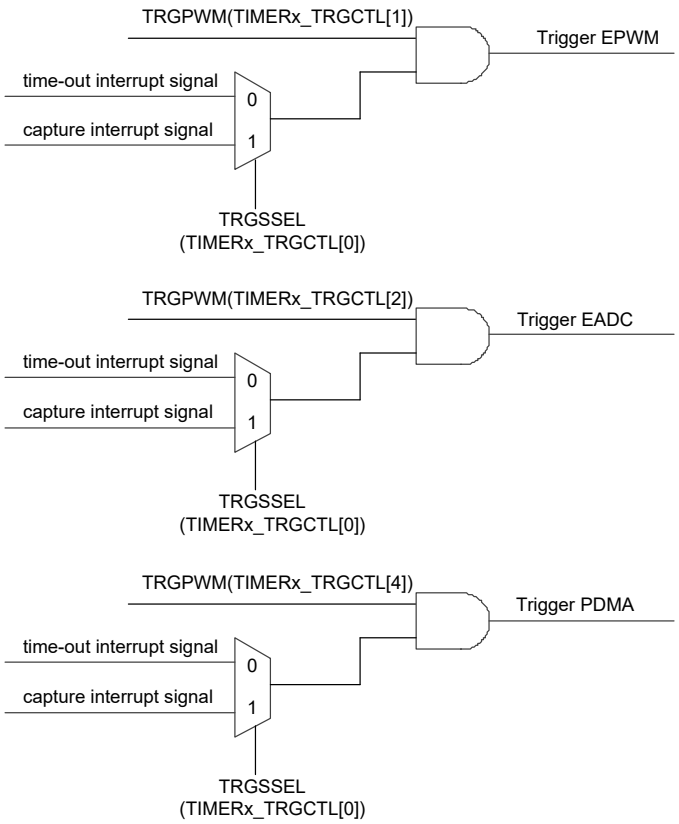


Figure 6.17-11 Internal Timer Trigger

6.17.5.11 Inter-Timer Trigger Capture Mode

In this mode, the Timer0/2/4/6/8/10 will be forced in event counting mode, counting with external event, and will generate an internal signal (INTR_TMR_TRG) to trigger Timer1/3/5/7/9/11 start or stop counting. Also, the Timer1/3/5/7/9/11 will be forced in capture mode and start/stop trigger-counting by Timer0/2/4/6/8/10 counter status.

Setting Timer0 Inter-timer Trigger Capture enabled, trigger-counting capture function is forced on Timer1. Setting Timer2 Inter-Timer Trigger enabled, trigger-counting capture function is forced on Timer3. Setting Timer4 Inter-Timer Trigger enabled, trigger-counting capture function is forced on Timer5. Setting Timer6 Inter-Timer Trigger enabled, trigger-counting capture function is forced on Timer7. Setting Timer8 Inter-Timer Trigger enabled, trigger-counting capture function is forced on Timer9. Setting Timer10 Inter-Timer Trigger enabled, trigger-counting capture function is forced on Timer11.

Start Trigger

While INTRGEN (TIMERx_CTL[19]) in Timer0/2/4/6/8/10 is set, the Timer0/2/4/6/8/10 will make a rising-edge transition of INTR_TMR_TRG while Timer0/2/4/6/8/10 24-bit counter value (CNT) is counting from 0x0 to 0x1 and Timer1/3/5/7/9/11 counter will start counting immediately and automatically.

Stop Trigger

When Timer0/2/4/6/8/10 CNT reaches the Timer0/2/4/6/8/10 CMPDAT value, the Timer0/2/4/6/8/10 will

make a falling-edge transition of INTR_TMR_TRG. Then Timer0/2/4/6/8/10 counter mode function will be disabled and INTRGEN (TIMERx_CTL[19]) will be cleared by hardware then Timer1/3/5/7/9/11 will stop counting also. At the same time, the Timer1/3 CNT value will be saved into Timer1/3/5/7/9/11 CAPDAT (TIMERx_CAP[23:0]).

User can use inter-timer trigger mode to measure the period of external event (TMx) more precisely. Figure 6.17-12 shows the sample flow of Inter-Timer Trigger Capture Mode for Timer0 as event counting mode and Timer1 as trigger-counting capture mode.

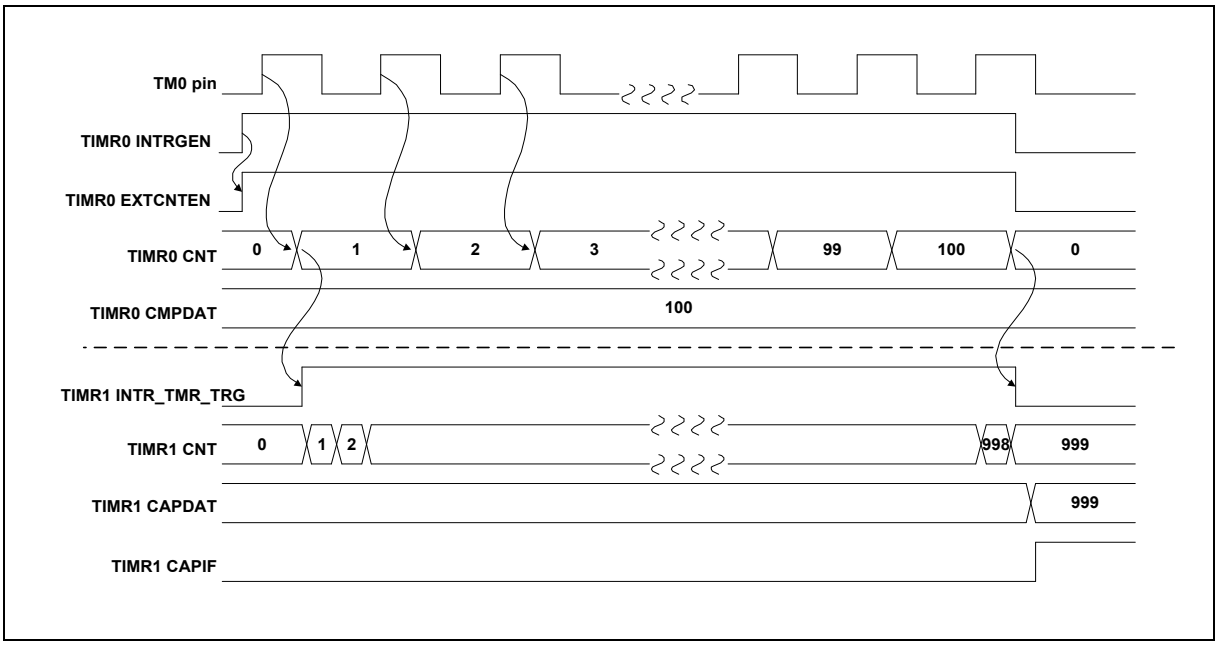


Figure 6.17-12 Inter-Timer Trigger Capture Timing

6.17.6 PWM Functional Description

6.17.6.1 PWM Prescale

The PWM prescale is used to divide clock source, and the clock of PWM counter is divided by (CLKPSC+ 1). The prescale is set by CLKPSC (TIMERx_PWMCLKPSC[11:0]). Figure 6.17-13 shows an example of PWM prescale waveform in up count type.

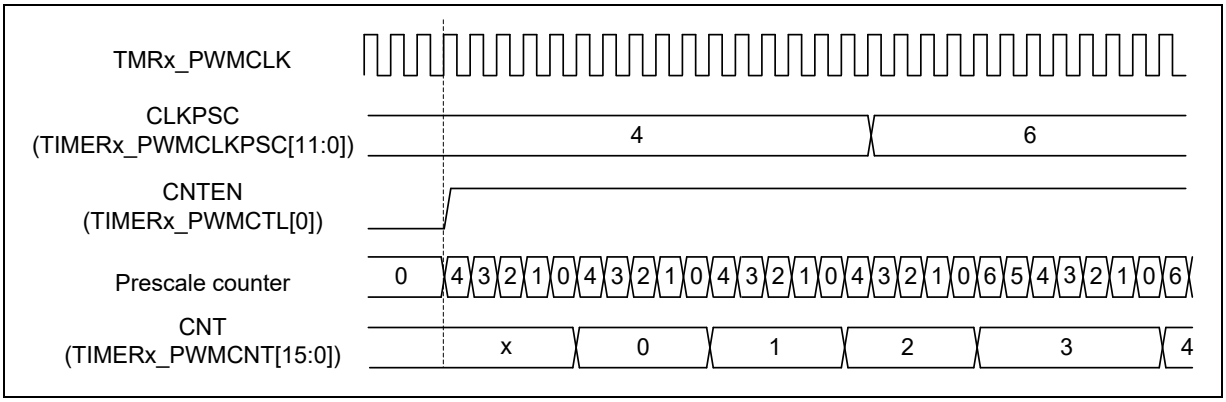


Figure 6.17-13 PWM Prescale Waveform in Up Count Type

6.17.6.2 PWM Counter

The PWM supports three counter types operation: up count, down count and up-down count types.

6.17.6.3 Up Count Type

When PWM counter is set to up count type, CNTTYPE (TIMERx_PWMCTL[2:1]) is 0x0, it starts up-counting from 0 to PERIOD (TIMERx_PWMPERIOD[15:0]). The current counter value can be read from the CNT (TIMERx_PWMCNT[15:0]). PWM generates a zero point event when both counter and prescale counts to 0. PWM generates a period point event when the counter counts to PERIOD and prescale counts to 0. Figure 6.17-14 shows an example of PWM up count type, where PWM period time is (PERIOD+1) * (CLKPSC+1) * TMRx_PWMCLK.

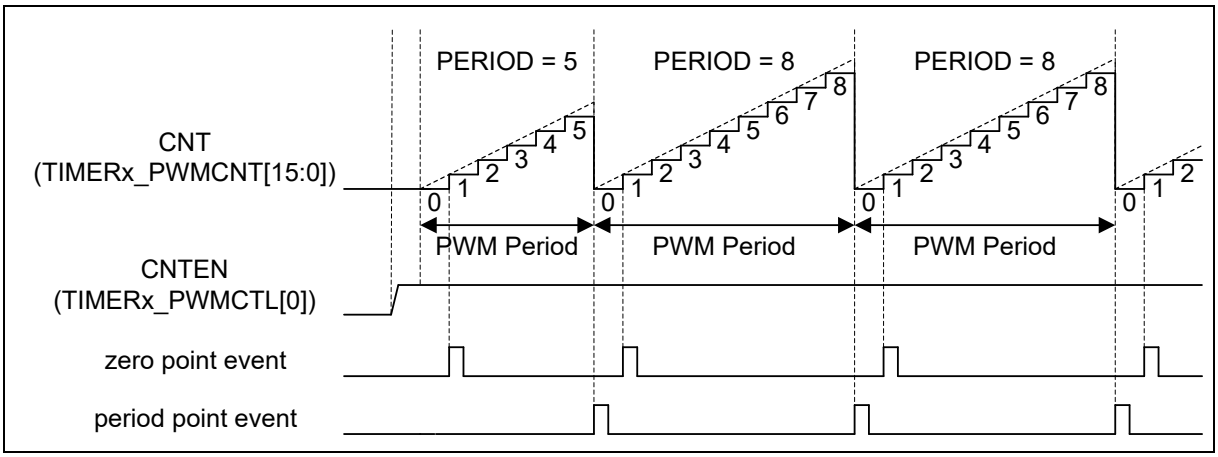


Figure 6.17-14 PWM Up Count Type

6.17.6.4 Down Count Type

When PWM counter is set to down count type, CNTTYPE (TIMERx_PWMCTL[2:1]) is 0x1, it starts down-counting from PERIOD to 0, current counter value can be read from CNT (TIMERx_PWMCNT[15:0]). PWM generates a zero point event when both counter and prescale counts to 0. PWM generates a period point event when the counter counts to PERIOD and prescale counts to 0. Figure 6.17-15 is an example of PWM down count type, where PWM period time is (PERIOD+1) * (CLKPSC+1) * TMRx_PWMCLK.

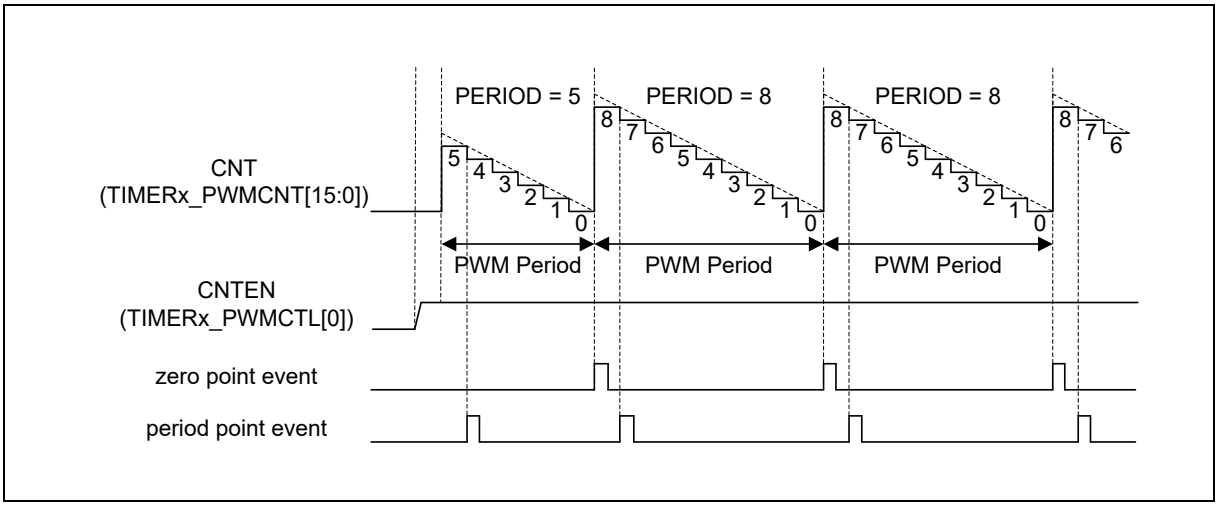
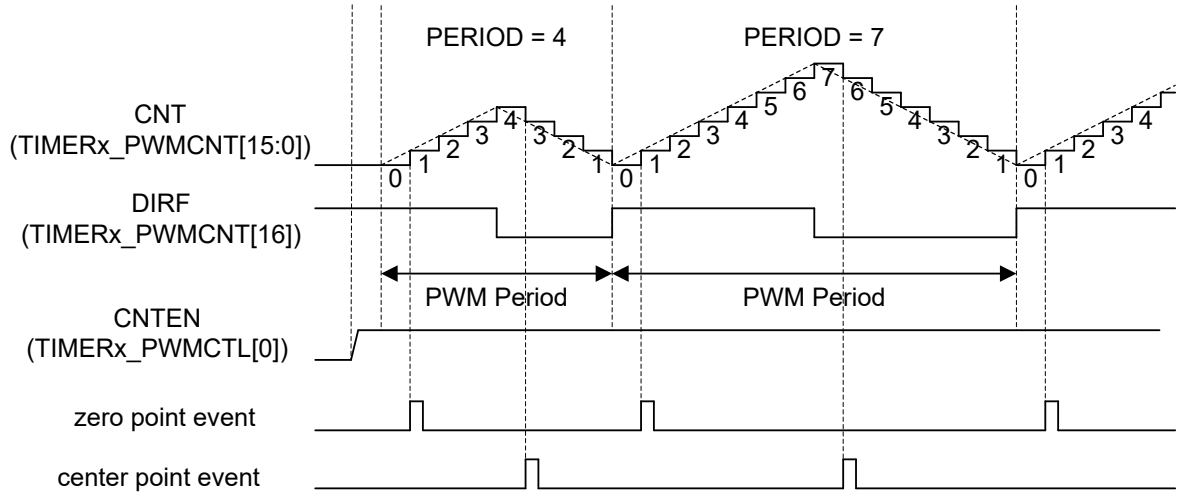


Figure 6.17-15 PWM Down Count Type

6.17.6.5 Up-Down Count Type

When PWM counter is set to up-down count type, CNTTYPE (TIMERx_PWMCTL[2:1]) is 0x2, it starts counting up from 0 to PERIOD and then starts counting down to 0. The current counter value can be read from CNT (TIMERx_PWMCNT[15:0]). PWM generates a zero point event when both counter and

prescale counts to 0. PWM generates a center point event when the counter counts to PERIOD and prescale counts to 0. Figure 6.17-16 shows an example of PWM up-down count type, where PWM period time is (2 * PERIOD) * (CLKPSC+1) * TMRx_PWMCLK. The DIRF (TIMERx_PWMCNT[16]) is counter direction indicator flag, where 1 is up counting, and 0 is down counting.



Note: When in up-down count type, period interrupt flag occurs at center point event.

Figure 6.17-16 PWM Up-Down Count Type

6.17.6.6 PWM Counter Operation mode

The PWM counter supports two operation modes: one-shot mode and auto-reload mode. PWM counter will operate in one-shot mode if CNTMODE (TIMERx_PWMCTL[3]) bit is set to 1, and operate in auto-reload mode if CNTMODE bit is set to 0.

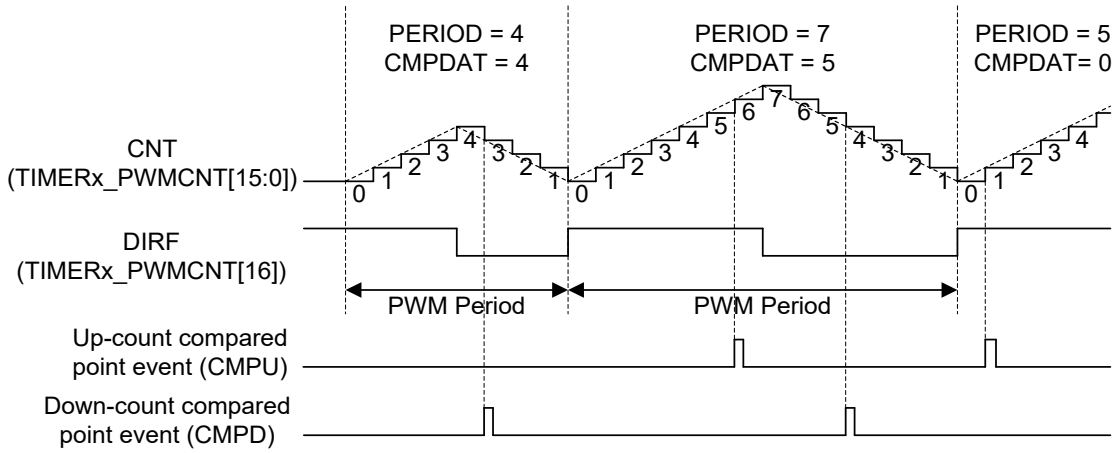
In both modes, CMP (TIMERx_PWMCMPDAT[15:0]) and PERIOD (TIMERx_PWMPERIOD[15:0]) should be written first and then set CNTEN (TIMERx_PWMCTL[0]) bit to 1 to start counter running.

In one-shot mode, PWM counter value will reload to default value according count type after one PWM period is completed. User can write CMP to continuous one-shot operation to generate next one-shot pulse once no matter current one-shot counter is running or completed.

In auto-reload mode, PWM counter is continuous running with current active PERIOD and CMP. If user sets PERIOD to zero in auto-reload mode, PWM counter value will reload to default value according count type after one PWM period is completed.

6.17.6.7 PWM Comparator

The CMP (TIMERx_PWMCMPDAT[15:0]) is comparator register of PWM. The CMP value is continuously compared to the corresponding counter value. When the counter is equal to CMP, PWM generates a compared point event. This event will generate PWM output pulse, interrupt signal or trigger EADC start convert. In up-down count type, two events will be generated in a PWM period as shown in Figure 6.17-17. The CMPU is up count compared point event and CMPD is down count compared point event.



Note: No CMPU event occurred when CMPDAT equals to PERIOD.

Figure 6.17-17 PWM Comparator Events in Up-Down Count Type

6.17.6.8 Period Loading Mode

When the IMMLDEN (TIMERx_PWMCTL[9]) bit set to 0, PWM operates at period loading mode. The PWM provides PBUF (TIMERx_PWMPBUF[15:0]) is the active PERIOD buffer register and CMPBUF (TIMERx_PWMCMPBUF[15:0]) is the active CMP buffer register. In period loading mode, both PERIOD (TIMERx_PWMPERIOD[15:0]) and CMP (TIMERx_PWMCMPDAT[15:0]) will load to their active PBUF and CMPBU register while each PWM period is completed. Figure 6.17-18 shows period loading timing of up count type, where PERIOD DATA0 denotes the initial data of PERIOD, PERIOD DATA1 denotes the first updated PERIOD data by user and so on, CMP also follows this rule. The following steps are the sequence of Figure 6.17-18.

1. User writes CMP DATA1 to CMP at point 1.
2. Period loading CMP DATA1 to CMPBUF at the end of PWM period at point 2.
3. User writes PERIOD DATA1 to PERIOD at point 3.
4. Period loading PERIOD DATA1 to PBUF at the end of PWM period at point 4.
5. User writes PERIOD DATA2 to PERIOD at point 5.
6. Period loading PERIOD DATA2 to PBUF at the end of PWM period at point 6.

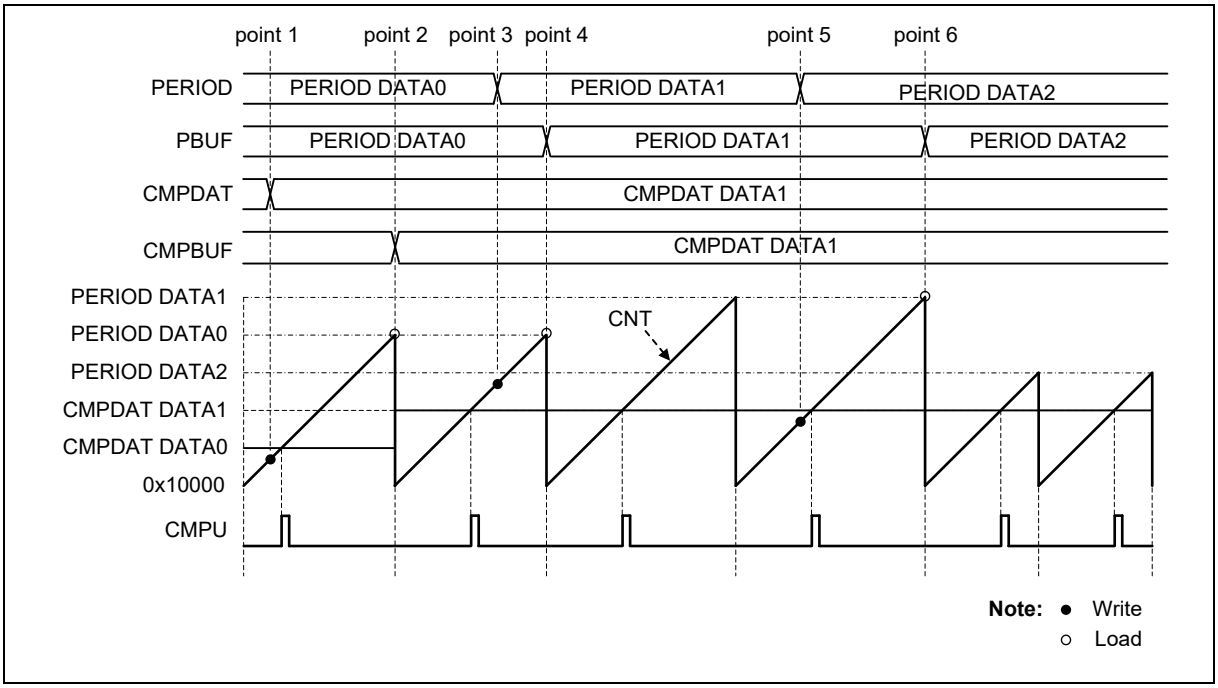


Figure 6.17-18 Period Loading Mode with Up Count Type

6.17.6.9 Immediately Loading Mode

When the IMMLDEN (TIMERx_PWMCTL[9]) bit set to 1, PWM operates at immediately loading mode. In immediately loading mode, when user update PERIOD (TIMERx_PWMPERIOD[15:0]) or CMP (TIMERx_PWMCMPDAT[15:0]), PERIOD or CMP will be load to active PBUF (TIMERx_PWMPBUF[15:0]) or CMPBUF (TIMERx_PWMCMPBUF[15:0]) after current counter count is completed. If the update PERIOD value is less than current counter value, counter will count wraparound. The following steps are the sequence of Figure 6.17-19.

1. User writes CMP DATA1 at point 1 and hardware will load CMP DATA1 to CMPBUF after current counter count is completed.
2. User writes PERIOD DATA1 at point 2 and PERIOD DATA1 is greater than current counter value, PWM counter will continuously count until equal to PERIOD DATA1 to complete one PWM period.
3. User writes PERIOD DATA2 at point 3 and PERIOD DATA2 is less than the current counter value, PWM counter will continuously count to maximum counter value 0x1FFFF and wraparound from 0x10000 to PERIOD DATA2 to complete one PWM period.

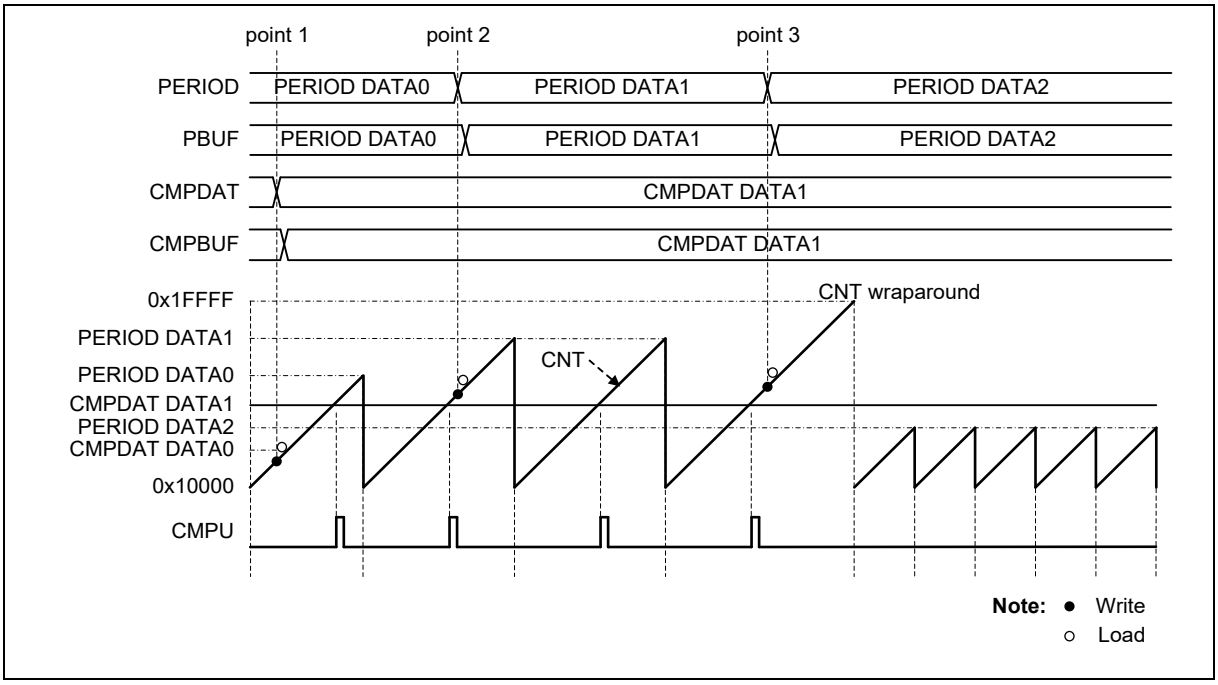


Figure 6.17-19 Immediately Loading Mode with Up Count Type

6.17.6.10 PWM Pulse Generator

The PWM pulse generator uses counter and comparator events to generate PWM output pulse. The events are zero point and period point in up count type and down count type, center point in up-down count type and counter equal to comparator point in three count types.

Each event point can generate PWM output waveform in differente count type as shown in Figure 6.17-20, Figure 6.17-21 and Figure 6.17-22.

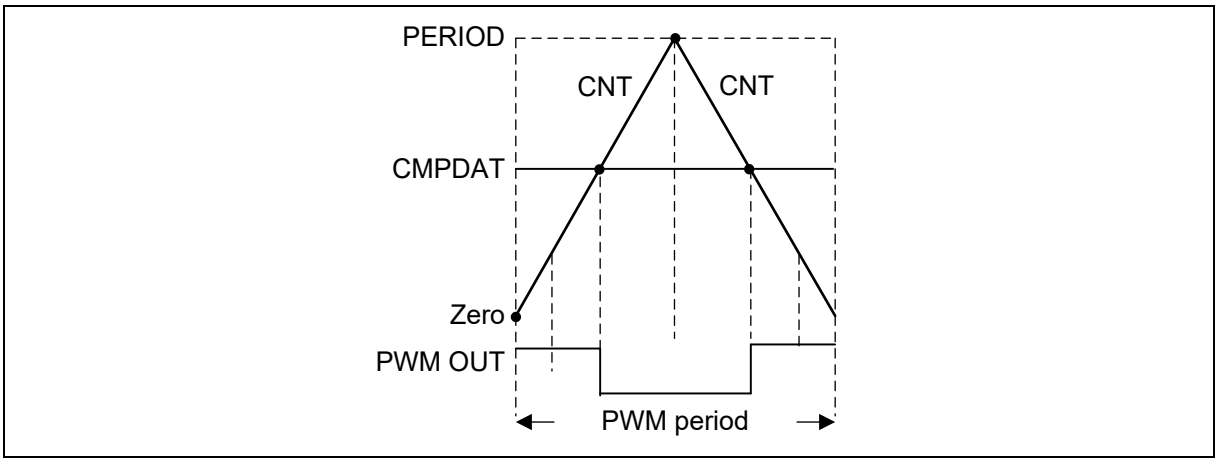


Figure 6.17-20 PWM Pulse Generation in Up-Down Count Type

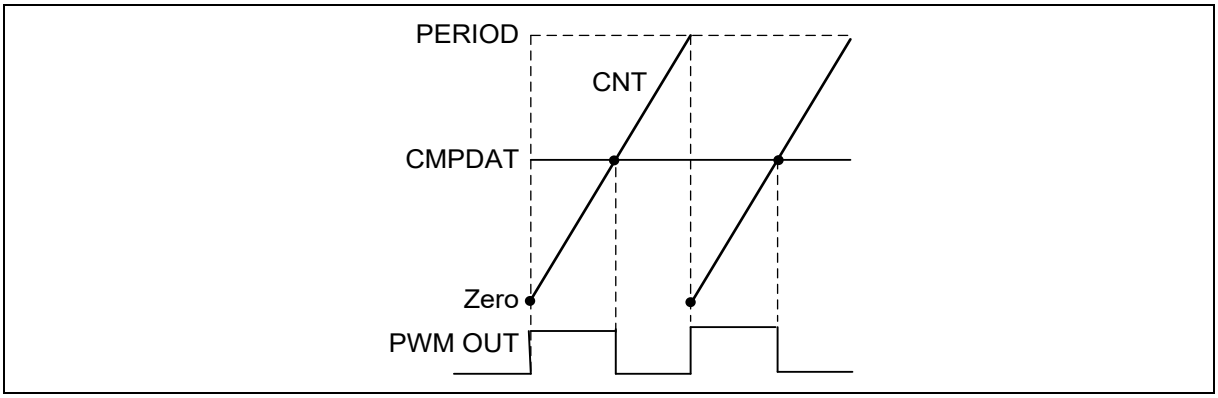


Figure 6.17-21 PWM Pulse Generation in Up Count Type

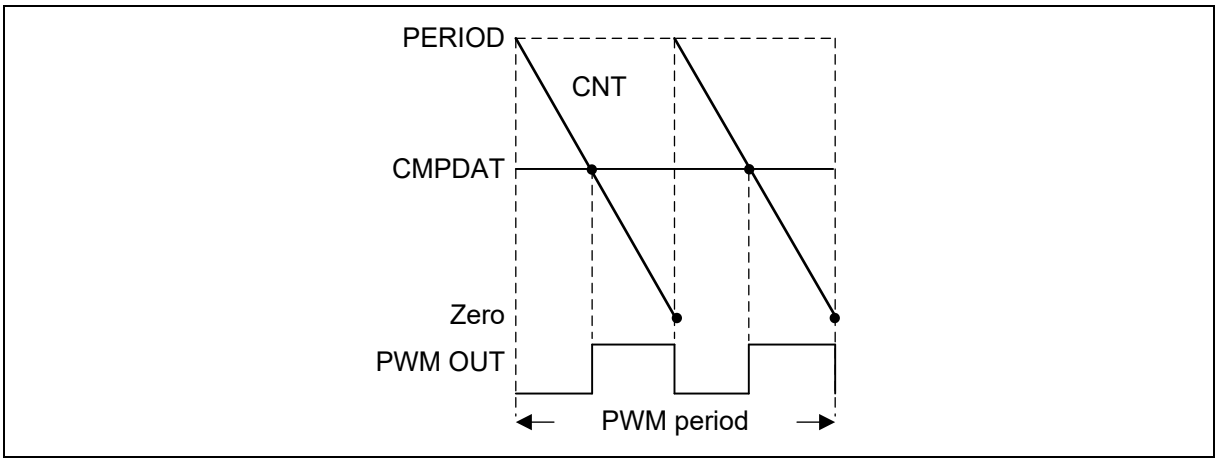


Figure 6.17-22 PWM Pulse Generation in Down Count Type

The PWM generation events may be sometimes generated at the same time, as the reason, events priority between different counter types should be taken care are listed in Table 6.17-3, Table 6.17-4 and Table 6.17-5, including event priority in up count type, event priority in down count type and event priority in up-down count type.

Priority	Zero And CMPU Point Event (CMP = 0)	PWM Output
1 (High)	Compare up event	Low
2 (Low)	Zero event	High

Table 6.17-3 PWM Pulse Generation Event Priority in Up Count Type

Priority	Zero And CMPD Point Event (CMP = 0)	PWM Output
1 (High)	Zero event	Low
2 (Low)	Compare down event	High
Priority	Period and CMPD point event (CMP = PERIOD)	PWM output

1 (High)	Compare down event	High
2 (Low)	Period event	Low

Table 6.17-4 PWM Pulse Generation Event Priority in Down Count Type

Priority	CMPU And CMPU Point Event (CMP = PERIOD)	PWM Output
1 (High)	Compare down event	High
2 (Low)	Compare up event	Low

Table 6.17-5 PWM Pulse Generation Event Priority in Up-Down Count Type

According to event priority limitation, the PWM generator can support 0% and 100% duty cycle PWM output waveform only in up count and up-down count type. Figure 6.17-23 is an example about PWM duty cycle from 0% to 100% in up count type and up-down count type where PERIOD is 4 with different CMP value.

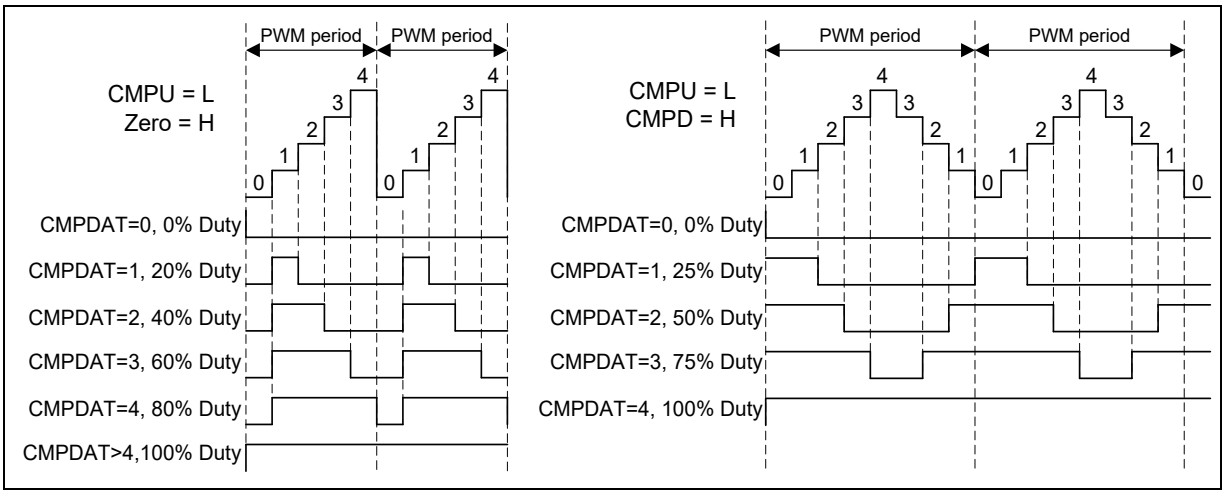


Figure 6.17-23 PWM 0% to 100% Duty Cycle in Up Count Type and Up-Down Count Type

6.17.6.11 PWM Output Mode

The PWM supports two output modes: independent mode that may be applied to DC motor system, complementary mode with dead-time insertion that may be used in the application of AC induction motor and permanent magnet synchronous motor.

6.17.6.12 Independent Mode

When OUTMODE (TIMERx_PWMCTL[16]) bit is set to 0, PWM output operates in independent mode. In this mode, both PWMx_CH0 and PWMx_CH1 can output the same waveform as shown in Figure 6.17-24.

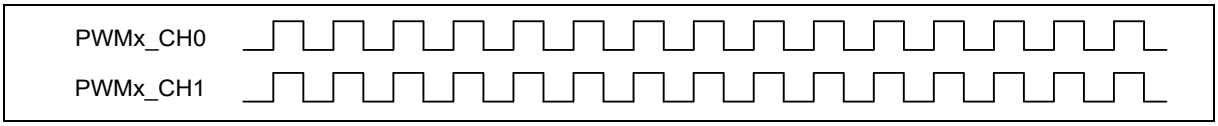


Figure 6.17-24 PWM Independent Mode Output Waveform

6.17.6.13 Complementary Mode

When OUTMODE (TIMERx_PWMCTL[16]) bit is set to 1, PWM output operates in complementary mode. In this mode, both PWMx_CH0 and PWMx_CH1 can output waveform and PWMx_CH1 must always be the complement of PWMx_CH0 as shown in Figure 6.17-25.

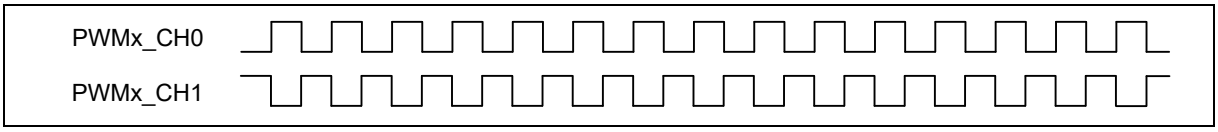


Figure 6.17-25 PWM Complementary Mode Output Waveform

6.17.6.14 PWM Output Control

After PWM pulse generator, there are four steps to control output waveform in independent output mode and five control steps in complementary output mode. User can set POEN0 (TIMERx_PWMPOEN[0]) and POEN1 (TIMERx_PWMPOEN[1]) 1 to enable PWMx_CH0 and PWMx_CH1 output waveform.

In Independent mode, there are mask control, brake control, polarity control and output enable control to control output waveform as shown in Figure 6.17-26.

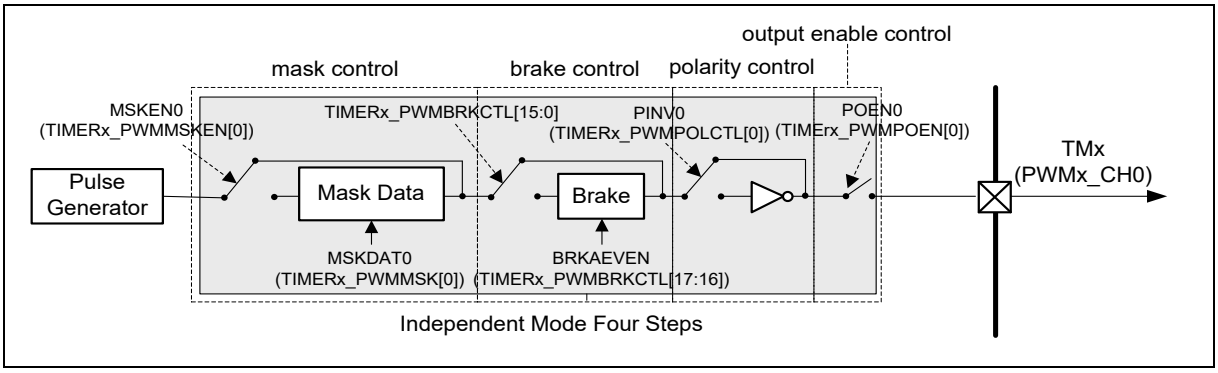


Figure 6.17-26 PWMx_CH0 Output Control in Independent Mode

In complementary mode, there are dead-time insertion control and four control steps the same as independent mode to control PWMx_CH0 and PWMx_CH1 outputs as shown in Figure 6.17-27.

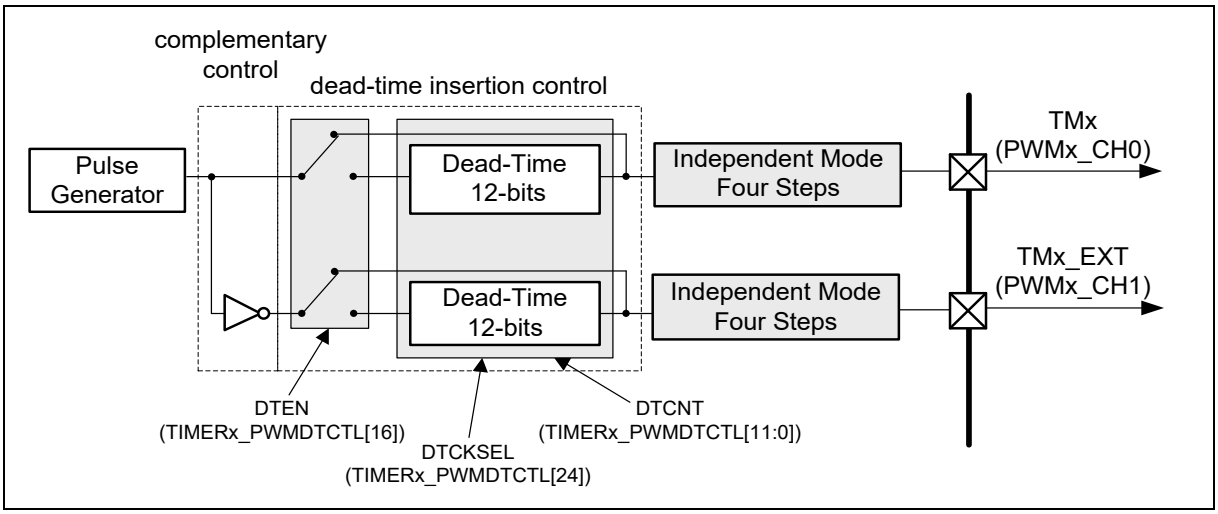


Figure 6.17-27 PWMx_CH0 and PWMx_CH1 Output Control in Complementary Mode

6.17.6.15 Dead-Time Insertion Control

In the complementary application, the complement channels may drive the external devices like power switches. The dead-time generator inserts a low level interval between complementary outputs PWMx_CH0 and PWMx_CH1 as shown in Figure 6.17-28. User sets DTEN (TIMERx_PWMDTCTL[16]) bit to enable dead-time control function, DTCNT (TIMERx_PWMDTCTL[11:0]) and DTCKSEL (TIMERx_PWMDTCTL[24]) to control dead-time interval. The dead-time interval can be calculated from the following formula:

Dead-time interval = (DTCNT + 1) * TMRx_PWMCLK period, if DTCKSEL is 0

Dead-time interval = (DTCNT + 1) * TMRx_PWMCLK * (CLKPSC + 1) period, if DTCKSEL is 1

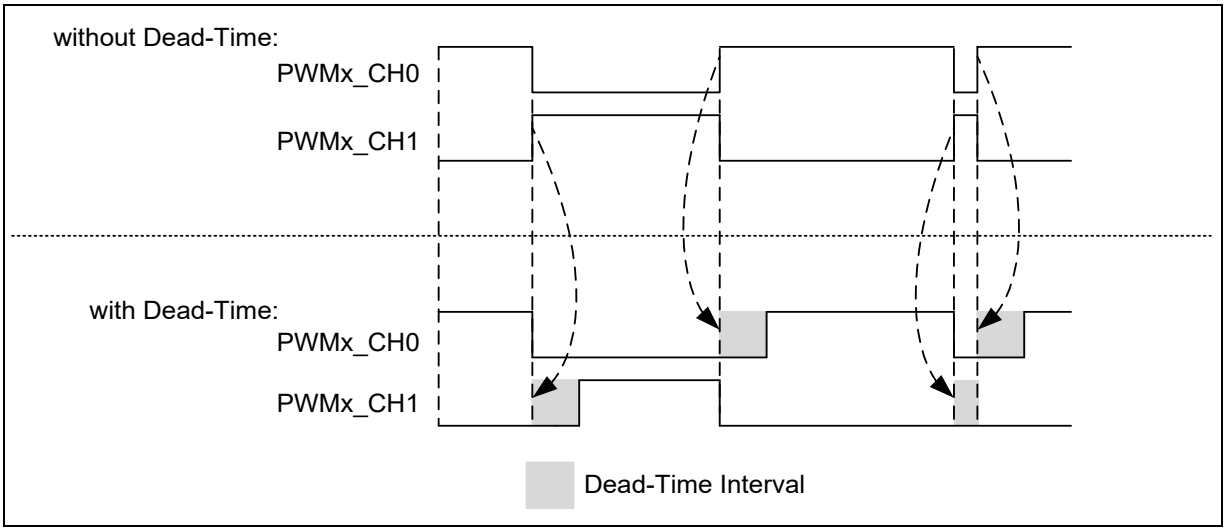


Figure 6.17-28 Dead-Time Insertion

6.17.6.16 PWM Mask Output Control

PWMx_CH0/CH1 output value can be masked to specified logic states by setting MSKEN0/1 (TIMERx_PWMMSKEN[1:0]) and MSKDAT0/1 (TIMERx_PWMMSK[1:0]). The PWM output mask function is useful when controlling various types of Electrically Commutated Motor (ECM) like a BLDC motor. Figure 6.17-29 shows an example of PWM output mask control in PWMx_CH0 and PWMx_CH1.

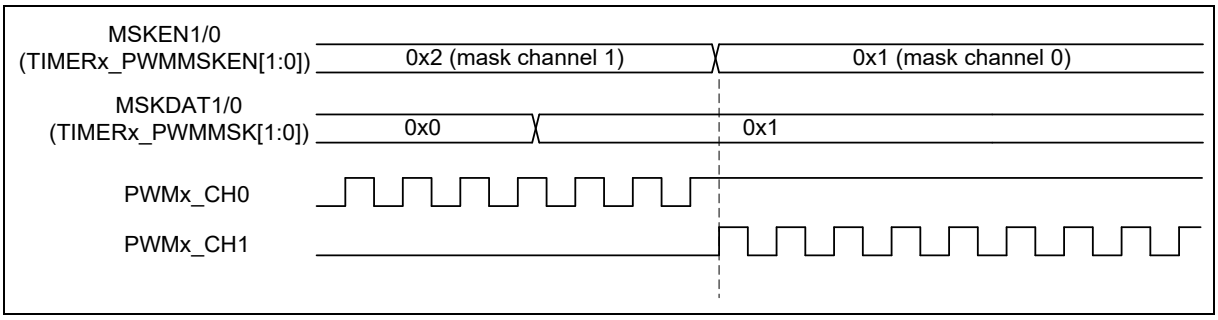


Figure 6.17-29 PWM Output Mask Control Waveform

6.17.6.17 PWM Brake Control

Each PWM generator supports one external input brake pin as PWM brake event source. User can select active brake pin source in BKPINSRC (TIMERx_PWMBNF[18:16]), EPWMx_BRAKEy (x=0,1,2 and y=0,1). There is a 3-bit noise filter counter to filter the external brake pin signal. User can enable BRKNFEN (TIMERx_PWMBNF[0]) to enable the brake pin noise filter function and the noise filter sampling clock can be selected by setting BRKNFSEL (TIMERx_PWMBNF[3:1]) to fit different noise properties. Moreover, by setting BRKFCNT (TIMERx_PWMBNF[6:4]), user can define by how many

sampling clock cycles a filter will recognize the effective edge of the brake pin signal. In addition, brake pin polar can be inversed by setting BRKPINV (TIMERx_PWMBNF[7]) to realize the polarity setup for the brake control signals. Set BRKPINV to 0, brake event will occurred when EPWMx_BRAKEy pin status from low to high; set BRKPINV to 1, brake event will occurred when EPWMx_BRAKEy pin status from high to low.

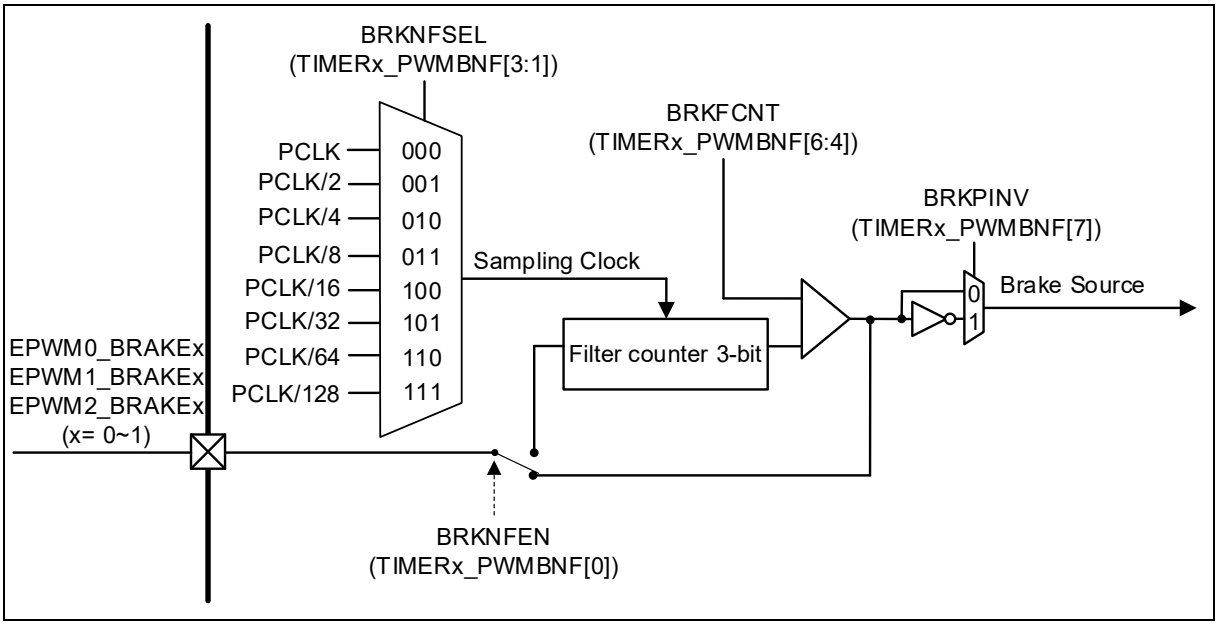


Figure 6.17-30 Brake Pin Noise Filter Block Diagram

User can set BRKAEVEN (TIMERx_PWMBRKCTL[17:16]) for PWMx_CH0 output state and BRKAODD (TIMERx_PWMBRKCTL[19:18]) for PWMx_CH1 output state when PWM brake event happened. There are two brake detector sources, edge detect brake source and level detect brake source when brake event happened. Figure 6.17-31 shows the brake event block diagram for PWMx_CH0 and PWMx_CH1.

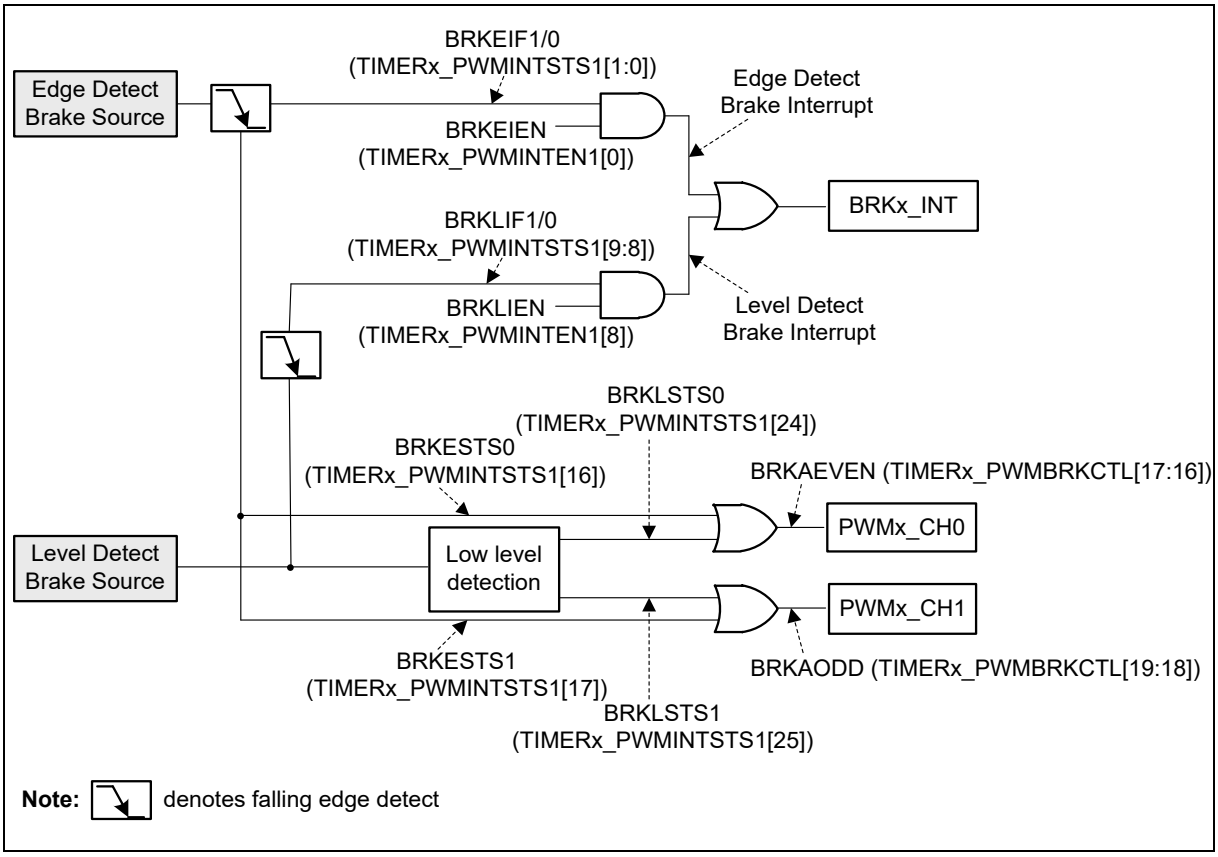


Figure 6.17-31 Brake Event Block Diagram for PWMx_CH0 and PWMx_CH1

When the edge detector detects the brake signal, the brake function generates interrupt status for PWMx_CH1/0 is BRKEIF1/0 (TIMERx_PWMINTSTS1[1:0]) and brake event status for PWMx_CH1/0 is BRKESTS1/0 (TIMERx_PWMINTSTS1[17:16]). The interrupt status BRKEIF1/0 can be cleared by writing 1 to it, and the brake event status BRKESTS1/0 will keep until the next PWM period starts when corresponding BRKEIF1/0 flag has been cleared and PWM generator can resume normal output.

Figure 6.17-32 shows an example of edge detector brake waveform for PWMx_CH0 and PWMx_CH1. In this case, the edge detect brake source has occurred twice for the brake events. When the first brake event occurs, both of the BRKEIF0 and BRKEIF1 flags are set and BRKESTS0 and BRKESTS1 status are also set to indicate brake state of PWMx_CH0 and PWMx_CH1. For the first occurring event, user writes 1 to clear the BRKEIF0. After that, the BRKESTS0 is cleared by hardware at the next start of the PWM period and the PWMx_CH0 outputs the normal waveform even though the edge brake event is still occurring. At the same time, BRKESTS1 keep 1 and PWMx_CH1 keep outputs low in brake state. The second event also triggers the same flags, but at this time, user writes 1 to clear the BRKEIF1. Afterward, PWMx_CH1 outputs normally at the next start of the PWM period.

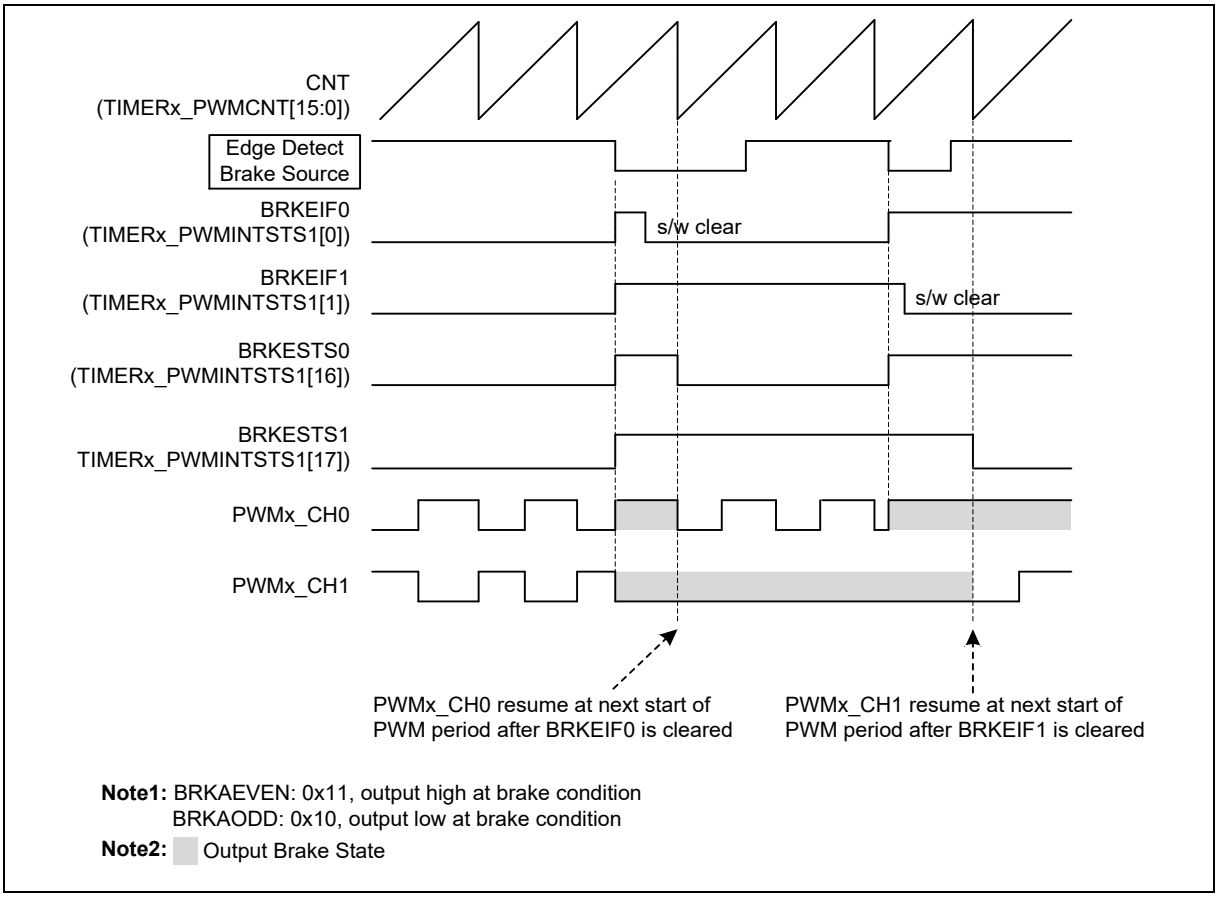


Figure 6.17-32 Edge Detector Brake Waveform for PWMx_CH0 and PWMx_CH1

When the level detector detects the brake signal, the brake function generates interrupt status for PWMx_CH1/0 is BRKLIF1/0 (TIMERx_PWMINTSTS1[9:8]) and brake event status for PWMx_CH1/0 is BRKLSTS1/0 (TIMERx_PWMINTSTS1[25:24]). The interrupt status BRKLIF1/0 can be cleared by writing 1 to it, and the brake event status BRKLSTS1/0 will be cleared only when current period is completed and brake condition removed, then PWM generator can resume normal output when next PWM period starts.

Figure 6.17-33 shows an example of level detector brake waveform for PWMx_CH0 and PWMx_CH1. In this case, the BRKLIF0 and BRKLIF1 can only indicate the brake event has occurred, writes 1 to clear this flags will not affect BRKLSTS0 and BRKLSTS1 brake event status. Both BRKLSTS0 and BRKLSTS1 brake states will automatically cleared at the start of the next PWM period when level brake condition has released no matter BRKLIF0 and BRKLIF1 status.

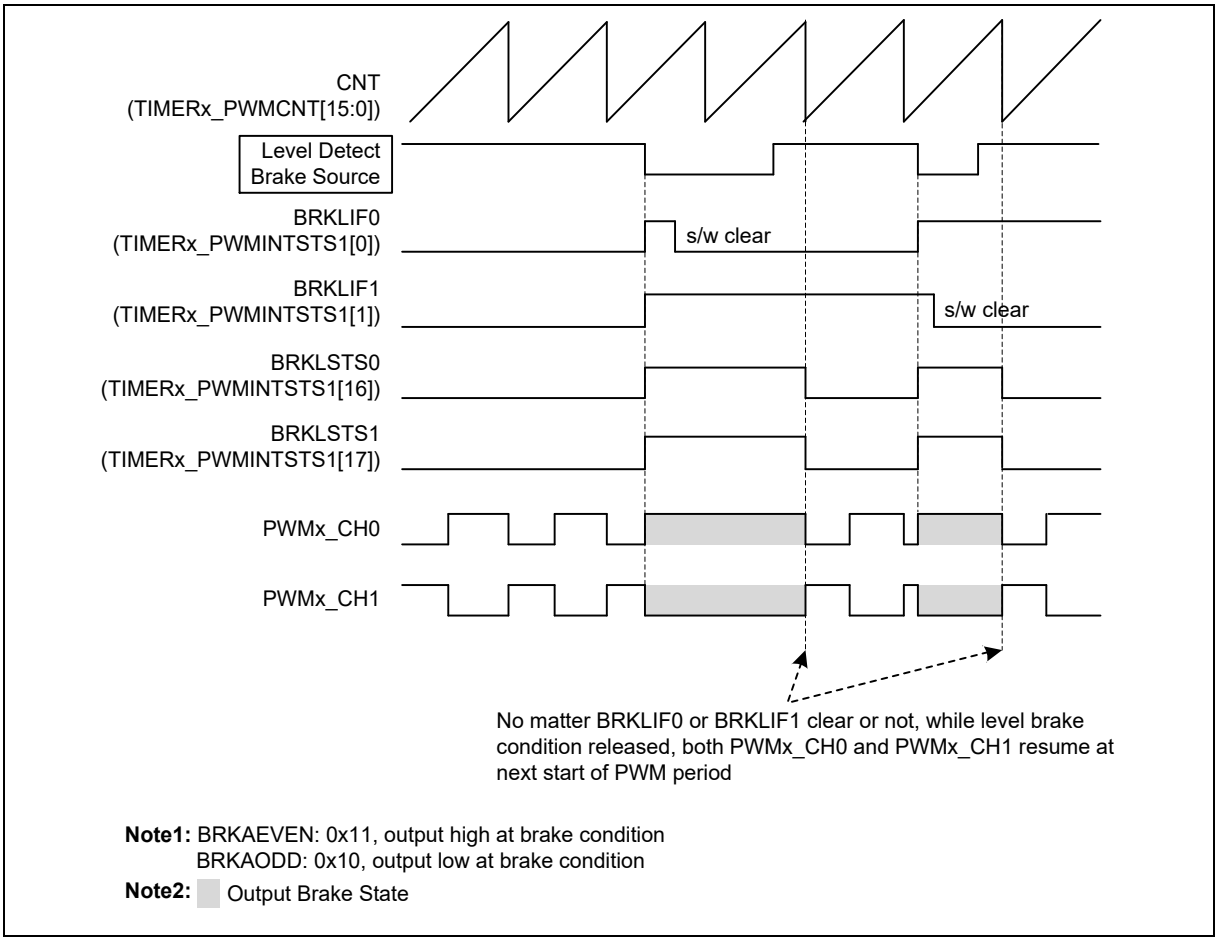


Figure 6.17-33 Level Detector Brake Waveform for PWMx_CH0 and PWMx_CH1

The two kinds of detectors detecting the same five brake sources as shown in Figure 6.17-34 are one from EPWMx_BRAKEy (x=0,1,2 and y=0,1) external input signals, one from system fail events and one from software trigger brake event.

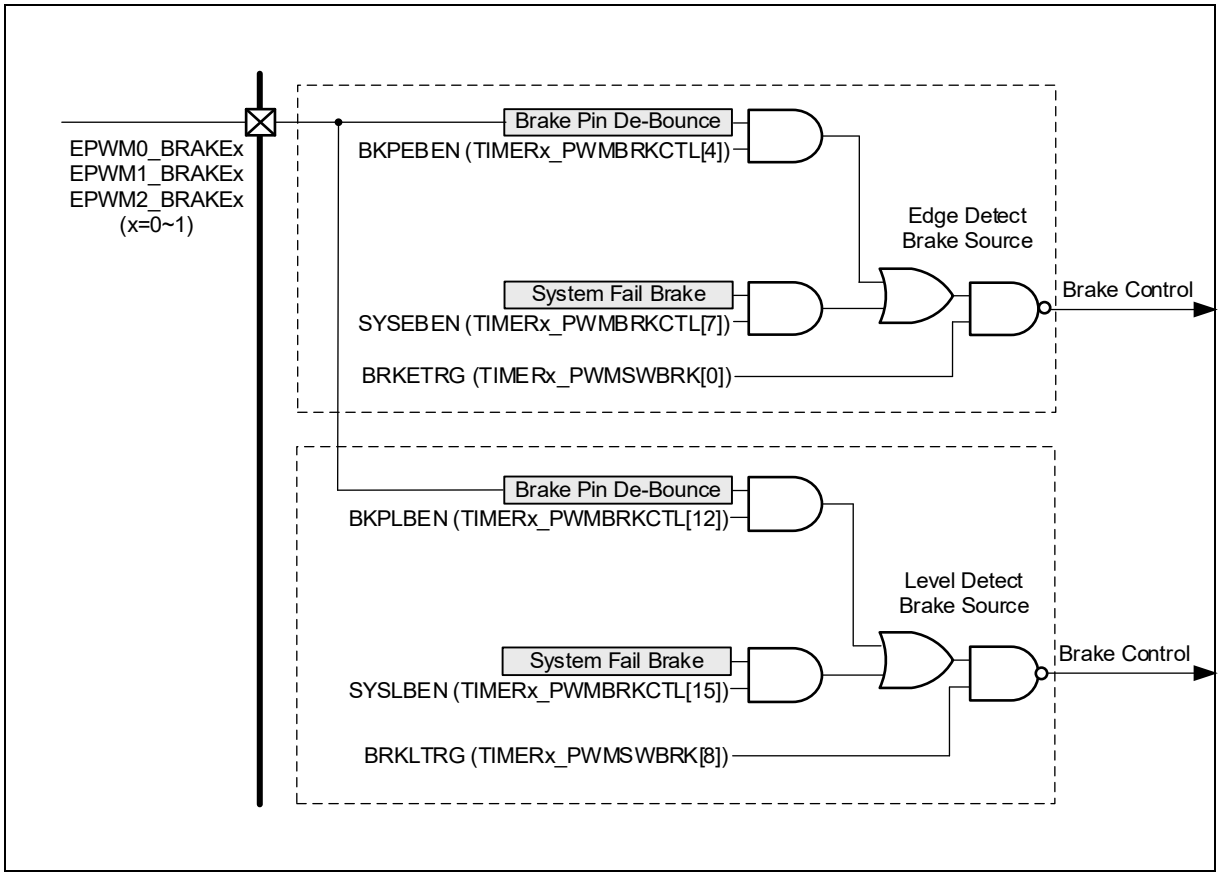


Figure 6.17-34 Brake Source Block Diagram

Among the above described brake sources, the brake source coming from system fail event can be specified to one of the different system fail conditions, these conditions include clock fail, LVDIF detect and CPU lockup as shown in Figure 6.17-35.

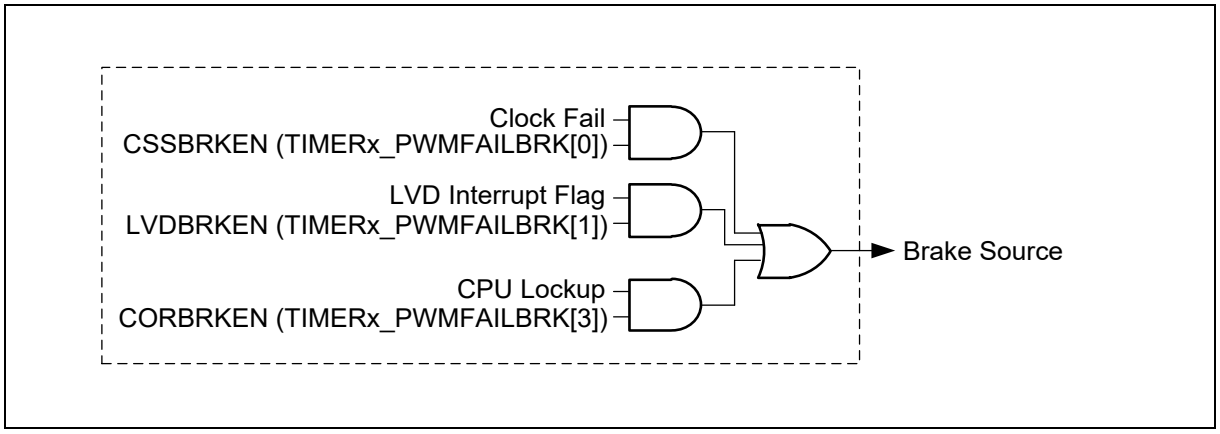
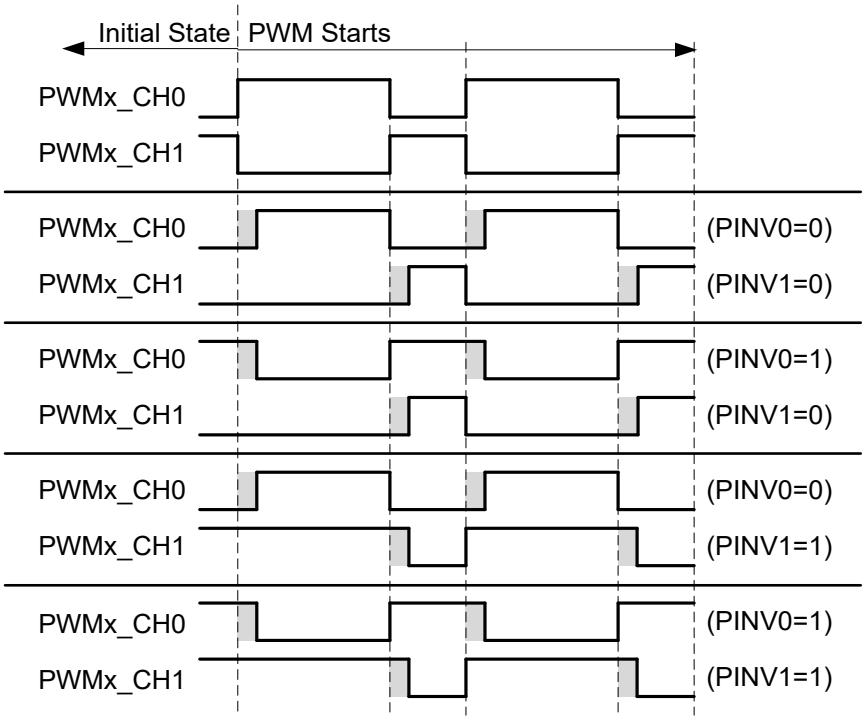


Figure 6.17-35 System Fail Brake Block Diagram

6.17.6.18 Polarity Control

Each PWMx_CH0 and PWMx_CH1 has an independent polarity control to configure the polarity of the active state of PWM output. User can control polarity state of PWMx_CH0 on PINV0 (TIMERx_PWMPOLCTL[0]) and PWMx_CH1 on PINV1 (TIMERx_PWMPOLCTL[1]). Figure 6.17-36 shows the PWMx_CH0 and PWMx_CH1 output with polarity control and dead-time insertion.



Note1: dead-time insertion

Note2: PINV0/PIV1, it controls the output polar inverse

Figure 6.17-36 PWMx_CH0 and PWMx_CH1 Polarity Control with Dead-Time Insertion

6.17.6.19 PWM Interrupt Generator

There are independent interrupts for each PWM as shown in Figure 6.17-37.

The PWM interrupt (PWMx_INT) comes from PWM complementary pair events. The counter can generate the zero point interrupt flag ZIF (TIMERx_PWMINTSTS0[0]) and the period point interrupt flag PIF (TIMERx_PWMINTSTS0[1]). When counter equals to the comparator value stored in CMP (TIMERx_PWMCMPDAT[15:0]), the different interrupt flags will be triggered depending on the counting direction. If counter and CMP matched occurs at up-count direction, the comparator up interrupt flag CMPUIF (TIMERx_PWMINTSTS0[2]) is set and if matched at down-count direction, the comparator down interrupt flag CMPDIF (TIMERx_PWMINTSTS0[3]) is set. If the corresponding interrupt enable bits are set, the interrupt trigger events will also generates interrupt signals. When PWM brake event occurred, the relatives interrupt event will be triggered according to PWM brake settings.

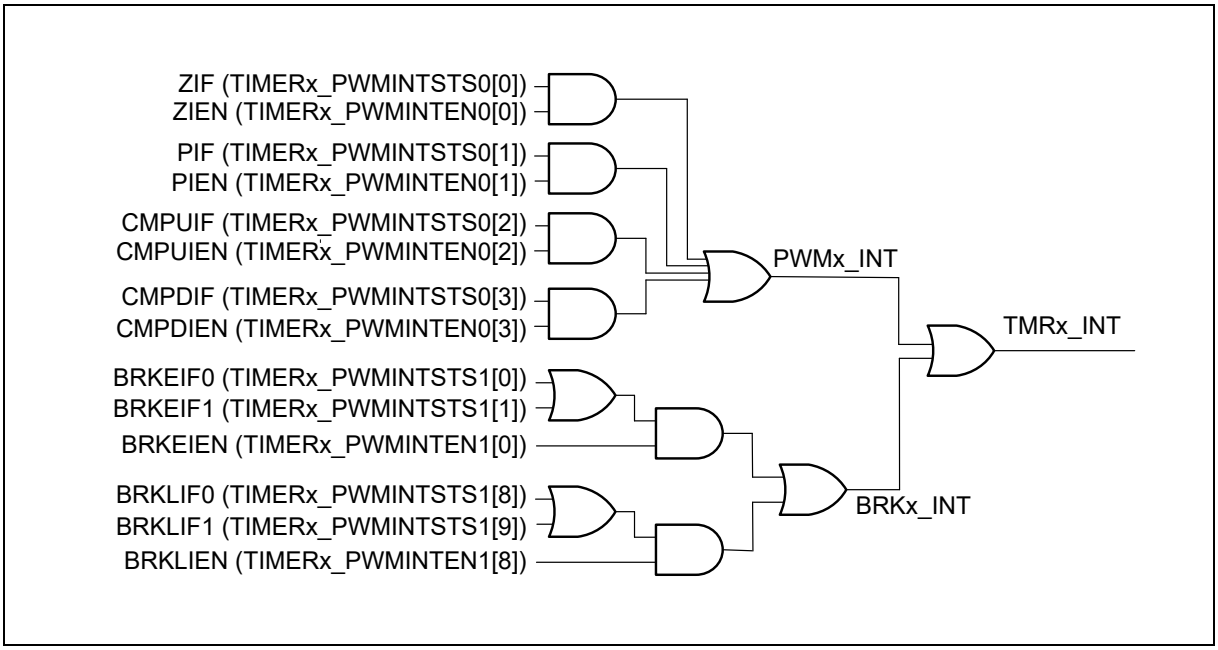


Figure 6.17-37 PWM Interrupt Architecture Diagram

6.17.6.20 PWM Trigger EADC Generator

PWM counter event can be one of the EADC conversion trigger source. User sets TRGSEL (TIMERx_PWMEADCTS[3:0]) to select which PWM counter event can trigger EADC conversion after TRGEN (TIMERx_PWMEADCTS [7]) is enabled.

There are five PWM counter events that can be selected as the trigger source to start EADC conversion as shown in Figure 6.17-38.

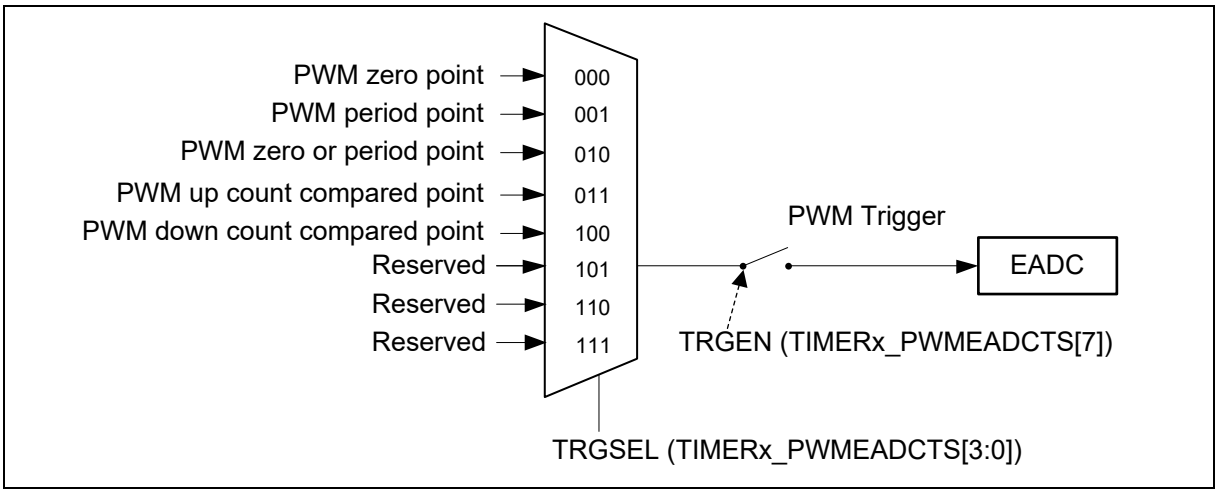


Figure 6.17-38 PWM Trigger EADC Block Diagram

6.17.7 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
TIMER Base Address: **TMR01_BA = 0x4050_0000** **TMR23_BA = 0x4051_0000** **TMR45_BA = 0x4052_0000** **TMR67_BA = 0x4053_0000** **TMR89_BA = 0x4054_0000** **TMR1011_BA = 0x4055_0000**				
TIMER0_CTL	TMR01_BA+0x00	R/W	Timer0 Control Register	0x0000_0005
TIMER0_CMP	TMR01_BA+0x04	R/W	Timer0 Comparator Register	0x0000_0000
TIMER0_INTSTS	TMR01_BA+0x08	R/W	Timer0 Interrupt Status Register	0x0000_0000
TIMER0_CNT	TMR01_BA+0x0C	R/W	Timer0 Data Register	0x0000_0000
TIMER0_CAP	TMR01_BA+0x10	R	Timer0 Capture Data Register	0x0000_0000
TIMER0_EXTCTL	TMR01_BA+0x14	R/W	Timer0 External Control Register	0x0000_0000
TIMER0_EINTSTS	TMR01_BA+0x18	R/W	Timer0 External Interrupt Status Register	0x0000_0000
TIMER0_TRGCTL	TMR01_BA+0x1C	R/W	Timer0 Trigger Control Register	0x0000_0000
TIMER0_ALTCTL	TMR01_BA+0x20	R/W	Timer0 Alternative Control Register	0x0000_0000
TIMER0_PWMCTL	TMR01_BA+0x40	R/W	Timer0 PWM Control Register	0x0000_0000
TIMER0_PWMCLKSRC	TMR01_BA+0x44	R/W	Timer0 PWM Counter Clock Source Register	0x0000_0000
TIMER0_PWMCLKPSC	TMR01_BA+0x48	R/W	Timer0 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER0_PWMCNTCLR	TMR01_BA+0x4C	R/W	Timer0 PWM Clear Counter Register	0x0000_0000
TIMER0_PWMPERIOD	TMR01_BA+0x50	R/W	Timer0 PWM Period Register	0x0000_0000
TIMER0_PWMCMPDAT	TMR01_BA+0x54	R/W	Timer0 PWM Comparator Register	0x0000_0000
TIMER0_PWMDTCTL	TMR01_BA+0x58	R/W	Timer0 PWM Dead-time Control Register	0x0000_0000
TIMER0_PWMCNT	TMR01_BA+0x5C	R	Timer0 PWM Counter Register	0x0000_0000
TIMER0_PWMMSKEN	TMR01_BA+0x60	R/W	Timer0 PWM Output Mask Enable Register	0x0000_0000
TIMER0_PWMMSK	TMR01_BA+0x64	R/W	Timer0 PWM Output Mask Data Control Register	0x0000_0000
TIMER0_PWMBNF	TMR01_BA+0x68	R/W	Timer0 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER0_PWMFAILBRK	TMR01_BA+0x6C	R/W	Timer0 PWM System Fail Brake Control Register	0x0000_0000
TIMER0_PWMBRKCTL	TMR01_BA+0x70	R/W	Timer0 PWM Brake Control Register	0x0000_0000
TIMER0_PWMPOLCTL	TMR01_BA+0x74	R/W	Timer0 PWM Pin Output Polar Control Register	0x0000_0000

TIMER0_PWMPOEN	TMR01_BA+0x78	R/W	Timer0 PWM Pin Output Enable Register	0x0000_0000
TIMER0_PWMSWBRK	TMR01_BA+0x7C	W	Timer0 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER0_PWMINTEN0	TMR01_BA+0x80	R/W	Timer0 PWM Interrupt Enable Register 0	0x0000_0000
TIMER0_PWMINTEN1	TMR01_BA+0x84	R/W	Timer0 PWM Interrupt Enable Register 1	0x0000_0000
TIMER0_PWMINTSTS0	TMR01_BA+0x88	R/W	Timer0 PWM Interrupt Status Register 0	0x0000_0000
TIMER0_PWMINTSTS1	TMR01_BA+0x8C	R/W	Timer0 PWM Interrupt Status Register 1	0x0000_0000
TIMER0_PWMEADCTS	TMR01_BA+0x90	R/W	Timer0 PWM EADC Trigger Source Select Register	0x0000_0000
TIMER0_PWMSCTL	TMR01_BA+0x94	R/W	Timer0 PWM Synchronous Control Register	0x0000_0000
TIMER0_PWMSTRG	TMR01_BA+0x98	W	Timer0 PWM Synchronous Trigger Register	0x0000_0000
TIMER0_PWMSTATUS	TMR01_BA+0x9C	R/W	Timer0 PWM Status Register	0x0000_0000
TIMER0_PWMPBUF	TMR01_BA+0xA0	R	Timer0 PWM Period Buffer Register	0x0000_0000
TIMER0_PWMCMPBUF	TMR01_BA+0xA4	R	Timer0 PWM Comparator Buffer Register	0x0000_0000
TIMER1_CTL	TMR01_BA+0x100	R/W	Timer1 Control Register	0x0000_0005
TIMER1_CMP	TMR01_BA+0x104	R/W	Timer1 Comparator Register	0x0000_0000
TIMER1_INTSTS	TMR01_BA+0x108	R/W	Timer1 Interrupt Status Register	0x0000_0000
TIMER1_CNT	TMR01_BA+0x10C	R/W	Timer1 Data Register	0x0000_0000
TIMER1_CAP	TMR01_BA+0x110	R	Timer1 Capture Data Register	0x0000_0000
TIMER1_EXTCTL	TMR01_BA+0x114	R/W	Timer1 External Control Register	0x0000_0000
TIMER1_EINTSTS	TMR01_BA+0x118	R/W	Timer1 External Interrupt Status Register	0x0000_0000
TIMER1_TRGCTL	TMR01_BA+0x11C	R/W	Timer1 Trigger Control Register	0x0000_0000
TIMER1_ALTCTL	TMR01_BA+0x120	R/W	Timer1 Alternative Control Register	0x0000_0000
TIMER1_PWMCTL	TMR01_BA+0x140	R/W	Timer1 PWM Control Register	0x0000_0000
TIMER1_PWMCLKSRC	TMR01_BA+0x144	R/W	Timer1 PWM Counter Clock Source Register	0x0000_0000
TIMER1_PWMCLKPSC	TMR01_BA+0x148	R/W	Timer1 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER1_PWMCNTCLR	TMR01_BA+0x14C	R/W	Timer1 PWM Clear Counter Register	0x0000_0000
TIMER1_PWMPERIOD	TMR01_BA+0x150	R/W	Timer1 PWM Period Register	0x0000_0000
TIMER1_PWMCMPDAT	TMR01_BA+0x154	R/W	Timer1 PWM Comparator Register	0x0000_0000
TIMER1_PWMDTCTL	TMR01_BA+0x158	R/W	Timer1 PWM Dead-time Control Register	0x0000_0000
TIMER1_PWMCNT	TMR01_BA+0x15C	R	Timer1 PWM Counter Register	0x0000_0000
TIMER1_PWMMSKEN	TMR01_BA+0x160	R/W	Timer1 PWM Output Mask Enable Register	0x0000_0000
TIMER1_PWMMSK	TMR01_BA+0x164	R/W	Timer1 PWM Output Mask Data Control Register	0x0000_0000

TIMER1_PWMBNF	TMR01_BA+0x168	R/W	Timer1 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER1_PWMFAILBRK	TMR01_BA+0x16C	R/W	Timer1 PWM System Fail Brake Control Register	0x0000_0000
TIMER1_PWMBRKCTL	TMR01_BA+0x170	R/W	Timer1 PWM Brake Control Register	0x0000_0000
TIMER1_PWMPOLCTL	TMR01_BA+0x174	R/W	Timer1 PWM Pin Output Polar Control Register	0x0000_0000
TIMER1_PWMPOEN	TMR01_BA+0x178	R/W	Timer1 PWM Pin Output Enable Register	0x0000_0000
TIMER1_PWMSWBRK	TMR01_BA+0x17C	W	Timer1 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER1_PWMINTEN0	TMR01_BA+0x180	R/W	Timer1 PWM Interrupt Enable Register 0	0x0000_0000
TIMER1_PWMINTEN1	TMR01_BA+0x184	R/W	Timer1 PWM Interrupt Enable Register 1	0x0000_0000
TIMER1_PWMINTSTS0	TMR01_BA+0x188	R/W	Timer1 PWM Interrupt Status Register 0	0x0000_0000
TIMER1_PWMINTSTS1	TMR01_BA+0x18C	R/W	Timer1 PWM Interrupt Status Register 1	0x0000_0000
TIMER1_PWMEADCTS	TMR01_BA+0x190	R/W	Timer1 PWM EADC Trigger Source Select Register	0x0000_0000
TIMER1_PWMSCTL	TMR01_BA+0x194	R/W	Timer1 PWM Synchronous Control Register	0x0000_0000
TIMER1_PWMSTATUS	TMR01_BA+0x19C	R/W	Timer1 PWM Status Register	0x0000_0000
TIMER1_PWMPBUF	TMR01_BA+0x1A0	R	Timer1 PWM Period Buffer Register	0x0000_0000
TIMER1_PWMCMPBUF	TMR01_BA+0x1A4	R	Timer1 PWM Comparator Buffer Register	0x0000_0000
TIMER2_CTL	TMR23_BA+0x00	R/W	Timer2 Control Register	0x0000_0005
TIMER2_CMP	TMR23_BA+0x04	R/W	Timer2 Comparator Register	0x0000_0000
TIMER2_INTSTS	TMR23_BA+0x08	R/W	Timer2 Interrupt Status Register	0x0000_0000
TIMER2_CNT	TMR23_BA+0x0C	R/W	Timer2 Data Register	0x0000_0000
TIMER2_CAP	TMR23_BA+0x10	R	Timer2 Capture Data Register	0x0000_0000
TIMER2_EXTCTL	TMR23_BA+0x14	R/W	Timer2 External Control Register	0x0000_0000
TIMER2_EINTSTS	TMR23_BA+0x18	R/W	Timer2 External Interrupt Status Register	0x0000_0000
TIMER2_TRGCTL	TMR23_BA+0x1C	R/W	Timer2 Trigger Control Register	0x0000_0000
TIMER2_ALTCTL	TMR23_BA+0x20	R/W	Timer2 Alternative Control Register	0x0000_0000
TIMER2_PWMCTL	TMR23_BA+0x40	R/W	Timer2 PWM Control Register	0x0000_0000
TIMER2_PWMCLKSRC	TMR23_BA+0x44	R/W	Timer2 PWM Counter Clock Source Register	0x0000_0000
TIMER2_PWMCLKPSC	TMR23_BA+0x48	R/W	Timer2 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER2_PWMCNTCLR	TMR23_BA+0x4C	R/W	Timer2 PWM Clear Counter Register	0x0000_0000
TIMER2_PWMPERIOD	TMR23_BA+0x50	R/W	Timer2 PWM Period Register	0x0000_0000
TIMER2_PWMCMPDAT	TMR23_BA+0x54	R/W	Timer2 PWM Comparator Register	0x0000_0000
TIMER2_PWMDTCTL	TMR23_BA+0x58	R/W	Timer2 PWM Dead-time Control Register	0x0000_0000

TIMER2_PWMCNT	TMR23_BA+0x5C	R	Timer2 PWM Counter Register	0x0000_0000
TIMER2_PWMMSKEN	TMR23_BA+0x60	R/W	Timer2 PWM Output Mask Enable Register	0x0000_0000
TIMER2_PWMMSK	TMR23_BA+0x64	R/W	Timer2 PWM Output Mask Data Control Register	0x0000_0000
TIMER2_PWMBNF	TMR23_BA+0x68	R/W	Timer2 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER2_PWMFAILBRK	TMR23_BA+0x6C	R/W	Timer2 PWM System Fail Brake Control Register	0x0000_0000
TIMER2_PWMBRKCTL	TMR23_BA+0x70	R/W	Timer2 PWM Brake Control Register	0x0000_0000
TIMER2_PWMPOLCTL	TMR23_BA+0x74	R/W	Timer2 PWM Pin Output Polar Control Register	0x0000_0000
TIMER2_PWMPOEN	TMR23_BA+0x78	R/W	Timer2 PWM Pin Output Enable Register	0x0000_0000
TIMER2_PWMSWBRK	TMR23_BA+0x7C	W	Timer2 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER2_PWMINTEN0	TMR23_BA+0x80	R/W	Timer2 PWM Interrupt Enable Register 0	0x0000_0000
TIMER2_PWMINTEN1	TMR23_BA+0x84	R/W	Timer2 PWM Interrupt Enable Register 1	0x0000_0000
TIMER2_PWMINTSTS0	TMR23_BA+0x88	R/W	Timer2 PWM Interrupt Status Register 0	0x0000_0000
TIMER2_PWMINTSTS1	TMR23_BA+0x8C	R/W	Timer2 PWM Interrupt Status Register 1	0x0000_0000
TIMER2_PWMEADCTS	TMR23_BA+0x90	R/W	Timer2 PWM EADC Trigger Source Select Register	0x0000_0000
TIMER2_PWMSCTL	TMR23_BA+0x94	R/W	Timer2 PWM Synchronous Control Register	0x0000_0000
TIMER2_PWMSTRG	TMR23_BA+0x98	W	Timer2 PWM Synchronous Trigger Register	0x0000_0000
TIMER2_PWMSTATUS	TMR23_BA+0x9C	R/W	Timer2 PWM Status Register	0x0000_0000
TIMER2_PWMPBUF	TMR23_BA+0xA0	R	Timer2 PWM Period Buffer Register	0x0000_0000
TIMER2_PWMCMPBUF	TMR23_BA+0xA4	R	Timer2 PWM Comparator Buffer Register	0x0000_0000
TIMER3_CTL	TMR23_BA+0x100	R/W	Timer3 Control Register	0x0000_0005
TIMER3_CMP	TMR23_BA+0x104	R/W	Timer3 Comparator Register	0x0000_0000
TIMER3_INTSTS	TMR23_BA+0x108	R/W	Timer3 Interrupt Status Register	0x0000_0000
TIMER3_CNT	TMR23_BA+0x10C	R/W	Timer3 Data Register	0x0000_0000
TIMER3_CAP	TMR23_BA+0x110	R	Timer3 Capture Data Register	0x0000_0000
TIMER3_EXTCTL	TMR23_BA+0x114	R/W	Timer3 External Control Register	0x0000_0000
TIMER3_EINTSTS	TMR23_BA+0x118	R/W	Timer3 External Interrupt Status Register	0x0000_0000
TIMER3_TRGCTL	TMR23_BA+0x11C	R/W	Timer3 Trigger Control Register	0x0000_0000
TIMER3_ALTCTL	TMR23_BA+0x120	R/W	Timer3 Alternative Control Register	0x0000_0000
TIMER3_PWMCTL	TMR23_BA+0x140	R/W	Timer3 PWM Control Register	0x0000_0000
TIMER3_PWMCLKSRC	TMR23_BA+0x144	R/W	Timer3 PWM Counter Clock Source Register	0x0000_0000
TIMER3_PWMCLKPSC	TMR23_BA+0x148	R/W	Timer3 PWM Counter Clock Pre-scale Register	0x0000_0000

TIMER3_PWMCNTCLR	TMR23_BA+0x14C	R/W	Timer3 PWM Clear Counter Register	0x0000_0000
TIMER3_PWMPERIOD	TMR23_BA+0x150	R/W	Timer3 PWM Period Register	0x0000_0000
TIMER3_PWMCMPDAT	TMR23_BA+0x154	R/W	Timer3 PWM Comparator Register	0x0000_0000
TIMER3_PWMDTCTL	TMR23_BA+0x158	R/W	Timer3 PWM Dead-time Control Register	0x0000_0000
TIMER3_PWMCNT	TMR23_BA+0x15C	R	Timer3 PWM Counter Register	0x0000_0000
TIMER3_PWMMSKEN	TMR23_BA+0x160	R/W	Timer3 PWM Output Mask Enable Register	0x0000_0000
TIMER3_PWMMSK	TMR23_BA+0x164	R/W	Timer3 PWM Output Mask Data Control Register	0x0000_0000
TIMER3_PWMBNF	TMR23_BA+0x168	R/W	Timer3 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER3_PWMFAILBRK	TMR23_BA+0x16C	R/W	Timer3 PWM System Fail Brake Control Register	0x0000_0000
TIMER3_PWMBRKCTL	TMR23_BA+0x170	R/W	Timer3 PWM Brake Control Register	0x0000_0000
TIMER3_PWMPOLCTL	TMR23_BA+0x174	R/W	Timer3 PWM Pin Output Polar Control Register	0x0000_0000
TIMER3_PWMPOEN	TMR23_BA+0x178	R/W	Timer3 PWM Pin Output Enable Register	0x0000_0000
TIMER3_PWMSWBRK	TMR23_BA+0x17C	W	Timer3 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER3_PWMINTEN0	TMR23_BA+0x180	R/W	Timer3 PWM Interrupt Enable Register 0	0x0000_0000
TIMER3_PWMINTEN1	TMR23_BA+0x184	R/W	Timer3 PWM Interrupt Enable Register 1	0x0000_0000
TIMER3_PWMINTSTS0	TMR23_BA+0x188	R/W	Timer3 PWM Interrupt Status Register 0	0x0000_0000
TIMER3_PWMINTSTS1	TMR23_BA+0x18C	R/W	Timer3 PWM Interrupt Status Register 1	0x0000_0000
TIMER3_PWMEADCTS	TMR23_BA+0x190	R/W	Timer3 PWM EADC Trigger Source Select Register	0x0000_0000
TIMER3_PWMSCTL	TMR23_BA+0x194	R/W	Timer3 PWM Synchronous Control Register	0x0000_0000
TIMER3_PWMSTATUS	TMR23_BA+0x19C	R/W	Timer3 PWM Status Register	0x0000_0000
TIMER3_PWMPBUF	TMR23_BA+0x1A0	R	Timer3 PWM Period Buffer Register	0x0000_0000
TIMER3_PWMCMPBUF	TMR23_BA+0x1A4	R	Timer3 PWM Comparator Buffer Register	0x0000_0000
TIMER4_CTL	TMR45_BA+0x00	R/W	Timer4 Control Register	0x0000_0005
TIMER4_CMP	TMR45_BA+0x04	R/W	Timer4 Comparator Register	0x0000_0000
TIMER4_INTSTS	TMR45_BA+0x08	R/W	Timer4 Interrupt Status Register	0x0000_0000
TIMER4_CNT	TMR45_BA+0x0C	R/W	Timer4 Data Register	0x0000_0000
TIMER4_CAP	TMR45_BA+0x10	R	Timer4 Capture Data Register	0x0000_0000
TIMER4_EXTCTL	TMR45_BA+0x14	R/W	Timer4 External Control Register	0x0000_0000
TIMER4_EINTSTS	TMR45_BA+0x18	R/W	Timer4 External Interrupt Status Register	0x0000_0000
TIMER4_TRGCTL	TMR45_BA+0x1C	R/W	Timer4 Trigger Control Register	0x0000_0000
TIMER4_ALTCTL	TMR45_BA+0x20	R/W	Timer4 Alternative Control Register	0x0000_0000

TIMER4_PWMCTL	TMR45_BA+0x40	R/W	Timer4 PWM Control Register	0x0000_0000
TIMER4_PWMCLKSRC	TMR45_BA+0x44	R/W	Timer4 PWM Counter Clock Source Register	0x0000_0000
TIMER4_PWMCLKPSC	TMR45_BA+0x48	R/W	Timer4 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER4_PWMCNTCLR	TMR45_BA+0x4C	R/W	Timer4 PWM Clear Counter Register	0x0000_0000
TIMER4_PWMPERIOD	TMR45_BA+0x50	R/W	Timer4 PWM Period Register	0x0000_0000
TIMER4_PWMCMPDAT	TMR45_BA+0x54	R/W	Timer4 PWM Comparator Register	0x0000_0000
TIMER4_PWMDTCTL	TMR45_BA+0x58	R/W	Timer4 PWM Dead-time Control Register	0x0000_0000
TIMER4_PWMCNT	TMR45_BA+0x5C	R	Timer4 PWM Counter Register	0x0000_0000
TIMER4_PWMMSKEN	TMR45_BA+0x60	R/W	Timer4 PWM Output Mask Enable Register	0x0000_0000
TIMER4_PWMMSK	TMR45_BA+0x64	R/W	Timer4 PWM Output Mask Data Control Register	0x0000_0000
TIMER4_PWMBNF	TMR45_BA+0x68	R/W	Timer4 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER4_PWMFAILBRK	TMR45_BA+0x6C	R/W	Timer4 PWM System Fail Brake Control Register	0x0000_0000
TIMER4_PWMBRKCTL	TMR45_BA+0x70	R/W	Timer4 PWM Brake Control Register	0x0000_0000
TIMER4_PWMPOLCTL	TMR45_BA+0x74	R/W	Timer4 PWM Pin Output Polar Control Register	0x0000_0000
TIMER4_PWMPOEN	TMR45_BA+0x78	R/W	Timer4 PWM Pin Output Enable Register	0x0000_0000
TIMER4_PWMSWBRK	TMR45_BA+0x7C	W	Timer4 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER4_PWMINTEN0	TMR45_BA+0x80	R/W	Timer4 PWM Interrupt Enable Register 0	0x0000_0000
TIMER4_PWMINTEN1	TMR45_BA+0x84	R/W	Timer4 PWM Interrupt Enable Register 1	0x0000_0000
TIMER4_PWMINTSTS0	TMR45_BA+0x88	R/W	Timer4 PWM Interrupt Status Register 0	0x0000_0000
TIMER4_PWMINTSTS1	TMR45_BA+0x8C	R/W	Timer4 PWM Interrupt Status Register 1	0x0000_0000
TIMER4_PWMEADCTS	TMR45_BA+0x90	R/W	Timer4 PWM EADC Trigger Source Select Register	0x0000_0000
TIMER4_PWMSCTL	TMR45_BA+0x94	R/W	Timer4 PWM Synchronous Control Register	0x0000_0000
TIMER4_PWMSTRG	TMR45_BA+0x98	W	Timer4 PWM Synchronous Trigger Register	0x0000_0000
TIMER4_PWMSTATUS	TMR45_BA+0x9C	R/W	Timer4 PWM Status Register	0x0000_0000
TIMER4_PWMPBUF	TMR45_BA+0xA0	R	Timer4 PWM Period Buffer Register	0x0000_0000
TIMER4_PWMCMPBUF	TMR45_BA+0xA4	R	Timer4 PWM Comparator Buffer Register	0x0000_0000
TIMER5_CTL	TMR45_BA+0x100	R/W	Timer5 Control Register	0x0000_0005
TIMER5_CMP	TMR45_BA+0x104	R/W	Timer5 Comparator Register	0x0000_0000
TIMER5_INTSTS	TMR45_BA+0x108	R/W	Timer5 Interrupt Status Register	0x0000_0000
TIMER5_CNT	TMR45_BA+0x10C	R/W	Timer5 Data Register	0x0000_0000
TIMER5_CAP	TMR45_BA+0x110	R	Timer5 Capture Data Register	0x0000_0000

TIMER5_EXTCTL	TMR45_BA+0x114	R/W	Timer5 External Control Register	0x0000_0000
TIMER5_EINTSTS	TMR45_BA+0x118	R/W	Timer5 External Interrupt Status Register	0x0000_0000
TIMER5_TRGCTL	TMR45_BA+0x11C	R/W	Timer5 Trigger Control Register	0x0000_0000
TIMER5_ALTCTL	TMR45_BA+0x120	R/W	Timer5 Alternative Control Register	0x0000_0000
TIMER5_PWMCTL	TMR45_BA+0x140	R/W	Timer5 PWM Control Register	0x0000_0000
TIMER5_PWMCLKSRC	TMR45_BA+0x144	R/W	Timer5 PWM Counter Clock Source Register	0x0000_0000
TIMER5_PWMCLKPSC	TMR45_BA+0x148	R/W	Timer5 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER5_PWMCNTCLR	TMR45_BA+0x14C	R/W	Timer5 PWM Clear Counter Register	0x0000_0000
TIMER5_PWMPERIOD	TMR45_BA+0x150	R/W	Timer5 PWM Period Register	0x0000_0000
TIMER5_PWMCMPDAT	TMR45_BA+0x154	R/W	Timer5 PWM Comparator Register	0x0000_0000
TIMER5_PWMDTCTL	TMR45_BA+0x158	R/W	Timer5 PWM Dead-time Control Register	0x0000_0000
TIMER5_PWMCNT	TMR45_BA+0x15C	R	Timer5 PWM Counter Register	0x0000_0000
TIMER5_PWMMSKEN	TMR45_BA+0x160	R/W	Timer5 PWM Output Mask Enable Register	0x0000_0000
TIMER5_PWMMSK	TMR45_BA+0x164	R/W	Timer5 PWM Output Mask Data Control Register	0x0000_0000
TIMER5_PWMBNF	TMR45_BA+0x168	R/W	Timer5 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER5_PWMFAILBRK	TMR45_BA+0x16C	R/W	Timer5 PWM System Fail Brake Control Register	0x0000_0000
TIMER5_PWMBRKCTL	TMR45_BA+0x170	R/W	Timer5 PWM Brake Control Register	0x0000_0000
TIMER5_PWMPOLCTL	TMR45_BA+0x174	R/W	Timer5 PWM Pin Output Polar Control Register	0x0000_0000
TIMER5_PWMPOEN	TMR45_BA+0x178	R/W	Timer5 PWM Pin Output Enable Register	0x0000_0000
TIMER5_PWMSWBRK	TMR45_BA+0x17C	W	Timer5 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER5_PWMINTEN0	TMR45_BA+0x180	R/W	Timer5 PWM Interrupt Enable Register 0	0x0000_0000
TIMER5_PWMINTEN1	TMR45_BA+0x184	R/W	Timer5 PWM Interrupt Enable Register 1	0x0000_0000
TIMER5_PWMINTSTS0	TMR45_BA+0x188	R/W	Timer5 PWM Interrupt Status Register 0	0x0000_0000
TIMER5_PWMINTSTS1	TMR45_BA+0x18C	R/W	Timer5 PWM Interrupt Status Register 1	0x0000_0000
TIMER5_PWMEADCTS	TMR45_BA+0x190	R/W	Timer5 PWM EADC Trigger Source Select Register	0x0000_0000
TIMER5_PWMSCTL	TMR45_BA+0x194	R/W	Timer5 PWM Synchronous Control Register	0x0000_0000
TIMER5_PWMSTATUS	TMR45_BA+0x19C	R/W	Timer5 PWM Status Register	0x0000_0000
TIMER5_PWMPBUF	TMR45_BA+0x1A0	R	Timer5 PWM Period Buffer Register	0x0000_0000
TIMER5_PWMCMPBUF	TMR45_BA+0x1A4	R	Timer5 PWM Comparator Buffer Register	0x0000_0000
TIMER6_CTL	TMR67_BA+0x00	R/W	Timer6 Control Register	0x0000_0005
TIMER6_CMP	TMR67_BA+0x04	R/W	Timer6 Comparator Register	0x0000_0000

TIMER6_INTSTS	TMR67_BA+0x08	R/W	Timer6 Interrupt Status Register	0x0000_0000
TIMER6_CNT	TMR67_BA+0x0C	R/W	Timer6 Data Register	0x0000_0000
TIMER6_CAP	TMR67_BA+0x10	R	Timer6 Capture Data Register	0x0000_0000
TIMER6_EXTCTL	TMR67_BA+0x14	R/W	Timer6 External Control Register	0x0000_0000
TIMER6_EINTSTS	TMR67_BA+0x18	R/W	Timer6 External Interrupt Status Register	0x0000_0000
TIMER6_TRGCTL	TMR67_BA+0x1C	R/W	Timer6 Trigger Control Register	0x0000_0000
TIMER6_ALTCTL	TMR67_BA+0x20	R/W	Timer6 Alternative Control Register	0x0000_0000
TIMER6_PWMCTL	TMR67_BA+0x40	R/W	Timer6 PWM Control Register	0x0000_0000
TIMER6_PWMCLKSRC	TMR67_BA+0x44	R/W	Timer6 PWM Counter Clock Source Register	0x0000_0000
TIMER6_PWMCLKPSC	TMR67_BA+0x48	R/W	Timer6 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER6_PWMCNTCLR	TMR67_BA+0x4C	R/W	Timer6 PWM Clear Counter Register	0x0000_0000
TIMER6_PWMPERIOD	TMR67_BA+0x50	R/W	Timer6 PWM Period Register	0x0000_0000
TIMER6_PWMCMPDAT	TMR67_BA+0x54	R/W	Timer6 PWM Comparator Register	0x0000_0000
TIMER6_PWMDTCTL	TMR67_BA+0x58	R/W	Timer6 PWM Dead-time Control Register	0x0000_0000
TIMER6_PWMCNT	TMR67_BA+0x5C	R	Timer6 PWM Counter Register	0x0000_0000
TIMER6_PWMMSKEN	TMR67_BA+0x60	R/W	Timer6 PWM Output Mask Enable Register	0x0000_0000
TIMER6_PWMMSK	TMR67_BA+0x64	R/W	Timer6 PWM Output Mask Data Control Register	0x0000_0000
TIMER6_PWMBNF	TMR67_BA+0x68	R/W	Timer6 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER6_PWMFAILBRK	TMR67_BA+0x6C	R/W	Timer6 PWM System Fail Brake Control Register	0x0000_0000
TIMER6_PWMBRKCTL	TMR67_BA+0x70	R/W	Timer6 PWM Brake Control Register	0x0000_0000
TIMER6_PWMPOLCTL	TMR67_BA+0x74	R/W	Timer6 PWM Pin Output Polar Control Register	0x0000_0000
TIMER6_PWMPOEN	TMR67_BA+0x78	R/W	Timer6 PWM Pin Output Enable Register	0x0000_0000
TIMER6_PWMSWBRK	TMR67_BA+0x7C	W	Timer6 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER6_PWMINTEN0	TMR67_BA+0x80	R/W	Timer6 PWM Interrupt Enable Register 0	0x0000_0000
TIMER6_PWMINTEN1	TMR67_BA+0x84	R/W	Timer6 PWM Interrupt Enable Register 1	0x0000_0000
TIMER6_PWMINTSTS0	TMR67_BA+0x88	R/W	Timer6 PWM Interrupt Status Register 0	0x0000_0000
TIMER6_PWMINTSTS1	TMR67_BA+0x8C	R/W	Timer6 PWM Interrupt Status Register 1	0x0000_0000
TIMER6_PWMEADCTS	TMR67_BA+0x90	R/W	Timer6 PWM EADC Trigger Source Select Register	0x0000_0000
TIMER6_PWMSCTL	TMR67_BA+0x94	R/W	Timer6 PWM Synchronous Control Register	0x0000_0000
TIMER6_PWMSTRG	TMR67_BA+0x98	W	Timer6 PWM Synchronous Trigger Register	0x0000_0000
TIMER6_PWMSTATUS	TMR67_BA+0x9C	R/W	Timer6 PWM Status Register	0x0000_0000

TIMER6_PWMPBUF	TMR67_BA+0xA0	R	Timer6 PWM Period Buffer Register	0x0000_0000
TIMER6_PWMCMPBUF	TMR67_BA+0xA4	R	Timer6 PWM Comparator Buffer Register	0x0000_0000
TIMER7_CTL	TMR67_BA+0x100	R/W	Timer7 Control Register	0x0000_0005
TIMER7_CMP	TMR67_BA+0x104	R/W	Timer7 Comparator Register	0x0000_0000
TIMER7_INTSTS	TMR67_BA+0x108	R/W	Timer7 Interrupt Status Register	0x0000_0000
TIMER7_CNT	TMR67_BA+0x10C	R/W	Timer7 Data Register	0x0000_0000
TIMER7_CAP	TMR67_BA+0x110	R	Timer7 Capture Data Register	0x0000_0000
TIMER7_EXTCTL	TMR67_BA+0x114	R/W	Timer7 External Control Register	0x0000_0000
TIMER7_EINTSTS	TMR67_BA+0x118	R/W	Timer7 External Interrupt Status Register	0x0000_0000
TIMER7_TRGCTL	TMR67_BA+0x11C	R/W	Timer7 Trigger Control Register	0x0000_0000
TIMER7_ALTCTL	TMR67_BA+0x120	R/W	Timer7 Alternative Control Register	0x0000_0000
TIMER7_PWMCTL	TMR67_BA+0x140	R/W	Timer7 PWM Control Register	0x0000_0000
TIMER7_PWMCLKSRC	TMR67_BA+0x144	R/W	Timer7 PWM Counter Clock Source Register	0x0000_0000
TIMER7_PWMCLKPSC	TMR67_BA+0x148	R/W	Timer7 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER7_PWMCNTCLR	TMR67_BA+0x14C	R/W	Timer7 PWM Clear Counter Register	0x0000_0000
TIMER7_PWMPERIOD	TMR67_BA+0x150	R/W	Timer7 PWM Period Register	0x0000_0000
TIMER7_PWMCMPDAT	TMR67_BA+0x154	R/W	Timer7 PWM Comparator Register	0x0000_0000
TIMER7_PWMDTCTL	TMR67_BA+0x158	R/W	Timer7 PWM Dead-time Control Register	0x0000_0000
TIMER7_PWMCNT	TMR67_BA+0x15C	R	Timer7 PWM Counter Register	0x0000_0000
TIMER7_PWMMSKEN	TMR67_BA+0x160	R/W	Timer7 PWM Output Mask Enable Register	0x0000_0000
TIMER7_PWMMSK	TMR67_BA+0x164	R/W	Timer7 PWM Output Mask Data Control Register	0x0000_0000
TIMER7_PWMBNF	TMR67_BA+0x168	R/W	Timer7 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER7_PWMFAILBRK	TMR67_BA+0x16C	R/W	Timer7 PWM System Fail Brake Control Register	0x0000_0000
TIMER7_PWMBRKCTL	TMR67_BA+0x170	R/W	Timer7 PWM Brake Control Register	0x0000_0000
TIMER7_PWMPOLCTL	TMR67_BA+0x174	R/W	Timer7 PWM Pin Output Polar Control Register	0x0000_0000
TIMER7_PWMPOEN	TMR67_BA+0x178	R/W	Timer7 PWM Pin Output Enable Register	0x0000_0000
TIMER7_PWMSWBRK	TMR67_BA+0x17C	W	Timer7 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER7_PWMINTEN0	TMR67_BA+0x180	R/W	Timer7 PWM Interrupt Enable Register 0	0x0000_0000
TIMER7_PWMINTEN1	TMR67_BA+0x184	R/W	Timer7 PWM Interrupt Enable Register 1	0x0000_0000
TIMER7_PWMINTSTS0	TMR67_BA+0x188	R/W	Timer7 PWM Interrupt Status Register 0	0x0000_0000
TIMER7_PWMINTSTS1	TMR67_BA+0x18C	R/W	Timer7 PWM Interrupt Status Register 1	0x0000_0000

TIMER7_PWMEADCTS	TMR67_BA+0x190	R/W	Timer7 PWM EADC Trigger Source Select Register	0x0000_0000
TIMER7_PWMSCTL	TMR67_BA+0x194	R/W	Timer7 PWM Synchronous Control Register	0x0000_0000
TIMER7_PWMSTATUS	TMR67_BA+0x19C	R/W	Timer7 PWM Status Register	0x0000_0000
TIMER7_PWMPBUF	TMR67_BA+0x1A0	R	Timer7 PWM Period Buffer Register	0x0000_0000
TIMER7_PWMCMPBUF	TMR67_BA+0x1A4	R	Timer7 PWM Comparator Buffer Register	0x0000_0000
TIMER8_CTL	TMR89_BA+0x00	R/W	Timer8 Control Register	0x0000_0005
TIMER8_CMP	TMR89_BA+0x04	R/W	Timer8 Comparator Register	0x0000_0000
TIMER8_INTSTS	TMR89_BA+0x08	R/W	Timer8 Interrupt Status Register	0x0000_0000
TIMER8_CNT	TMR89_BA+0x0C	R/W	Timer8 Data Register	0x0000_0000
TIMER8_CAP	TMR89_BA+0x10	R	Timer8 Capture Data Register	0x0000_0000
TIMER8_EXTCTL	TMR89_BA+0x14	R/W	Timer8 External Control Register	0x0000_0000
TIMER8_EINTSTS	TMR89_BA+0x18	R/W	Timer8 External Interrupt Status Register	0x0000_0000
TIMER8_TRGCTL	TMR89_BA+0x1C	R/W	Timer8 Trigger Control Register	0x0000_0000
TIMER8_ALTCTL	TMR89_BA+0x20	R/W	Timer8 Alternative Control Register	0x0000_0000
TIMER8_PWMCTL	TMR89_BA+0x40	R/W	Timer8 PWM Control Register	0x0000_0000
TIMER8_PWMCLKSRC	TMR89_BA+0x44	R/W	Timer8 PWM Counter Clock Source Register	0x0000_0000
TIMER8_PWMCLKPSC	TMR89_BA+0x48	R/W	Timer8 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER8_PWMCNTCLR	TMR89_BA+0x4C	R/W	Timer8 PWM Clear Counter Register	0x0000_0000
TIMER8_PWMPERIOD	TMR89_BA+0x50	R/W	Timer8 PWM Period Register	0x0000_0000
TIMER8_PWMCMPDAT	TMR89_BA+0x54	R/W	Timer8 PWM Comparator Register	0x0000_0000
TIMER8_PWMDTCTL	TMR89_BA+0x58	R/W	Timer8 PWM Dead-time Control Register	0x0000_0000
TIMER8_PWMCNT	TMR89_BA+0x5C	R	Timer8 PWM Counter Register	0x0000_0000
TIMER8_PWMMSKEN	TMR89_BA+0x60	R/W	Timer8 PWM Output Mask Enable Register	0x0000_0000
TIMER8_PWMMSK	TMR89_BA+0x64	R/W	Timer8 PWM Output Mask Data Control Register	0x0000_0000
TIMER8_PWMBNF	TMR89_BA+0x68	R/W	Timer8 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER8_PWMFAILBRK	TMR89_BA+0x6C	R/W	Timer8 PWM System Fail Brake Control Register	0x0000_0000
TIMER8_PWMBRKCTL	TMR89_BA+0x70	R/W	Timer8 PWM Brake Control Register	0x0000_0000
TIMER8_PWMPOLCTL	TMR89_BA+0x74	R/W	Timer8 PWM Pin Output Polar Control Register	0x0000_0000
TIMER8_PWMPOEN	TMR89_BA+0x78	R/W	Timer8 PWM Pin Output Enable Register	0x0000_0000
TIMER8_PWMSWBRK	TMR89_BA+0x7C	W	Timer8 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER8_PWMINTEN0	TMR89_BA+0x80	R/W	Timer8 PWM Interrupt Enable Register 0	0x0000_0000

TIMER8_PWMINTEN1	TMR89_BA+0x84	R/W	Timer8 PWM Interrupt Enable Register 1	0x0000_0000
TIMER8_PWMINTSTS0	TMR89_BA+0x88	R/W	Timer8 PWM Interrupt Status Register 0	0x0000_0000
TIMER8_PWMINTSTS1	TMR89_BA+0x8C	R/W	Timer8 PWM Interrupt Status Register 1	0x0000_0000
TIMER8_PWMEADCTS	TMR89_BA+0x90	R/W	Timer8 PWM EADC Trigger Source Select Register	0x0000_0000
TIMER8_PWMSCTL	TMR89_BA+0x94	R/W	Timer8 PWM Synchronous Control Register	0x0000_0000
TIMER8_PWMSTRG	TMR89_BA+0x98	W	Timer8 PWM Synchronous Trigger Register	0x0000_0000
TIMER8_PWMSTATUS	TMR89_BA+0x9C	R/W	Timer8 PWM Status Register	0x0000_0000
TIMER8_PWMPBUF	TMR89_BA+0xA0	R	Timer8 PWM Period Buffer Register	0x0000_0000
TIMER8_PWMCMPBUF	TMR89_BA+0xA4	R	Timer8 PWM Comparator Buffer Register	0x0000_0000
TIMER9_CTL	TMR89_BA+0x100	R/W	Timer9 Control Register	0x0000_0005
TIMER9_CMP	TMR89_BA+0x104	R/W	Timer9 Comparator Register	0x0000_0000
TIMER9_INTSTS	TMR89_BA+0x108	R/W	Timer9 Interrupt Status Register	0x0000_0000
TIMER9_CNT	TMR89_BA+0x10C	R/W	Timer9 Data Register	0x0000_0000
TIMER9_CAP	TMR89_BA+0x110	R	Timer9 Capture Data Register	0x0000_0000
TIMER9_EXTCTL	TMR89_BA+0x114	R/W	Timer9 External Control Register	0x0000_0000
TIMER9_EINTSTS	TMR89_BA+0x118	R/W	Timer9 External Interrupt Status Register	0x0000_0000
TIMER9_TRGCTL	TMR89_BA+0x11C	R/W	Timer9 Trigger Control Register	0x0000_0000
TIMER9_ALTCTL	TMR89_BA+0x120	R/W	Timer9 Alternative Control Register	0x0000_0000
TIMER9_PWMCTL	TMR89_BA+0x140	R/W	Timer9 PWM Control Register	0x0000_0000
TIMER9_PWMCLKSRC	TMR89_BA+0x144	R/W	Timer9 PWM Counter Clock Source Register	0x0000_0000
TIMER9_PWMCLKPSC	TMR89_BA+0x148	R/W	Timer9 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER9_PWMCNTCLR	TMR89_BA+0x14C	R/W	Timer9 PWM Clear Counter Register	0x0000_0000
TIMER9_PWMPERIOD	TMR89_BA+0x150	R/W	Timer9 PWM Period Register	0x0000_0000
TIMER9_PWMCMPDAT	TMR89_BA+0x154	R/W	Timer9 PWM Comparator Register	0x0000_0000
TIMER9_PWMDTCTL	TMR89_BA+0x158	R/W	Timer9 PWM Dead-time Control Register	0x0000_0000
TIMER9_PWMCNT	TMR89_BA+0x15C	R	Timer9 PWM Counter Register	0x0000_0000
TIMER9_PWMMSKEN	TMR89_BA+0x160	R/W	Timer9 PWM Output Mask Enable Register	0x0000_0000
TIMER9_PWMMSK	TMR89_BA+0x164	R/W	Timer9 PWM Output Mask Data Control Register	0x0000_0000
TIMER9_PWMBNF	TMR89_BA+0x168	R/W	Timer9 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER9_PWMFAILBRK	TMR89_BA+0x16C	R/W	Timer9 PWM System Fail Brake Control Register	0x0000_0000
TIMER9_PWMBRKCTL	TMR89_BA+0x170	R/W	Timer9 PWM Brake Control Register	0x0000_0000

TIMER9_PWMPOLCTL	TMR89_BA+0x174	R/W	Timer9 PWM Pin Output Polar Control Register	0x0000_0000
TIMER9_PWMPOEN	TMR89_BA+0x178	R/W	Timer9 PWM Pin Output Enable Register	0x0000_0000
TIMER9_PWMSWBRK	TMR89_BA+0x17C	W	Timer9 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER9_PWMINTEN0	TMR89_BA+0x180	R/W	Timer9 PWM Interrupt Enable Register 0	0x0000_0000
TIMER9_PWMINTEN1	TMR89_BA+0x184	R/W	Timer9 PWM Interrupt Enable Register 1	0x0000_0000
TIMER9_PWMINTSTS0	TMR89_BA+0x188	R/W	Timer9 PWM Interrupt Status Register 0	0x0000_0000
TIMER9_PWMINTSTS1	TMR89_BA+0x18C	R/W	Timer9 PWM Interrupt Status Register 1	0x0000_0000
TIMER9_PWMEADCTS	TMR89_BA+0x190	R/W	Timer9 PWM EADC Trigger Source Select Register	0x0000_0000
TIMER9_PWMSCTL	TMR89_BA+0x194	R/W	Timer9 PWM Synchronous Control Register	0x0000_0000
TIMER9_PWMSTATUS	TMR89_BA+0x19C	R/W	Timer9 PWM Status Register	0x0000_0000
TIMER9_PWMPBUF	TMR89_BA+0x1A0	R	Timer9 PWM Period Buffer Register	0x0000_0000
TIMER9_PWMCMPBUF	TMR89_BA+0x1A4	R	Timer9 PWM Comparator Buffer Register	0x0000_0000
TIMER10_CTL	TMR1011_BA+0x00	R/W	Timer10 Control Register	0x0000_0005
TIMER10_CMP	TMR1011_BA+0x04	R/W	Timer10 Comparator Register	0x0000_0000
TIMER10_INTSTS	TMR1011_BA+0x08	R/W	Timer10 Interrupt Status Register	0x0000_0000
TIMER10_CNT	TMR1011_BA+0x0C	R/W	Timer10 Data Register	0x0000_0000
TIMER10_CAP	TMR1011_BA+0x10	R	Timer10 Capture Data Register	0x0000_0000
TIMER10_EXTCTL	TMR1011_BA+0x14	R/W	Timer10 External Control Register	0x0000_0000
TIMER10_EINTSTS	TMR1011_BA+0x18	R/W	Timer10 External Interrupt Status Register	0x0000_0000
TIMER10_TRGCTL	TMR1011_BA+0x1C	R/W	Timer10 Trigger Control Register	0x0000_0000
TIMER10_ALTCTL	TMR1011_BA+0x20	R/W	Timer10 Alternative Control Register	0x0000_0000
TIMER10_PWMCTL	TMR1011_BA+0x40	R/W	Timer10 PWM Control Register	0x0000_0000
TIMER10_PWMCLKSRC	TMR1011_BA+0x44	R/W	Timer10 PWM Counter Clock Source Register	0x0000_0000
TIMER10_PWMCLKPSC	TMR1011_BA+0x48	R/W	Timer10 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER10_PWMCNTCLR	TMR1011_BA+0x4C	R/W	Timer10 PWM Clear Counter Register	0x0000_0000
TIMER10_PWMPERIOD	TMR1011_BA+0x50	R/W	Timer10 PWM Period Register	0x0000_0000
TIMER10_PWMCMPDAT	TMR1011_BA+0x54	R/W	Timer10 PWM Comparator Register	0x0000_0000
TIMER10_PWMDTCTL	TMR1011_BA+0x58	R/W	Timer10 PWM Dead-time Control Register	0x0000_0000
TIMER10_PWMCNT	TMR1011_BA+0x5C	R	Timer10 PWM Counter Register	0x0000_0000
TIMER10_PWMMSKEN	TMR1011_BA+0x60	R/W	Timer10 PWM Output Mask Enable Register	0x0000_0000
TIMER10_PWMMSK	TMR1011_BA+0x64	R/W	Timer10 PWM Output Mask Data Control Register	0x0000_0000

TIMER10_PWMBNF	TMR1011_BA+0x68	R/W	Timer10 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER10_PWMFAILBRK	TMR1011_BA+0x6C	R/W	Timer10 PWM System Fail Brake Control Register	0x0000_0000
TIMER10_PWMBRKCTL	TMR1011_BA+0x70	R/W	Timer10 PWM Brake Control Register	0x0000_0000
TIMER10_PWMPOLCTL	TMR1011_BA+0x74	R/W	Timer10 PWM Pin Output Polar Control Register	0x0000_0000
TIMER10_PWMPOEN	TMR1011_BA+0x78	R/W	Timer10 PWM Pin Output Enable Register	0x0000_0000
TIMER10_PWMSWBRK	TMR1011_BA+0x7C	W	Timer10 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER10_PWMINTEN0	TMR1011_BA+0x80	R/W	Timer10 PWM Interrupt Enable Register 0	0x0000_0000
TIMER10_PWMINTEN1	TMR1011_BA+0x84	R/W	Timer10 PWM Interrupt Enable Register 1	0x0000_0000
TIMER10_PWMINTSTS0	TMR1011_BA+0x88	R/W	Timer10 PWM Interrupt Status Register 0	0x0000_0000
TIMER10_PWMINTSTS1	TMR1011_BA+0x8C	R/W	Timer10 PWM Interrupt Status Register 1	0x0000_0000
TIMER10_PWMEADCTS	TMR1011_BA+0x90	R/W	Timer10 PWM EADC Trigger Source Select Register	0x0000_0000
TIMER10_PWMSCTL	TMR1011_BA+0x94	R/W	Timer10 PWM Synchronous Control Register	0x0000_0000
TIMER10_PWMSTRG	TMR1011_BA+0x98	W	Timer10 PWM Synchronous Trigger Register	0x0000_0000
TIMER10_PWMSTATUS	TMR1011_BA+0x9C	R/W	Timer10 PWM Status Register	0x0000_0000
TIMER10_PWMPBUF	TMR1011_BA+0xA0	R	Timer10 PWM Period Buffer Register	0x0000_0000
TIMER10_PWMCMPBUF	TMR1011_BA+0xA4	R	Timer10 PWM Comparator Buffer Register	0x0000_0000
TIMER11_CTL	TMR1011_BA+0x100	R/W	Timer11 Control Register	0x0000_0005
TIMER11_CMP	TMR1011_BA+0x104	R/W	Timer11 Comparator Register	0x0000_0000
TIMER11_INTSTS	TMR1011_BA+0x108	R/W	Timer11 Interrupt Status Register	0x0000_0000
TIMER11_CNT	TMR1011_BA+0x10C	R/W	Timer11 Data Register	0x0000_0000
TIMER11_CAP	TMR1011_BA+0x110	R	Timer11 Capture Data Register	0x0000_0000
TIMER11_EXTCTL	TMR1011_BA+0x114	R/W	Timer11 External Control Register	0x0000_0000
TIMER11_EINTSTS	TMR1011_BA+0x118	R/W	Timer11 External Interrupt Status Register	0x0000_0000
TIMER11_TRGCTL	TMR1011_BA+0x11C	R/W	Timer11 Trigger Control Register	0x0000_0000
TIMER11_ALTCTL	TMR1011_BA+0x120	R/W	Timer11 Alternative Control Register	0x0000_0000
TIMER11_PWMCTL	TMR1011_BA+0x140	R/W	Timer11 PWM Control Register	0x0000_0000
TIMER11_PWMCLKSRC	TMR1011_BA+0x144	R/W	Timer11 PWM Counter Clock Source Register	0x0000_0000
TIMER11_PWMCLKPSC	TMR1011_BA+0x148	R/W	Timer11 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER11_PWMCNTCLR	TMR1011_BA+0x14C	R/W	Timer11 PWM Clear Counter Register	0x0000_0000
TIMER11_PWMPERIOD	TMR1011_BA+0x150	R/W	Timer11 PWM Period Register	0x0000_0000
TIMER11_PWMCMPDAT	TMR1011_BA+0x154	R/W	Timer11 PWM Comparator Register	0x0000_0000

TIMER11_PWMDTCTL	TMR1011_BA+0x158	R/W	Timer11 PWM Dead-time Control Register	0x0000_0000
TIMER11_PWMCNT	TMR1011_BA+0x15C	R	Timer11 PWM Counter Register	0x0000_0000
TIMER11_PWMMSKEN	TMR1011_BA+0x160	R/W	Timer11 PWM Output Mask Enable Register	0x0000_0000
TIMER11_PWMMSK	TMR1011_BA+0x164	R/W	Timer11 PWM Output Mask Data Control Register	0x0000_0000
TIMER11_PWMBNF	TMR1011_BA+0x168	R/W	Timer11 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER11_PWMFAILBRK	TMR1011_BA+0x16C	R/W	Timer11 PWM System Fail Brake Control Register	0x0000_0000
TIMER11_PWMBRKCTL	TMR1011_BA+0x170	R/W	Timer11 PWM Brake Control Register	0x0000_0000
TIMER11_PWMPOLCTL	TMR1011_BA+0x174	R/W	Timer11 PWM Pin Output Polar Control Register	0x0000_0000
TIMER11_PWMPOEN	TMR1011_BA+0x178	R/W	Timer11 PWM Pin Output Enable Register	0x0000_0000
TIMER11_PWMSWBRK	TMR1011_BA+0x17C	W	Timer11 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER11_PWMINTEN0	TMR1011_BA+0x180	R/W	Timer11 PWM Interrupt Enable Register 0	0x0000_0000
TIMER11_PWMINTEN1	TMR1011_BA+0x184	R/W	Timer11 PWM Interrupt Enable Register 1	0x0000_0000
TIMER11_PWMINTSTS0	TMR1011_BA+0x188	R/W	Timer11 PWM Interrupt Status Register 0	0x0000_0000
TIMER11_PWMINTSTS1	TMR1011_BA+0x18C	R/W	Timer11 PWM Interrupt Status Register 1	0x0000_0000
TIMER11_PWMEADCTS	TMR1011_BA+0x190	R/W	Timer11 PWM EADC Trigger Source Select Register	0x0000_0000
TIMER11_PWMSCTL	TMR1011_BA+0x194	R/W	Timer11 PWM Synchronous Control Register	0x0000_0000
TIMER11_PWMSTATUS	TMR1011_BA+0x19C	R/W	Timer11 PWM Status Register	0x0000_0000
TIMER11_PWMPBUF	TMR1011_BA+0x1A0	R	Timer11 PWM Period Buffer Register	0x0000_0000
TIMER11_PWMCMPBUF	TMR1011_BA+0x1A4	R	Timer11 PWM Comparator Buffer Register	0x0000_0000

6.17.8 Register Description

Timer Control Register (TIMERx_CTL)

Register	Offset	R/W	Description	Reset Value
TIMER0_CTL	TMR01_BA+0x00	R/W	Timer0 Control Register	0x0000_0005
TIMER1_CTL	TMR01_BA+0x100	R/W	Timer1 Control Register	0x0000_0005
TIMER2_CTL	TMR23_BA+0x00	R/W	Timer2 Control Register	0x0000_0005
TIMER3_CTL	TMR23_BA+0x100	R/W	Timer3 Control Register	0x0000_0005
TIMER4_CTL	TMR45_BA+0x00	R/W	Timer4 Control Register	0x0000_0005
TIMER5_CTL	TMR45_BA+0x100	R/W	Timer5 Control Register	0x0000_0005
TIMER6_CTL	TMR67_BA+0x00	R/W	Timer6 Control Register	0x0000_0005
TIMER7_CTL	TMR67_BA+0x100	R/W	Timer7 Control Register	0x0000_0005
TIMER8_CTL	TMR89_BA+0x00	R/W	Timer8 Control Register	0x0000_0005
TIMER9_CTL	TMR89_BA+0x100	R/W	Timer9 Control Register	0x0000_0005
TIMER10_CTL	TMR1011_BA+0x00	R/W	Timer10 Control Register	0x0000_0005
TIMER11_CTL	TMR1011_BA+0x100	R/W	Timer11 Control Register	0x0000_0005

31	30	29	28	27	26	25	24
ICEDEBUG	CNTEN	INTEN	OPMODE		Reserved	ACTSTS	EXTCNTEN
23	22	21	20	19	18	17	16
WKEN	CAPSRC	TGLPINSEL	PERIOSEL	INTRGEN	Reserved		
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
PSC							

Bits	Description	
[31]	**ICEDEBUG**	**ICE Debug Mode Acknowledge Disable Bit (Write Protect)** 0 = ICE debug mode acknowledgement effects TIMER counting. TIMER counter will be held while CPU is held by ICE. 1 = ICE debug mode acknowledgement Disabled. TIMER counter will keep going no matter CPU is held by ICE or not. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register.
[30]	**CNTEN**	**Timer Counting Enable Bit** 0 = Stops/Suspends counting.

		1 = Starts counting. **Note 1:** In stop status, and then set CNTEN to 1 will enable the 24-bit up counter to keep counting from the last stop counting value. **Note 2:** This bit is auto-cleared by hardware in one-shot mode (TIMER_CTL[28:27] = 00) when the timer time-out interrupt flag TIF (TIMERx_INTSTS[0]) is generated. **Note 3:** Set enable/disable this bit needs 2 * TMR_CLK period to become active, user can read ACTSTS (TIMERx_CTL[25]) to check enable/disable command is completed or not.
[29]	**INTEN**	**Timer Interrupt Enable Bit** 0 = Timer time-out interrupt Disabled. 1 = Timer time-out interrupt Enabled. **Note:** If this bit is enabled, when the timer time-out interrupt flag TIF is set to 1, the timer interrupt signal is generated and inform to CPU.
[28:27]	**OPMODE**	**Timer Counting Mode Select** 00 = The timer controller is operated in One-shot mode. 01 = The timer controller is operated in Periodic mode. 10 = The timer controller is operated in Toggle-output mode. 11 = The timer controller is operated in Continuous Counting mode.
[26]	**Reserved**	Reserved.
[25]	**ACTSTS**	**Timer Active Status Bit (Read Only)** This bit indicates the 24-bit up counter status. 0 = 24-bit up counter is not active. 1 = 24-bit up counter is active. **Note:** This bit may active when CNT 0 transition to CNT 1.
[24]	**EXTCNTEN**	**Event Counter Mode Enable Bit** This bit is for external counting pin function enabled. 0 = Event counter mode Disabled. 1 = Event counter mode Enabled. **Note:** When timer is used as an event counter, this bit should be set to 1 and select PCLK as timer clock source.
[23]	**WKEN**	**Wake-up Function Enable Bit** If this bit is set to 1, while timer interrupt flag TIF (TIMERx_INTSTS[0]) is 1 and INTEN (TIMERx_CTL[29]) is enabled, the timer interrupt signal will generate a wake-up trigger event to CPU. 0 = Wake-up function Disabled if timer interrupt signal generated. 1 = Wake-up function Enabled if timer interrupt signal generated.
[22]	**CAPSRC**	**Capture Pin Source Selection** 0 = Capture Function source is from TMx_EXT (x= 0~11) pin. 1 = Capture Function source is from internal clock source (HIRC 12M, LIRC) or external clock (HXT, LXT). **Note:** When CAPSRC = 1, user can set ICAPSSEL (TIMERx_EXTCTL[10:8]) to decide which clock is as timer capture source.
[21]	**TGLPINSEL**	**Toggle-output Pin Select** 0 = Toggle mode output to TMx (Timer Event Counter Pin). 1 = Toggle mode output to TMx_EXT (Timer External Capture Pin).
[20]	**PERIOSEL**	**Periodic Mode Behavior Selection Enable Bit** 0 = The behavior selection in periodic mode is Disabled. When user updates CMPDAT while timer is running in periodic mode, CNT will be reset to default value. 1 = The behavior selection in periodic mode is Enabled. When user update CMPDAT while timer is running in periodic mode, the limitations as bellows list,

		If updated CMPDAT value > CNT, CMPDAT will be updated and CNT keep running continually. If updated CMPDAT value = CNT, timer time-out interrupt will be asserted immediately. If updated CMPDAT value < CNT, CNT will be reset to default value.
[19]	**INTRGEN**	**Inter-timer Trigger Mode Enable Bit** Setting this bit will enable the inter-timer trigger capture function. The Timer0/2/4/6/8/10 will be in event counter mode and counting with external clock source or event. Also, Timer1/3/5/7/9/11 will be in trigger-counting mode of capture function. 0 = Inter-Timer Trigger Capture mode Disabled. 1 = Inter-Timer Trigger Capture mode Enabled. **Note:** For Timer1/3/5/7/9/11, this bit is ignored and the read back value is always 0.
[18:8]	**Reserved**	Reserved.
[7:0]	**PSC**	**Prescale Counter** Timer input clock or event source is divided by (PSC+1) before it is fed to the timer up counter. If this field is 0 (PSC = 0), then there is no scaling. **Note:** Update prescale counter value will reset internal 8-bit prescale counter and 24-bit up counter value.

Timer Comparator Register (TIMERx_CMP)

Register	Offset	R/W	Description	Reset Value
TIMER0_CMP	TMR01_BA+0x04	R/W	Timer0 Comparator Register	0x0000_0000
TIMER1_CMP	TMR01_BA+0x104	R/W	Timer1 Comparator Register	0x0000_0000
TIMER2_CMP	TMR23_BA+0x04	R/W	Timer2 Comparator Register	0x0000_0000
TIMER3_CMP	TMR23_BA+0x104	R/W	Timer3 Comparator Register	0x0000_0000
TIMER4_CMP	TMR45_BA+0x04	R/W	Timer4 Comparator Register	0x0000_0000
TIMER5_CMP	TMR45_BA+0x104	R/W	Timer5 Comparator Register	0x0000_0000
TIMER6_CMP	TMR67_BA+0x04	R/W	Timer6 Comparator Register	0x0000_0000
TIMER7_CMP	TMR67_BA+0x104	R/W	Timer7 Comparator Register	0x0000_0000
TIMER8_CMP	TMR89_BA+0x04	R/W	Timer8 Comparator Register	0x0000_0000
TIMER9_CMP	TMR89_BA+0x104	R/W	Timer9 Comparator Register	0x0000_0000
TIMER10_CMP	TMR1011_BA+0x04	R/W	Timer10 Comparator Register	0x0000_0000
TIMER11_CMP	TMR1011_BA+0x104	R/W	Timer11 Comparator Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CMPDAT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CMPDAT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CMPDAT							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**CMPDAT**	**Timer Comparator Value** CMPDAT is a 24-bit compared value register. When the internal 24-bit up counter value is equal to CMPDAT value, the TIF (TIMERx_INTSTS[0] Timer Interrupt Flag) will set to 1. Time-out period = (Period of timer clock input) * (8-bit PSC + 1) * (24-bit CMPDAT). **Note 1:** Never write 0x0 or 0x1 in CMPDAT field, or the core will run into unknown state. **Note 2:** When timer is operating at continuous counting mode, the 24-bit up counter will keep counting continuously even if user writes a new value into CMPDAT field. But if timer is operating at other modes, the 24-bit up counter will restart counting from 0 and using newest CMPDAT value to be the timer compared value while user writes a new value into CMPDAT field.

Timer Interrupt Status Register (TIMERx_INTSTS)

Register	Offset	R/W	Description	Reset Value
TIMER0_INTSTS	TMR01_BA+0x08	R/W	Timer0 Interrupt Status Register	0x0000_0000
TIMER1_INTSTS	TMR01_BA+0x108	R/W	Timer1 Interrupt Status Register	0x0000_0000
TIMER2_INTSTS	TMR23_BA+0x08	R/W	Timer2 Interrupt Status Register	0x0000_0000
TIMER3_INTSTS	TMR23_BA+0x108	R/W	Timer3 Interrupt Status Register	0x0000_0000
TIMER4_INTSTS	TMR45_BA+0x08	R/W	Timer4 Interrupt Status Register	0x0000_0000
TIMER5_INTSTS	TMR45_BA+0x108	R/W	Timer5 Interrupt Status Register	0x0000_0000
TIMER6_INTSTS	TMR67_BA+0x08	R/W	Timer6 Interrupt Status Register	0x0000_0000
TIMER7_INTSTS	TMR67_BA+0x108	R/W	Timer7 Interrupt Status Register	0x0000_0000
TIMER8_INTSTS	TMR89_BA+0x08	R/W	Timer8 Interrupt Status Register	0x0000_0000
TIMER9_INTSTS	TMR89_BA+0x108	R/W	Timer9 Interrupt Status Register	0x0000_0000
TIMER10_INTSTS	TMR1011_BA+0x08	R/W	Timer10 Interrupt Status Register	0x0000_0000
TIMER11_INTSTS	TMR1011_BA+0x108	R/W	Timer11 Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						TWKF	TIF

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**TWKF**	**Timer Wake-up Flag** This bit indicates the interrupt wake-up flag status of timer. 0 = Timer does not cause CPU wake-up. 1 = CPU wake-up from Idle or Power-down mode if timer time-out interrupt signal generated. **Note:** This bit is cleared by writing 1 to it.
[0]	**TIF**	**Timer Interrupt Flag** This bit indicates the interrupt flag status of Timer while 24-bit timer up counter CNT (TIMERx_CNT[23:0]) value reaches to CMPDAT (TIMERx_CMP[23:0]) value. 0 = No effect.

		1 = CNT value matches the CMPDAT value. **Note:** This bit is cleared by writing 1 to it.

Timer Data Register (TIMERx_CNT)

Register	Offset	R/W	Description	Reset Value
TIMER0_CNT	TMR01_BA+0x0C	R/W	Timer0 Data Register	0x0000_0000
TIMER1_CNT	TMR01_BA+0x10C	R/W	Timer1 Data Register	0x0000_0000
TIMER2_CNT	TMR23_BA+0x0C	R/W	Timer2 Data Register	0x0000_0000
TIMER3_CNT	TMR23_BA+0x10C	R/W	Timer3 Data Register	0x0000_0000
TIMER4_CNT	TMR45_BA+0x0C	R/W	Timer4 Data Register	0x0000_0000
TIMER5_CNT	TMR45_BA+0x10C	R/W	Timer5 Data Register	0x0000_0000
TIMER6_CNT	TMR67_BA+0x0C	R/W	Timer6 Data Register	0x0000_0000
TIMER7_CNT	TMR67_BA+0x10C	R/W	Timer7 Data Register	0x0000_0000
TIMER8_CNT	TMR89_BA+0x0C	R/W	Timer8 Data Register	0x0000_0000
TIMER9_CNT	TMR89_BA+0x10C	R/W	Timer9 Data Register	0x0000_0000
TIMER10_CNT	TMR1011_BA+0x0C	R/W	Timer10 Data Register	0x0000_0000
TIMER11_CNT	TMR1011_BA+0x10C	R/W	Timer11 Data Register	0x0000_0000

31	30	29	28	27	26	25	24
RSTACT	**Reserved**						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CNT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CNT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNT							

Bits	Description	
[31]	**RSTACT**	**Timer Data Register Reset Active (Read Only)** This bit indicates if the counter reset operation active. When user writes this CNT register, timer starts to reset its internal 24-bit timer up-counter to 0 and reload 8-bit pre-scale counter. At the same time, timer set this flag to 1 to indicate the counter reset operation is in progress. Once the counter reset operation done, timer clear this bit to 0 automatically. 0 = Reset operation is done. 1 = Reset operation triggered by writing TIMERx_CNT is in progress.
[30:24]	**Reserved**	Reserved.
[23:0]	**CNT**	**Timer Data Register** Read operation.

		Read this register to get CNT value. For example: If EXTCNTEN (TIMERx_CTL[24]) is 0, user can read CNT value for getting current 24-bit counter value. If EXTCNTEN (TIMERx_CTL[24]) is 1, user can read CNT value for getting current 24-bit event input counter value. Write operation. Writing any value to this register will reset current CNT value to 0 and reload internal 8-bit prescale counter.

Timer Capture Data Register (TIMERx_CAP)

Register	Offset	R/W	Description	Reset Value
TIMER0_CAP	TMR01_BA+0x10	R	Timer0 Capture Data Register	0x0000_0000
TIMER1_CAP	TMR01_BA+0x110	R	Timer1 Capture Data Register	0x0000_0000
TIMER2_CAP	TMR23_BA+0x10	R	Timer2 Capture Data Register	0x0000_0000
TIMER3_CAP	TMR23_BA+0x110	R	Timer3 Capture Data Register	0x0000_0000
TIMER4_CAP	TMR45_BA+0x10	R	Timer4 Capture Data Register	0x0000_0000
TIMER5_CAP	TMR45_BA+0x110	R	Timer5 Capture Data Register	0x0000_0000
TIMER6_CAP	TMR67_BA+0x10	R	Timer6 Capture Data Register	0x0000_0000
TIMER7_CAP	TMR67_BA+0x110	R	Timer7 Capture Data Register	0x0000_0000
TIMER8_CAP	TMR89_BA+0x10	R	Timer8 Capture Data Register	0x0000_0000
TIMER9_CAP	TMR89_BA+0x110	R	Timer9 Capture Data Register	0x0000_0000
TIMER10_CAP	TMR1011_BA+0x10	R	Timer10 Capture Data Register	0x0000_0000
TIMER11_CAP	TMR1011_BA+0x110	R	Timer11 Capture Data Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CAPDAT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CAPDAT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CAPDAT							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**CAPDAT**	**Timer Capture Data Register** When CAPEN (TIMERx_EXTCTL[3]) bit is set, CAPFUNCS (TIMERx_EXTCTL[4]) bit is 0, and a transition on TMx_EXT pin matched the CAPEDGE (TIMERx_EXTCTL[14:12]) setting, CAPIF (TIMERx_EINTSTS[0]) will set to 1 and the current timer counter value CNT (TIMERx_CNT[23:0]) will be auto-loaded into this CAPDAT field.

Timer External Control Register (TIMERx_EXTCTL)

Register	Offset	R/W	Description	Reset Value
TIMER0_EXTCTL	TMR01_BA+0x14	R/W	Timer0 External Control Register	0x0000_0000
TIMER1_EXTCTL	TMR01_BA+0x114	R/W	Timer1 External Control Register	0x0000_0000
TIMER2_EXTCTL	TMR23_BA+0x14	R/W	Timer2 External Control Register	0x0000_0000
TIMER3_EXTCTL	TMR23_BA+0x114	R/W	Timer3 External Control Register	0x0000_0000
TIMER4_EXTCTL	TMR45_BA+0x14	R/W	Timer4 External Control Register	0x0000_0000
TIMER5_EXTCTL	TMR45_BA+0x114	R/W	Timer5 External Control Register	0x0000_0000
TIMER6_EXTCTL	TMR67_BA+0x14	R/W	Timer6 External Control Register	0x0000_0000
TIMER7_EXTCTL	TMR67_BA+0x114	R/W	Timer7 External Control Register	0x0000_0000
TIMER8_EXTCTL	TMR89_BA+0x14	R/W	Timer8 External Control Register	0x0000_0000
TIMER9_EXTCTL	TMR89_BA+0x114	R/W	Timer9 External Control Register	0x0000_0000
TIMER10_EXTCTL	TMR1011_BA+0x14	R/W	Timer10 External Control Register	0x0000_0000
TIMER11_EXTCTL	TMR1011_BA+0x114	R/W	Timer11 External Control Register	0x0000_0000

31	30	29	28	27	26	25	24
CAPSDIV				**Reserved**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							**ECNTSSEL**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**CAPEDGE**			**Reserved**	**ICAPSSEL**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNTDBEN	**CAPDBEN**	**CAPIEN**	**CAPFUNCS**	**CAPEN**	**Reserved**		**CNTPHASE**

Bits	Description	
[31:28]	**CAPSDIV**	**Timer Capture Source Divider** This bits indicate the divide scale for capture source. 0000 = Capture source/1. 0001 = Capture source/2. 0010 = Capture source/4. 0011 = Capture source/8. 0100 = Capture source/16. 0101 = Capture source/32. 0110 = Capture source/64. 0111 = Capture source/128.

		1000 = Capture source/256. 1001~1111 = Reserved. **Note:** Sets ICAPSSEL (TIMERx_EXTCTL[10:8]) and CAPSRC (TIMERx_CTL[22]) to select capture source.
[27:17]	**Reserved**	Reserved.
[16]	**ECNTSSEL**	**Event Counter Source Selection to Trigger Event Counter Function** 0 = Event Counter input source is from TMx (x= 0~11) pin.
[15]	**Reserved**	Reserved.
[14:12]	**CAPEDGE**	**Timer External Capture Pin Edge Detect** When first capture event is generated, the CNT (TIMERx_CNT[23:0]) will be reset to 0 and first CAPDAT (TIMERx_CAP[23:0]) should be to 0. 000 = Capture event occurred when detect falling edge transfer on TMx_EXT (x= 0~11) pin. 001 = Capture event occurred when detect rising edge transfer on TMx_EXT (x= 0~11) pin. 010 = Capture event occurred when detect both falling and rising edge transfer on TMx_EXT (x= 0~11) pin, and first capture event occurred at falling edge transfer. 011 = Capture event occurred when detect both rising and falling edge transfer on TMx_EXT (x= 0~11) pin, and first capture event occurred at rising edge transfer. 110 = First capture event occurred at falling edge, follows capture events are at rising edge transfer on TMx_EXT (x= 0~11) pin. 111 = First capture event occurred at rising edge, follows capture events are at falling edge transfer on TMx_EXT (x= 0~11) pin. 100, 101 = Reserved.
[11]	**Reserved**	Reserved.
[10:8]	**ICAPSSEL**	**Internal Capture Source Selection** 010 = Capture Function source is from HXT. 011 = Capture Function source is from LXT. 100 = Capture Function source is from HIRC. 101 = Capture Function source is from LIRC. 110 = Reserved. 111 = Reserved. **Note:** these bits only available when CAPSRC (TIMERx_CTL[22]) is 1.
[7]	**CNTDBEN**	**Timer Counter Pin De-bounce Enable Bit** 0 = TMx (x= 0~11) pin de-bounce Disabled. 1 = TMx (x= 0~11) pin de-bounce Enabled. **Note:** If this bit is enabled, the edge detection of TMx pin is detected with de-bounce circuit.
[6]	**CAPDBEN**	**Timer External Capture Pin De-bounce Enable Bit** 0 = TMx_EXT (x= 0~11) pin de-bounce Disabled. 1 = TMx_EXT (x= 0~11) pin de-bounce Enabled. **Note:** If this bit is enabled, the edge detection of TMx_EXT pin is detected with de-bounce circuit.
[5]	**CAPIEN**	**Timer External Capture Interrupt Enable Bit** 0 = TMx_EXT (x= 0~11) pin detection Interrupt Disabled. 1 = TMx_EXT (x= 0~11) pin detection Interrupt Enabled. **Note:** CAPIEN is used to enable timer external interrupt. If CAPIEN enabled, timer will rise an interrupt when CAPIF (TIMERx_EINTSTS[0]) is 1. For example, while CAPIEN = 1, CAPEN = 1, and CAPEDGE = 00, a 1 to 0 transition on the TMx_EXT pin will cause the CAPIF to be set then the interrupt signal is generated and sent to NVIC to inform CPU.
[4]	**CAPFUNCS**	**Capture Function Selection** 0 = External Capture Mode Enabled.

		1 = External Reset Mode Enabled. **Note 1:** When CAPFUNCS is 0, transition on TMx_EXT (x= 0~11) pin is using to save current 24-bit timer counter value (CNT value) to CAPDAT field. **Note 2:** When CAPFUNCS is 1, transition on TMx_EXT (x= 0~11) pin is using to save current 24-bit timer counter value (CNT value) to CAPDAT field then CNT value will be reset immediately.
[3]	**CAPEN**	**Timer External Capture Pin Enable Bit** This bit enables the TMx_EXT capture pin input function. 0 =TMx_EXT (x= 0~11) pin Disabled. 1 =TMx_EXT (x= 0~11) pin Enabled.
[2:1]	**Reserved**	Reserved.
[0]	**CNTPHASE**	**Timer External Count Phase** This bit indicates the detection phase of external counting pin TMx (x= 0~11). 0 = A falling edge of external counting pin will be counted. 1 = A rising edge of external counting pin will be counted.

Timer External Interrupt Status Register (TIMERx_EINTSTS)

Register	Offset	R/W	Description	Reset Value
TIMER0_EINTSTS	TMR01_BA+0x18	R/W	Timer0 External Interrupt Status Register	0x0000_0000
TIMER1_EINTSTS	TMR01_BA+0x118	R/W	Timer1 External Interrupt Status Register	0x0000_0000
TIMER2_EINTSTS	TMR23_BA+0x18	R/W	Timer2 External Interrupt Status Register	0x0000_0000
TIMER3_EINTSTS	TMR23_BA+0x118	R/W	Timer3 External Interrupt Status Register	0x0000_0000
TIMER4_EINTSTS	TMR45_BA+0x18	R/W	Timer4 External Interrupt Status Register	0x0000_0000
TIMER5_EINTSTS	TMR45_BA+0x118	R/W	Timer5 External Interrupt Status Register	0x0000_0000
TIMER6_EINTSTS	TMR67_BA+0x18	R/W	Timer6 External Interrupt Status Register	0x0000_0000
TIMER7_EINTSTS	TMR67_BA+0x118	R/W	Timer7 External Interrupt Status Register	0x0000_0000
TIMER8_EINTSTS	TMR89_BA+0x18	R/W	Timer8 External Interrupt Status Register	0x0000_0000
TIMER9_EINTSTS	TMR89_BA+0x118	R/W	Timer9 External Interrupt Status Register	0x0000_0000
TIMER10_EINTSTS	TMR1011_BA+0x18	R/W	Timer10 External Interrupt Status Register	0x0000_0000
TIMER11_EINTSTS	TMR1011_BA+0x118	R/W	Timer11 External Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**CAPIF**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**CAPIF**	**Timer External Capture Interrupt Flag** This bit indicates the timer external capture interrupt flag status. 0 = TMx_EXT (x= 0~11) pin interrupt did not occur. 1 = TMx_EXT (x= 0~11) pin interrupt occurred. **Note 1:** This bit is cleared by writing 1 to it. **Note 2:** When CAPEN (TIMERx_EXTCTL[3]) bit is set, CAPFUNCS (TIMERx_EXTCTL[4]) bit is 0, and a transition on TMx_EXT (x= 0~11) pin matched the CAPEDGE (TIMERx_EXTCTL[2:1]) setting, this bit will set to 1 by hardware. **Note 3:** There is a new incoming capture event detected before CPU clearing the CAPIF status. If the above condition occurred, the Timer will keep register TIMERx_CAP unchanged and drop the new capture value.

Timer Trigger Control Register (TIMERx_TRGCTL)

Register	Offset	R/W	Description	Reset Value
TIMER0_TRGCTL	TMR01_BA+0x1C	R/W	Timer0 Trigger Control Register	0x0000_0000
TIMER1_TRGCTL	TMR01_BA+0x11C	R/W	Timer1 Trigger Control Register	0x0000_0000
TIMER2_TRGCTL	TMR23_BA+0x1C	R/W	Timer2 Trigger Control Register	0x0000_0000
TIMER3_TRGCTL	TMR23_BA+0x11C	R/W	Timer3 Trigger Control Register	0x0000_0000
TIMER4_TRGCTL	TMR45_BA+0x1C	R/W	Timer4 Trigger Control Register	0x0000_0000
TIMER5_TRGCTL	TMR45_BA+0x11C	R/W	Timer5 Trigger Control Register	0x0000_0000
TIMER6_TRGCTL	TMR67_BA+0x1C	R/W	Timer6 Trigger Control Register	0x0000_0000
TIMER7_TRGCTL	TMR67_BA+0x11C	R/W	Timer7 Trigger Control Register	0x0000_0000
TIMER8_TRGCTL	TMR89_BA+0x1C	R/W	Timer8 Trigger Control Register	0x0000_0000
TIMER9_TRGCTL	TMR89_BA+0x11C	R/W	Timer9 Trigger Control Register	0x0000_0000
TIMER10_TRGCTL	TMR1011_BA+0x1C	R/W	Timer10 Trigger Control Register	0x0000_0000
TIMER11_TRGCTL	TMR1011_BA+0x11C	R/W	Timer11 Trigger Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	3	4	3	2	1	0
Reserved			TRGPDMA	Reserved	TRGEADC	TRGPWM	TRGSSEL

Bits	Description	
[31:5]	**Reserved**	Reserved.
[4]	**TRGPDMA**	**Trigger PDMA Enable Bit** If this bit is set to 1, each timer time-out event or capture event can be triggered PDMA transfer. 0 = Timer interrupt trigger PDMA Disabled. 1 = Timer interrupt trigger PDMA Enabled. **Note:** If TRGSSEL (TIMERx_TRGCTL[0]) = 0, time-out interrupt signal will trigger PDMA transfer. If TRGSSEL (TIMERx_TRGCTL[0]) = 1, capture interrupt signal will trigger PDMA transfer.
[3]	**Reserved**	Reserved.
[2]	**TRGEADC**	**Trigger EADC Enable Bit**

		If this bit is set to 1, each timer time-out event or capture event can be triggered EADC conversion. 0 = Timer interrupt trigger EADC Disabled. 1 = Timer interrupt trigger EADC Enabled. **Note:** If TRGSSEL (TIMERx_TRGCTL[0]) = 0, time-out interrupt signal will trigger EADC conversion. If TRGSSEL (TIMERx_TRGCTL[0]) = 1, capture interrupt signal will trigger EADC conversion.
[1]	**TRGPWM**	**Trigger EPWM Enable Bit** If this bit is set to 1, each timer time-out event or capture event can be as EPWM counter clock source. 0 = Timer interrupt trigger EPWM Disabled. 1 = Timer interrupt trigger EPWM Enabled. **Note:** If TRGSSEL (TIMERx_TRGCTL[0]) = 0, time-out interrupt signal as EPWM counter clock source. If TRGSSEL (TIMERx_TRGCTL[0]) = 1, capture interrupt signal as EPWM counter clock source.
[0]	**TRGSSEL**	**Trigger Source Select Bit** This bit is used to select internal trigger source is form timer time-out interrupt signal or capture interrupt signal. 0 = Time-out interrupt signal is used to internal trigger EPWM, PDMA and EADC. 1 = Capture interrupt signal is used to internal trigger EPWM, PDMA and EADC.

Timer Alternative Control Register (TIMERx_ALTCTL)

Register	Offset	R/W	Description	Reset Value
TIMER0_ALTCTL	TMR01_BA+0x20	R/W	Timer0 Alternative Control Register	0x0000_0000
TIMER1_ALTCTL	TMR01_BA+0x120	R/W	Timer1 Alternative Control Register	0x0000_0000
TIMER2_ALTCTL	TMR23_BA+0x20	R/W	Timer2 Alternative Control Register	0x0000_0000
TIMER3_ALTCTL	TMR23_BA+0x120	R/W	Timer3 Alternative Control Register	0x0000_0000
TIMER4_ALTCTL	TMR45_BA+0x20	R/W	Timer4 Alternative Control Register	0x0000_0000
TIMER5_ALTCTL	TMR45_BA+0x120	R/W	Timer5 Alternative Control Register	0x0000_0000
TIMER6_ALTCTL	TMR67_BA+0x20	R/W	Timer6 Alternative Control Register	0x0000_0000
TIMER7_ALTCTL	TMR67_BA+0x120	R/W	Timer7 Alternative Control Register	0x0000_0000
TIMER8_ALTCTL	TMR89_BA+0x20	R/W	Timer8 Alternative Control Register	0x0000_0000
TIMER9_ALTCTL	TMR89_BA+0x120	R/W	Timer9 Alternative Control Register	0x0000_0000
TIMER10_ALTCTL	TMR1011_BA+0x20	R/W	Timer10 Alternative Control Register	0x0000_0000
TIMER11_ALTCTL	TMR1011_BA+0x120	R/W	Timer11 Alternative Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**FUNCSEL**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**FUNCSEL**	**Function Selection** 0 = Timer controller is used as timer function. 1 = Timer controller is used as PWM function. **Note:** When timer is used as PWM, the clock source of time controller will be forced to PCLKx automatically.

Timer PWM Control Register (TIMERx_PWMCTL)

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMCTL	TMR01_BA+0x40	R/W	Timer0 PWM Control Register	0x0000_0000
TIMER1_PWMCTL	TMR01_BA+0x140	R/W	Timer1 PWM Control Register	0x0000_0000
TIMER2_PWMCTL	TMR23_BA+0x40	R/W	Timer2 PWM Control Register	0x0000_0000
TIMER3_PWMCTL	TMR23_BA+0x140	R/W	Timer3 PWM Control Register	0x0000_0000
TIMER4_PWMCTL	TMR45_BA+0x40	R/W	Timer4 PWM Control Register	0x0000_0000
TIMER5_PWMCTL	TMR45_BA+0x140	R/W	Timer5 PWM Control Register	0x0000_0000
TIMER6_PWMCTL	TMR67_BA+0x40	R/W	Timer6 PWM Control Register	0x0000_0000
TIMER7_PWMCTL	TMR67_BA+0x140	R/W	Timer7 PWM Control Register	0x0000_0000
TIMER8_PWMCTL	TMR89_BA+0x40	R/W	Timer8 PWM Control Register	0x0000_0000
TIMER9_PWMCTL	TMR89_BA+0x140	R/W	Timer9 PWM Control Register	0x0000_0000
TIMER10_PWMCTL	TMR1011_BA+0x40	R/W	Timer10 PWM Control Register	0x0000_0000
TIMER11_PWMCTL	TMR1011_BA+0x140	R/W	Timer11 PWM Control Register	0x0000_0000

31	30	29	28	27	26	25	24
DBGTRIOFF	**DBGHALT**	**Reserved**					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							**OUTMODE**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**IMMLDEN**	**CTRLD**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**CNTMODE**	**CNTTYPE**		**CNTEN**

Bits	Description	
[31]	**DBGTRIOFF**	**ICE Debug Mode Acknowledge Disable Bit (Write Protect)** 0 = ICE debug mode acknowledgement effects PWM output. PWM output pin will be forced as tri-state while ICE debug mode acknowledged. 1 = ICE debug mode acknowledgement disabled. PWM output pin will keep output no matter ICE debug mode acknowledged or not. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register
[30]	**DBGHALT**	**ICE Debug Mode Counter Halt (Write Protect)** If debug mode counter halt is enabled, PWM counter will keep current value until exit ICE debug mode. 0 = ICE debug mode counter halt disable. 1 = ICE debug mode counter halt enable. **Note:** This bit is write protected. Refer to SYS_RLKTZS or SYS_RLKTZND or SYS_RLKSUBM register.

[29:17]	**Reserved**	Reserved.
[16]	**OUTMODE**	**PWM Output Mode** This bit controls the output mode of corresponding PWM channel. 0 = PWM independent mode. 1 = PWM complementary mode.
[15:10]	**Reserved**	Reserved.
[9]	**IMMLDEN**	**Immediately Load Enable Bit** 0 = PERIOD will load to PBUF when current PWM period is completed no matter CTRLD is enabled/disabled. If CTRLD is disabled, CMP will load to CMPBUF when current PWM period is completed; if CTRLD is enabled in up-down count type, CMP will load to CMPBUF at the center point of current period. 1 = PERIOD/CMP will load to PBUF/CMPBUF immediately when user update PERIOD/CMP. **Note:** If IMMLDEN is enabled, CTRLD will be invalid.
[8]	**CTRLD**	**Center Re-load** In up-down count type, PERIOD will load to PBUF when current PWM period is completed always and CMP will load to CMPBUF at the center point of current period.
[7:4]	**Reserved**	Reserved.
[3]	**CNTMODE**	**PWM Counter Mode** 0 = Auto-reload mode. 1 = One-shot mode.
[2:1]	**CNTTYPE**	**PWM Counter Behavior Type** 00 = Up count type. 01 = Down count type. 10 = Up-down count type. 11 = Reserved.
[0]	**CNTEN**	**PWM Counter Enable Bit** 0 = PWM counter and clock prescale Stop Running. 1 = PWM counter and clock prescale Start Running.

Timer PWM Counter Clock Source Register (TIMERx PWMCLKSRC)

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMCLKSRC	TMR01_BA+0x44	R/W	Timer0 PWM Counter Clock Source Register	0x0000_0000
TIMER1_PWMCLKSRC	TMR01_BA+0x144	R/W	Timer1 PWM Counter Clock Source Register	0x0000_0000
TIMER2_PWMCLKSRC	TMR23_BA+0x44	R/W	Timer2 PWM Counter Clock Source Register	0x0000_0000
TIMER3_PWMCLKSRC	TMR23_BA+0x144	R/W	Timer3 PWM Counter Clock Source Register	0x0000_0000
TIMER4_PWMCLKSRC	TMR45_BA+0x44	R/W	Timer4 PWM Counter Clock Source Register	0x0000_0000
TIMER5_PWMCLKSRC	TMR45_BA+0x144	R/W	Timer5 PWM Counter Clock Source Register	0x0000_0000
TIMER6_PWMCLKSRC	TMR67_BA+0x44	R/W	Timer6 PWM Counter Clock Source Register	0x0000_0000
TIMER7_PWMCLKSRC	TMR67_BA+0x144	R/W	Timer7 PWM Counter Clock Source Register	0x0000_0000
TIMER8_PWMCLKSRC	TMR89_BA+0x44	R/W	Timer8 PWM Counter Clock Source Register	0x0000_0000
TIMER9_PWMCLKSRC	TMR89_BA+0x144	R/W	Timer9 PWM Counter Clock Source Register	0x0000_0000
TIMER10_PWMCLKSRC	TMR1011_BA+0x44	R/W	Timer10 PWM Counter Clock Source Register	0x0000_0000
TIMER11_PWMCLKSRC	TMR1011_BA+0x144	R/W	Timer11 PWM Counter Clock Source Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				CLKSRC			

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3:0]	**CLKSRC**	**PWM Counter Clock Source Select** The PWM counter clock source can be selected from TMRx_CLK or internal timer time-out or capture event. 0000 = TMRx_CLK. 0001 = Internal TIMER0 time-out or capture event. 0010 = Internal TIMER1 time-out or capture event. 0011 = Internal TIMER2 time-out or capture event. 0100 = Internal TIMER3 time-out or capture event. 0101 = Internal TIMER4 time-out or capture event. 0110 = Internal TIMER5 time-out or capture event. 0111 = Internal TIMER6 time-out or capture event.

		1000 = Internal TIMER7 time-out or capture event. 1001 = Internal TIMER8 time-out or capture event. 1010 = Internal TIMER9 time-out or capture event. 1011 = Internal TIMER10 time-out or capture event. 1100 = Internal TIMER11 time-out or capture event. Others = Reserved. **Note:** If TIMER0 PWM function is enabled, the PWM counter clock source can be selected from TMR0_CLK, TIMER1 interrupt events, TIMER2 interrupt events, or TIMER3 interrupt events.

Timer PWM Counter Clock Pre-scale Register (TIMERx PWMCLKPSC)

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMCLKPSC	TMR01_BA+0x48	R/W	Timer0 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER1_PWMCLKPSC	TMR01_BA+0x148	R/W	Timer1 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER2_PWMCLKPSC	TMR23_BA+0x48	R/W	Timer2 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER3_PWMCLKPSC	TMR23_BA+0x148	R/W	Timer3 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER4_PWMCLKPSC	TMR45_BA+0x48	R/W	Timer4 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER5_PWMCLKPSC	TMR45_BA+0x148	R/W	Timer5 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER6_PWMCLKPSC	TMR67_BA+0x48	R/W	Timer6 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER7_PWMCLKPSC	TMR67_BA+0x148	R/W	Timer7 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER8_PWMCLKPSC	TMR89_BA+0x48	R/W	Timer8 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER9_PWMCLKPSC	TMR89_BA+0x148	R/W	Timer9 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER10_PWMCLKPSC	TMR1011_BA+0x48	R/W	Timer10 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER11_PWMCLKPSC	TMR1011_BA+0x148	R/W	Timer11 PWM Counter Clock Pre-scale Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				CLKPSC			
7	6	5	4	3	2	1	0
CLKPSC							

Bits	Description	
[31:12]	**Reserved**	Reserved.
[11:0]	**CLKPSC**	**PWM Counter Clock Pre-scale** The active clock of PWM counter is decided by counter clock prescale and divided by (CLKPSC + 1). If CLKPSC is 0, then there is no scaling in PWM counter clock source.

Timer PWM Clear Counter Register (TIMERx PWMCNTCLR)

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMCNTCLR	TMR01_BA+0x4C	R/W	Timer0 PWM Clear Counter Register	0x0000_0000
TIMER1_PWMCNTCLR	TMR01_BA+0x14C	R/W	Timer1 PWM Clear Counter Register	0x0000_0000
TIMER2_PWMCNTCLR	TMR23_BA+0x4C	R/W	Timer2 PWM Clear Counter Register	0x0000_0000
TIMER3_PWMCNTCLR	TMR23_BA+0x14C	R/W	Timer3 PWM Clear Counter Register	0x0000_0000
TIMER4_PWMCNTCLR	TMR45_BA+0x4C	R/W	Timer4 PWM Clear Counter Register	0x0000_0000
TIMER5_PWMCNTCLR	TMR45_BA+0x14C	R/W	Timer5 PWM Clear Counter Register	0x0000_0000
TIMER6_PWMCNTCLR	TMR67_BA+0x4C	R/W	Timer6 PWM Clear Counter Register	0x0000_0000
TIMER7_PWMCNTCLR	TMR67_BA+0x14C	R/W	Timer7 PWM Clear Counter Register	0x0000_0000
TIMER8_PWMCNTCLR	TMR89_BA+0x4C	R/W	Timer8 PWM Clear Counter Register	0x0000_0000
TIMER9_PWMCNTCLR	TMR89_BA+0x14C	R/W	Timer9 PWM Clear Counter Register	0x0000_0000
TIMER10_PWMCNTCLR	TMR1011_BA+0x4C	R/W	Timer10 PWM Clear Counter Register	0x0000_0000
TIMER11_PWMCNTCLR	TMR1011_BA+0x14C	R/W	Timer11 PWM Clear Counter Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							CNTCLR

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**CNTCLR**	**Clear PWM Counter Control Bit** It is automatically cleared by hardware. 0 = No effect. 1 = Clear 16-bit PWM counter to 0x10000 in up and up-down count type and reset counter value to PERIOD in down count type.

Timer PWM Period Register (TIMERx PWMPERIOD)

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMPERIOD	TMR01_BA+0x50	R/W	Timer0 PWM Period Register	0x0000_0000
TIMER1_PWMPERIOD	TMR01_BA+0x150	R/W	Timer1 PWM Period Register	0x0000_0000
TIMER2_PWMPERIOD	TMR23_BA+0x50	R/W	Timer2 PWM Period Register	0x0000_0000
TIMER3_PWMPERIOD	TMR23_BA+0x150	R/W	Timer3 PWM Period Register	0x0000_0000
TIMER4_PWMPERIOD	TMR45_BA+0x50	R/W	Timer4 PWM Period Register	0x0000_0000
TIMER5_PWMPERIOD	TMR45_BA+0x150	R/W	Timer5 PWM Period Register	0x0000_0000
TIMER6_PWMPERIOD	TMR67_BA+0x50	R/W	Timer6 PWM Period Register	0x0000_0000
TIMER7_PWMPERIOD	TMR67_BA+0x150	R/W	Timer7 PWM Period Register	0x0000_0000
TIMER8_PWMPERIOD	TMR89_BA+0x50	R/W	Timer8 PWM Period Register	0x0000_0000
TIMER9_PWMPERIOD	TMR89_BA+0x150	R/W	Timer9 PWM Period Register	0x0000_0000
TIMER10_PWMPERIOD	TMR1011_BA+0x50	R/W	Timer10 PWM Period Register	0x0000_0000
TIMER11_PWMPERIOD	TMR1011_BA+0x150	R/W	Timer11 PWM Period Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PERIOD							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PERIOD							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**PERIOD**	**PWM Period Register** In up count type: PWM counter counts from 0 to PERIOD, and restarts from 0. In down count type: PWM counter counts from PERIOD to 0, and restarts from PERIOD. In up-down count type: PWM counter counts from 0 to PERIOD, then decrements to 0 and repeats again. In up and down count type: PWM period time = (PERIOD + 1) * (CLKPSC + 1) * TMRx_PWMCLK. In up-down count type: PWM period time = 2 * PERIOD * (CLKPSC+ 1) * TMRx_PWMCLK. **Note:** User should take care DIRF (TIMERx_PWMCNT[16]) bit in up/down/up-down count type to monitor current counter direction in each count type.

Timer PWM Comparator Register (TIMERx PWMCMPDAT)

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMCMPDAT	TMR01_BA+0x54	R/W	Timer0 PWM Comparator Register	0x0000_0000
TIMER1_PWMCMPDAT	TMR01_BA+0x154	R/W	Timer1 PWM Comparator Register	0x0000_0000
TIMER2_PWMCMPDAT	TMR23_BA+0x54	R/W	Timer2 PWM Comparator Register	0x0000_0000
TIMER3_PWMCMPDAT	TMR23_BA+0x154	R/W	Timer3 PWM Comparator Register	0x0000_0000
TIMER4_PWMCMPDAT	TMR45_BA+0x54	R/W	Timer4 PWM Comparator Register	0x0000_0000
TIMER5_PWMCMPDAT	TMR45_BA+0x154	R/W	Timer5 PWM Comparator Register	0x0000_0000
TIMER6_PWMCMPDAT	TMR67_BA+0x54	R/W	Timer6 PWM Comparator Register	0x0000_0000
TIMER7_PWMCMPDAT	TMR67_BA+0x154	R/W	Timer7 PWM Comparator Register	0x0000_0000
TIMER8_PWMCMPDAT	TMR89_BA+0x54	R/W	Timer8 PWM Comparator Register	0x0000_0000
TIMER9_PWMCMPDAT	TMR89_BA+0x154	R/W	Timer9 PWM Comparator Register	0x0000_0000
TIMER10_PWMCMPDAT	TMR1011_BA+0x54	R/W	Timer10 PWM Comparator Register	0x0000_0000
TIMER11_PWMCMPDAT	TMR1011_BA+0x154	R/W	Timer11 PWM Comparator Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CMP							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CMP							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**CMP**	**PWM Comparator Register** PWM CMP is used to compare with PWM CNT to generate PWM output waveform, interrupt events and trigger EADC to start conversion.

Timer PWM Dead-time Control Register (TIMERx PWMDTCTL)

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMDTCTL	TMR01_BA+0x58	R/W	Timer0 PWM Dead-time Control Register	0x0000_0000
TIMER1_PWMDTCTL	TMR01_BA+0x158	R/W	Timer1 PWM Dead-time Control Register	0x0000_0000
TIMER2_PWMDTCTL	TMR23_BA+0x58	R/W	Timer2 PWM Dead-time Control Register	0x0000_0000
TIMER3_PWMDTCTL	TMR23_BA+0x158	R/W	Timer3 PWM Dead-time Control Register	0x0000_0000
TIMER4_PWMDTCTL	TMR45_BA+0x58	R/W	Timer4 PWM Dead-time Control Register	0x0000_0000
TIMER5_PWMDTCTL	TMR45_BA+0x158	R/W	Timer5 PWM Dead-time Control Register	0x0000_0000
TIMER6_PWMDTCTL	TMR67_BA+0x58	R/W	Timer6 PWM Dead-time Control Register	0x0000_0000
TIMER7_PWMDTCTL	TMR67_BA+0x158	R/W	Timer7 PWM Dead-time Control Register	0x0000_0000
TIMER8_PWMDTCTL	TMR89_BA+0x58	R/W	Timer8 PWM Dead-time Control Register	0x0000_0000
TIMER9_PWMDTCTL	TMR89_BA+0x158	R/W	Timer9 PWM Dead-time Control Register	0x0000_0000
TIMER10_PWMDTCTL	TMR1011_BA+0x58	R/W	Timer10 PWM Dead-time Control Register	0x0000_0000
TIMER11_PWMDTCTL	TMR1011_BA+0x158	R/W	Timer11 PWM Dead-time Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							DTCKSEL
23	22	21	20	19	18	17	16
Reserved							DTEN
15	14	13	12	11	10	9	8
Reserved				DTCNT			
7	6	5	4	3	2	1	0
DTCNT							

Bits	Description	
[31:25]	**Reserved**	Reserved.
[24]	**DTCKSEL**	**Dead-time Clock Select (Write Protect)** 0 = Dead-time clock source from TMRx_PWMCLK without counter clock prescale. 1 = Dead-time clock source from TMRx_PWMCLK with counter clock prescale. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register
[23:17]	**Reserved**	Reserved.
[16]	**DTEN**	**Enable Dead-time Insertion for PWMx_CH0 and PWMx_CH1 (Write Protect)** Dead-time insertion function is only active when PWM complementary mode is enabled. If dead- time insertion is inactive, the outputs of PWMx_CH0 and PWMx_CH1 are complementary without any delay. 0 = Dead-time insertion Disabled on the pin pair.

		1 = Dead-time insertion Enabled on the pin pair. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register
[15:12]	**Reserved**	Reserved.
[11:0]	**DTCNT**	**Dead-time Counter (Write Protect)** The dead-time can be calculated from the following two formulas: Dead-time = (DTCNT[11:0] + 1) * TMRx_PWMCLK, if DTCKSEL is 0. Dead-time = (DTCNT[11:0] + 1) * TMRx_PWMCLK * (CLKPSC + 1), if DTCKSEL is 1. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register

Timer PWM Counter Register (TIMERx PWMCNT)

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMCNT	TMR01_BA+0x5C	R	Timer0 PWM Counter Register	0x0000_0000
TIMER1_PWMCNT	TMR01_BA+0x15C	R	Timer1 PWM Counter Register	0x0000_0000
TIMER2_PWMCNT	TMR23_BA+0x5C	R	Timer2 PWM Counter Register	0x0000_0000
TIMER3_PWMCNT	TMR23_BA+0x15C	R	Timer3 PWM Counter Register	0x0000_0000
TIMER4_PWMCNT	TMR45_BA+0x5C	R	Timer4 PWM Counter Register	0x0000_0000
TIMER5_PWMCNT	TMR45_BA+0x15C	R	Timer5 PWM Counter Register	0x0000_0000
TIMER6_PWMCNT	TMR67_BA+0x5C	R	Timer6 PWM Counter Register	0x0000_0000
TIMER7_PWMCNT	TMR67_BA+0x15C	R	Timer7 PWM Counter Register	0x0000_0000
TIMER8_PWMCNT	TMR89_BA+0x5C	R	Timer8 PWM Counter Register	0x0000_0000
TIMER9_PWMCNT	TMR89_BA+0x15C	R	Timer9 PWM Counter Register	0x0000_0000
TIMER10_PWMCNT	TMR1011_BA+0x5C	R	Timer10 PWM Counter Register	0x0000_0000
TIMER11_PWMCNT	TMR1011_BA+0x15C	R	Timer11 PWM Counter Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							DIRF
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CNT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNT							

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**DIRF**	**PWM Counter Direction Indicator Flag (Read Only)** 0 = Counter is active in down count. 1 = Counter is active up count.
[15:0]	**CNT**	**PWM Counter Value Register (Read Only)** User can monitor CNT to know the current counter value in 16-bit period counter.

Timer PWM Output Mask Enable Register (TIMERx PWMMSKEN)

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMMSKEN	TMR01_BA+0x60	R/W	Timer0 PWM Output Mask Enable Register	0x0000_0000
TIMER1_PWMMSKEN	TMR01_BA+0x160	R/W	Timer1 PWM Output Mask Enable Register	0x0000_0000
TIMER2_PWMMSKEN	TMR23_BA+0x60	R/W	Timer2 PWM Output Mask Enable Register	0x0000_0000
TIMER3_PWMMSKEN	TMR23_BA+0x160	R/W	Timer3 PWM Output Mask Enable Register	0x0000_0000
TIMER4_PWMMSKEN	TMR45_BA+0x60	R/W	Timer4 PWM Output Mask Enable Register	0x0000_0000
TIMER5_PWMMSKEN	TMR45_BA+0x160	R/W	Timer5 PWM Output Mask Enable Register	0x0000_0000
TIMER6_PWMMSKEN	TMR67_BA+0x60	R/W	Timer6 PWM Output Mask Enable Register	0x0000_0000
TIMER7_PWMMSKEN	TMR67_BA+0x160	R/W	Timer7 PWM Output Mask Enable Register	0x0000_0000
TIMER8_PWMMSKEN	TMR89_BA+0x60	R/W	Timer8 PWM Output Mask Enable Register	0x0000_0000
TIMER9_PWMMSKEN	TMR89_BA+0x160	R/W	Timer9 PWM Output Mask Enable Register	0x0000_0000
TIMER10_PWMMSKEN	TMR1011_BA+0x60	R/W	Timer10 PWM Output Mask Enable Register	0x0000_0000
TIMER11_PWMMSKEN	TMR1011_BA+0x160	R/W	Timer11 PWM Output Mask Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						MSKEN1	MSKEN0

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**MSKEN1**	**PWMx_CH1 Output Mask Enable Bit** The PWMx_CH1 output signal will be masked when this bit is enabled. The PWMx_CH1 will output MSKDAT1 (TIMER_PWMMSK[1]) data. 0 = PWMx_CH1 output signal is non-masked. 1 = PWMx_CH1 output signal is masked and output MSKDAT1 data.
[0]	**MSKEN0**	**PWMx_CH0 Output Mask Enable Bit** The PWMx_CH0 output signal will be masked when this bit is enabled. The PWMx_CH0 will output MSKDAT0 (TIMER_PWMMSK[0]) data. 0 = PWMx_CH0 output signal is non-masked. 1 = PWMx_CH0 output signal is masked and output MSKDAT0 data.

Timer PWM Output Mask Data Control Register (TIMERx PWMMSK)

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMMSK	TMR01_BA+0x64	R/W	Timer0 PWM Output Mask Data Control Register	0x0000_0000
TIMER1_PWMMSK	TMR01_BA+0x164	R/W	Timer1 PWM Output Mask Data Control Register	0x0000_0000
TIMER2_PWMMSK	TMR23_BA+0x64	R/W	Timer2 PWM Output Mask Data Control Register	0x0000_0000
TIMER3_PWMMSK	TMR23_BA+0x164	R/W	Timer3 PWM Output Mask Data Control Register	0x0000_0000
TIMER4_PWMMSK	TMR45_BA+0x64	R/W	Timer4 PWM Output Mask Data Control Register	0x0000_0000
TIMER5_PWMMSK	TMR45_BA+0x164	R/W	Timer5 PWM Output Mask Data Control Register	0x0000_0000
TIMER6_PWMMSK	TMR67_BA+0x64	R/W	Timer6 PWM Output Mask Data Control Register	0x0000_0000
TIMER7_PWMMSK	TMR67_BA+0x164	R/W	Timer7 PWM Output Mask Data Control Register	0x0000_0000
TIMER8_PWMMSK	TMR89_BA+0x64	R/W	Timer8 PWM Output Mask Data Control Register	0x0000_0000
TIMER9_PWMMSK	TMR89_BA+0x164	R/W	Timer9 PWM Output Mask Data Control Register	0x0000_0000
TIMER10_PWMMSK	TMR1011_BA+0x64	R/W	Timer10 PWM Output Mask Data Control Register	0x0000_0000
TIMER11_PWMMSK	TMR1011_BA+0x164	R/W	Timer11 PWM Output Mask Data Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**MSKDAT1**	**MSKDAT0**

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**MSKDAT1**	**PWMx_CH1 Output Mask Data Control Bit** This bit is used to control the output state of PWMx_CH1 pin when PWMx_CH1 output mask function is enabled (MSKEN1 = 1). 0 = Output logic Low to PWMx_CH1. 1 = Output logic High to PWMx_CH1.
[0]	**MSKDAT0**	**PWMx_CH0 Output Mask Data Control Bit** This bit is used to control the output state of PWMx_CH0 pin when PWMx_CH0 output mask function is enabled (MSKEN0 = 1). 0 = Output logic Low to PWMx_CH0. 1 = Output logic High to PWMx_CH0.

Timer PWM Brake Pin Noise Filter Register (TIMERx PWMBNF)

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMBNF	TMR01_BA+0x68	R/W	Timer0 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER1_PWMBNF	TMR01_BA+0x168	R/W	Timer1 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER2_PWMBNF	TMR23_BA+0x68	R/W	Timer2 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER3_PWMBNF	TMR23_BA+0x168	R/W	Timer3 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER4_PWMBNF	TMR45_BA+0x68	R/W	Timer4 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER5_PWMBNF	TMR45_BA+0x168	R/W	Timer5 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER6_PWMBNF	TMR67_BA+0x68	R/W	Timer6 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER7_PWMBNF	TMR67_BA+0x168	R/W	Timer7 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER8_PWMBNF	TMR89_BA+0x68	R/W	Timer8 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER9_PWMBNF	TMR89_BA+0x168	R/W	Timer9 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER10_PWMBNF	TMR1011_BA+0x68	R/W	Timer10 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER11_PWMBNF	TMR1011_BA+0x168	R/W	Timer11 PWM Brake Pin Noise Filter Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved					BKPINSRC		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BRKPINV	BRKFCNT			BRKNFSEL			BRKNFEN

Bits	Description	
[31:19]	**Reserved**	Reserved.
[18:16]	**BKPINSRC**	**Brake Pin Source Select** 000 = Brake pin source comes from PWM0_BRAKE0 pin. 001 = Brake pin source comes from PWM0_BRAKE1 pin. 010 = Brake pin source comes from PWM1_BRAKE0 pin. 011 = Brake pin source comes from PWM1_BRAKE1 pin. 100 = Brake pin source comes from PWM2_BRAKE0 pin. 101 = Brake pin source comes from PWM2_BRAKE1 pin.
[15:8]	**Reserved**	Reserved.
[7]	**BRKPINV**	**Brake Pin Detection Control Bit** 0 = Brake pin event will be detected if EPWMx_BRAKEy pin status transfer from low to high in edge-

		detect, or pin status is high in level-detect. 1 = Brake pin event will be detected if EPWMx_BRAKEy pin status transfer from high to low in edge-detect, or pin status is low in level-detect .
[6:4]	**BRKFCNT**	**Brake Pin Noise Filter Count** The fields is used to control the active noise filter sample time. Once noise filter sample time = (Period time of BRKDBCS) * BRKFCNT.
[3:1]	**BRKNFSEL**	**Brake Pin Noise Filter Clock Selection** 000 = Noise filter clock is PCLKx. 001 = Noise filter clock is PCLKx/2. 010 = Noise filter clock is PCLKx/4. 011 = Noise filter clock is PCLKx/8. 100 = Noise filter clock is PCLKx/16. 101 = Noise filter clock is PCLKx/32. 110 = Noise filter clock is PCLKx/64. 111 = Noise filter clock is PCLKx/128.
[0]	**BRKNFEN**	**Brake Pin Noise Filter Enable Bit** 0 = Pin noise filter detect of EPWMx_BRAKEy Disabled. 1 = Pin noise filter detect of EPWMx_BRAKEy Enabled.

Timer PWM System Fail Brake Control Register (TIMERx PWMFAILBRK)

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMFAILBRK	TMR01_BA+0x6C	R/W	Timer0 PWM System Fail Brake Control Register	0x0000_0000
TIMER1_PWMFAILBRK	TMR01_BA+0x16C	R/W	Timer1 PWM System Fail Brake Control Register	0x0000_0000
TIMER2_PWMFAILBRK	TMR23_BA+0x6C	R/W	Timer2 PWM System Fail Brake Control Register	0x0000_0000
TIMER3_PWMFAILBRK	TMR23_BA+0x16C	R/W	Timer3 PWM System Fail Brake Control Register	0x0000_0000
TIMER4_PWMFAILBRK	TMR45_BA+0x6C	R/W	Timer4 PWM System Fail Brake Control Register	0x0000_0000
TIMER5_PWMFAILBRK	TMR45_BA+0x16C	R/W	Timer5 PWM System Fail Brake Control Register	0x0000_0000
TIMER6_PWMFAILBRK	TMR67_BA+0x6C	R/W	Timer6 PWM System Fail Brake Control Register	0x0000_0000
TIMER7_PWMFAILBRK	TMR67_BA+0x16C	R/W	Timer7 PWM System Fail Brake Control Register	0x0000_0000
TIMER8_PWMFAILBRK	TMR89_BA+0x6C	R/W	Timer8 PWM System Fail Brake Control Register	0x0000_0000
TIMER9_PWMFAILBRK	TMR89_BA+0x16C	R/W	Timer9 PWM System Fail Brake Control Register	0x0000_0000
TIMER10_PWMFAILBRK	TMR1011_BA+0x6C	R/W	Timer10 PWM System Fail Brake Control Register	0x0000_0000
TIMER11_PWMFAILBRK	TMR1011_BA+0x16C	R/W	Timer11 PWM System Fail Brake Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	2	4	3	2	1	0
Reserved				CORBRKEN	Reserved	LVDBRKEN	CSSBRKEN

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3]	**CORBRKEN**	**Core Lockup Detection Trigger PWM Brake Function Enable Bit** 0 = Brake Function triggered by core lockup event Disabled. 1 = Brake Function triggered by core lockup event Enabled.
[2]	**Reserved**	Reserved.
[1]	**LVDBRKEN**	**Low Voltage Interrupt Flag Detection Trigger PWM Brake Function Enable Bit** 0 = Brake Function triggered by LVD interrupt flag event Disabled. 1 = Brake Function triggered by LVD interrupt flag event Enabled.
[0]	**CSSBRKEN**	**Clock Security System Detection Trigger PWM Brake Function Enable Bit** 0 = Brake Function triggered by clock fail detection Disabled.

		1 = Brake Function triggered by clock fail detection Enabled.

Timer PWM Brake Control Register (TIMERx PWMBRKCTL)

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMBRKCTL	TMR01_BA+0x70	R/W	Timer0 PWM Brake Control Register	0x0000_0000
TIMER1_PWMBRKCTL	TMR01_BA+0x170	R/W	Timer1 PWM Brake Control Register	0x0000_0000
TIMER2_PWMBRKCTL	TMR23_BA+0x70	R/W	Timer2 PWM Brake Control Register	0x0000_0000
TIMER3_PWMBRKCTL	TMR23_BA+0x170	R/W	Timer3 PWM Brake Control Register	0x0000_0000
TIMER4_PWMBRKCTL	TMR45_BA+0x70	R/W	Timer4 PWM Brake Control Register	0x0000_0000
TIMER5_PWMBRKCTL	TMR45_BA+0x170	R/W	Timer5 PWM Brake Control Register	0x0000_0000
TIMER6_PWMBRKCTL	TMR67_BA+0x70	R/W	Timer6 PWM Brake Control Register	0x0000_0000
TIMER7_PWMBRKCTL	TMR67_BA+0x170	R/W	Timer7 PWM Brake Control Register	0x0000_0000
TIMER8_PWMBRKCTL	TMR89_BA+0x70	R/W	Timer8 PWM Brake Control Register	0x0000_0000
TIMER9_PWMBRKCTL	TMR89_BA+0x170	R/W	Timer9 PWM Brake Control Register	0x0000_0000
TIMER10_PWMBRKCTL	TMR1011_BA+0x70	R/W	Timer10 PWM Brake Control Register	0x0000_0000
TIMER11_PWMBRKCTL	TMR1011_BA+0x170	R/W	Timer11 PWM Brake Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				BRKAODD		BRKAEVEN	
15	14	13	12	11	10	9	8
SYSLBEN	Reserved		BRKPLEN	Reserved			
7	6	5	4	3	2	1	0
SYSEBEN	Reserved		BRKPEEN	Reserved			

Bits	Description	
[31:20]	**Reserved**	Reserved.
[19:18]	**BRKAODD**	**PWM Brake Action Select for PWMx_CH1 (Write Protect)** 00 = EPWMx_BRAKEy brake event will not affect PWMx_CH1 output. 01 = PWMx_CH1 output tri-state when EPWMx_BRAKEy brake event happened. 10 = PWMx_CH1 output low level when EPWMx_BRAKEy brake event happened. 11 = PWMx_CH1 output high level when EPWMx_BRAKEy brake event happened. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register
[17:16]	**BRKAEVEN**	**PWM Brake Action Select for PWMx_CH0 (Write Protect)** 00 = EPWMx_BRAKEy brake event will not affect PWMx_CH0 output. 01 = PWMx_CH0 output tri-state when EPWMx_BRAKEy brake event happened.

		10 = PWMx_CH0 output low level when EPWMx_BRAKEy brake event happened. 11 = PWMx_CH0 output high level when EPWMx_BRAKEy brake event happened. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register
[15]	**SYSLBEN**	**Enable System Fail As Level-detect Brake Source (Write Protect)** 0 = System fail condition as level-detect brake source Disabled. 1 = System fail condition as level-detect brake source Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register
[14:13]	**Reserved**	Reserved.
[12]	**BRKPLEN**	**Enable TM_BRAKEx Pin As Level-detect Brake Source (Write Protect)** 0 = EPWMx_BRAKEy pin event as level-detect brake source Disabled. 1 = EPWMx_BRAKEy pin event as level-detect brake source Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register
[11:8]	**Reserved**	Reserved.
[7]	**SYSEBEN**	**Enable System Fail As Edge-detect Brake Source (Write Protect)** 0 = System fail condition as edge-detect brake source Disabled. 1 = System fail condition as edge-detect brake source Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register
[6:5]	**Reserved**	Reserved.
[4]	**BRKPEEN**	**Enable TM_BRAKEx Pin As Edge-detect Brake Source (Write Protect)** 0 = EPWMx_BRAKEy pin event as edge-detect brake source Disabled. 1 = EPWMx_BRAKEy pin event as edge-detect brake source Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register
[3:0]	**Reserved**	Reserved.

Timer PWM Pin Output Polar Control Register (TIMERx PWMPOLCTL)

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMPOLCTL	TMR01_BA+0x74	R/W	Timer0 PWM Pin Output Polar Control Register	0x0000_0000
TIMER1_PWMPOLCTL	TMR01_BA+0x174	R/W	Timer1 PWM Pin Output Polar Control Register	0x0000_0000
TIMER2_PWMPOLCTL	TMR23_BA+0x74	R/W	Timer2 PWM Pin Output Polar Control Register	0x0000_0000
TIMER3_PWMPOLCTL	TMR23_BA+0x174	R/W	Timer3 PWM Pin Output Polar Control Register	0x0000_0000
TIMER4_PWMPOLCTL	TMR45_BA+0x74	R/W	Timer4 PWM Pin Output Polar Control Register	0x0000_0000
TIMER5_PWMPOLCTL	TMR45_BA+0x174	R/W	Timer5 PWM Pin Output Polar Control Register	0x0000_0000
TIMER6_PWMPOLCTL	TMR67_BA+0x74	R/W	Timer6 PWM Pin Output Polar Control Register	0x0000_0000
TIMER7_PWMPOLCTL	TMR67_BA+0x174	R/W	Timer7 PWM Pin Output Polar Control Register	0x0000_0000
TIMER8_PWMPOLCTL	TMR89_BA+0x74	R/W	Timer8 PWM Pin Output Polar Control Register	0x0000_0000
TIMER9_PWMPOLCTL	TMR89_BA+0x174	R/W	Timer9 PWM Pin Output Polar Control Register	0x0000_0000
TIMER10_PWMPOLCTL	TMR1011_BA+0x74	R/W	Timer10 PWM Pin Output Polar Control Register	0x0000_0000
TIMER11_PWMPOLCTL	TMR1011_BA+0x174	R/W	Timer11 PWM Pin Output Polar Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						PINV1	PINV0

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**PINV1**	**PWMx_CH1 Output Pin Polar Control Bit** The bit is used to control polarity state of PWMx_CH1 output pin. 0 = PWMx_CH1 output pin polar inverse Disabled. 1 = PWMx_CH1 output pin polar inverse Enabled.
[0]	**PINV0**	**PWMx_CH0 Output Pin Polar Control Bit** The bit is used to control polarity state of PWMx_CH0 output pin. 0 = PWMx_CH0 output pin polar inverse Disabled. 1 = PWMx_CH0 output pin polar inverse Enabled.

Timer PWM Pin Output Enable Register (TIMERx PWMPOEN)

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMPOEN	TMR01_BA+0x78	R/W	Timer0 PWM Pin Output Enable Register	0x0000_0000
TIMER1_PWMPOEN	TMR01_BA+0x178	R/W	Timer1 PWM Pin Output Enable Register	0x0000_0000
TIMER2_PWMPOEN	TMR23_BA+0x78	R/W	Timer2 PWM Pin Output Enable Register	0x0000_0000
TIMER3_PWMPOEN	TMR23_BA+0x178	R/W	Timer3 PWM Pin Output Enable Register	0x0000_0000
TIMER4_PWMPOEN	TMR45_BA+0x78	R/W	Timer4 PWM Pin Output Enable Register	0x0000_0000
TIMER5_PWMPOEN	TMR45_BA+0x178	R/W	Timer5 PWM Pin Output Enable Register	0x0000_0000
TIMER6_PWMPOEN	TMR67_BA+0x78	R/W	Timer6 PWM Pin Output Enable Register	0x0000_0000
TIMER7_PWMPOEN	TMR67_BA+0x178	R/W	Timer7 PWM Pin Output Enable Register	0x0000_0000
TIMER8_PWMPOEN	TMR89_BA+0x78	R/W	Timer8 PWM Pin Output Enable Register	0x0000_0000
TIMER9_PWMPOEN	TMR89_BA+0x178	R/W	Timer9 PWM Pin Output Enable Register	0x0000_0000
TIMER10_PWMPOEN	TMR1011_BA+0x78	R/W	Timer10 PWM Pin Output Enable Register	0x0000_0000
TIMER11_PWMPOEN	TMR1011_BA+0x178	R/W	Timer11 PWM Pin Output Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**POEN1**	**POEN0**

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**POEN1**	**PWMx_CH1 Output Pin Enable Bit** 0 = PWMx_CH1 pin at tri-state mode. 1 = PWMx_CH1 pin in output mode.
[0]	**POEN0**	**PWMx_CH0 Output Pin Enable Bit** 0 = PWMx_CH0 pin at tri-state mode. 1 = PWMx_CH0 pin in output mode.

Timer PWM Software Trigger Brake Control Register (TIMERx PWMSWBRK)

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMSWBRK	TMR01_BA+0x7C	W	Timer0 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER1_PWMSWBRK	TMR01_BA+0x17C	W	Timer1 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER2_PWMSWBRK	TMR23_BA+0x7C	W	Timer2 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER3_PWMSWBRK	TMR23_BA+0x17C	W	Timer3 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER4_PWMSWBRK	TMR45_BA+0x7C	W	Timer4 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER5_PWMSWBRK	TMR45_BA+0x17C	W	Timer5 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER6_PWMSWBRK	TMR67_BA+0x7C	W	Timer6 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER7_PWMSWBRK	TMR67_BA+0x17C	W	Timer7 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER8_PWMSWBRK	TMR89_BA+0x7C	W	Timer8 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER9_PWMSWBRK	TMR89_BA+0x17C	W	Timer9 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER10_PWMSWBRK	TMR1011_BA+0x7C	W	Timer10 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER11_PWMSWBRK	TMR1011_BA+0x17C	W	Timer11 PWM Software Trigger Brake Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							BRKLTRG
7	6	5	4	3	2	1	0
Reserved							BRKETRG

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8]	**BRKLTRG**	**Software Trigger Level-detect Brake Source (Write Only) (Write Protect)** Write 1 to this bit will trigger PWM level-detect brake source, then BRKLIF0 and BRKLIF1 will set to 1 automatically in TIMERx_PWMINTSTS1 register. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register
[7:1]	**Reserved**	Reserved.
[0]	**BRKETRG**	**Software Trigger Edge-detect Brake Source (Write Only) (Write Protect)** Write 1 to this bit will trigger PWM edge-detect brake source, then BRKEIF0 and BRKEIF1 will set to 1 automatically in TIMERx_PWMINTSTS1 register. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register

Timer PWM Interrupt Enable Register 0 (TIMERx PWMINTEN0)

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMINTEN0	TMR01_BA+0x80	R/W	Timer0 PWM Interrupt Enable Register 0	0x0000_0000
TIMER1_PWMINTEN0	TMR01_BA+0x180	R/W	Timer1 PWM Interrupt Enable Register 0	0x0000_0000
TIMER2_PWMINTEN0	TMR23_BA+0x80	R/W	Timer2 PWM Interrupt Enable Register 0	0x0000_0000
TIMER3_PWMINTEN0	TMR23_BA+0x180	R/W	Timer3 PWM Interrupt Enable Register 0	0x0000_0000
TIMER4_PWMINTEN0	TMR45_BA+0x80	R/W	Timer4 PWM Interrupt Enable Register 0	0x0000_0000
TIMER5_PWMINTEN0	TMR45_BA+0x180	R/W	Timer5 PWM Interrupt Enable Register 0	0x0000_0000
TIMER6_PWMINTEN0	TMR67_BA+0x80	R/W	Timer6 PWM Interrupt Enable Register 0	0x0000_0000
TIMER7_PWMINTEN0	TMR67_BA+0x180	R/W	Timer7 PWM Interrupt Enable Register 0	0x0000_0000
TIMER8_PWMINTEN0	TMR89_BA+0x80	R/W	Timer8 PWM Interrupt Enable Register 0	0x0000_0000
TIMER9_PWMINTEN0	TMR89_BA+0x180	R/W	Timer9 PWM Interrupt Enable Register 0	0x0000_0000
TIMER10_PWMINTEN0	TMR1011_BA+0x80	R/W	Timer10 PWM Interrupt Enable Register 0	0x0000_0000
TIMER11_PWMINTEN0	TMR1011_BA+0x180	R/W	Timer11 PWM Interrupt Enable Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				CMPDIEN	CMPUIEN	PIEN	ZIEN

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3]	**CMPDIEN**	**PWM Compare Down Count Interrupt Enable Bit** 0 = Compare down count interrupt Disabled. 1 = Compare down count interrupt Enabled.
[2]	**CMPUIEN**	**PWM Compare Up Count Interrupt Enable Bit** 0 = Compare up count interrupt Disabled. 1 = Compare up count interrupt Enabled.
[1]	**PIEN**	**PWM Period Point Interrupt Enable Bit** 0 = Period point interrupt Disabled. 1 = Period point interrupt Enabled.

		Note: In up-down count type, period point means the center point of current PWM period.
[0]	**ZIEN**	**PWM Zero Point Interrupt Enable Bit** 0 = Zero point interrupt Disabled. 1 = Zero point interrupt Enabled.

Timer PWM Interrupt Enable Register 1 (TIMERx PWMINTEN1)

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMINTEN1	TMR01_BA+0x84	R/W	Timer0 PWM Interrupt Enable Register 1	0x0000_0000
TIMER1_PWMINTEN1	TMR01_BA+0x184	R/W	Timer1 PWM Interrupt Enable Register 1	0x0000_0000
TIMER2_PWMINTEN1	TMR23_BA+0x84	R/W	Timer2 PWM Interrupt Enable Register 1	0x0000_0000
TIMER3_PWMINTEN1	TMR23_BA+0x184	R/W	Timer3 PWM Interrupt Enable Register 1	0x0000_0000
TIMER4_PWMINTEN1	TMR45_BA+0x84	R/W	Timer4 PWM Interrupt Enable Register 1	0x0000_0000
TIMER5_PWMINTEN1	TMR45_BA+0x184	R/W	Timer5 PWM Interrupt Enable Register 1	0x0000_0000
TIMER6_PWMINTEN1	TMR67_BA+0x84	R/W	Timer6 PWM Interrupt Enable Register 1	0x0000_0000
TIMER7_PWMINTEN1	TMR67_BA+0x184	R/W	Timer7 PWM Interrupt Enable Register 1	0x0000_0000
TIMER8_PWMINTEN1	TMR89_BA+0x84	R/W	Timer8 PWM Interrupt Enable Register 1	0x0000_0000
TIMER9_PWMINTEN1	TMR89_BA+0x184	R/W	Timer9 PWM Interrupt Enable Register 1	0x0000_0000
TIMER10_PWMINTEN1	TMR1011_BA+0x84	R/W	Timer10 PWM Interrupt Enable Register 1	0x0000_0000
TIMER11_PWMINTEN1	TMR1011_BA+0x184	R/W	Timer11 PWM Interrupt Enable Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**BRKLIEN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**BRKEIEN**

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8]	**BRKLIEN**	**PWM Level-detect Brake Interrupt Enable Bit** 0 = PWM level-detect brake interrupt Disabled. 1 = PWM level-detect brake interrupt Enabled.
[7:1]	**Reserved**	Reserved.
[0]	**BRKEIEN**	**PWM Edge-detect Brake Interrupt Enable Bit** 0 = PWM edge-detect brake interrupt Disabled. 1 = PWM edge-detect brake interrupt Enabled.

Timer PWM Interrupt Status Register 0 (TIMERx PWMINTSTS0)

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMINTSTS0	TMR01_BA+0x88	R/W	Timer0 PWM Interrupt Status Register 0	0x0000_0000
TIMER1_PWMINTSTS0	TMR01_BA+0x188	R/W	Timer1 PWM Interrupt Status Register 0	0x0000_0000
TIMER2_PWMINTSTS0	TMR23_BA+0x88	R/W	Timer2 PWM Interrupt Status Register 0	0x0000_0000
TIMER3_PWMINTSTS0	TMR23_BA+0x188	R/W	Timer3 PWM Interrupt Status Register 0	0x0000_0000
TIMER4_PWMINTSTS0	TMR45_BA+0x88	R/W	Timer4 PWM Interrupt Status Register 0	0x0000_0000
TIMER5_PWMINTSTS0	TMR45_BA+0x188	R/W	Timer5 PWM Interrupt Status Register 0	0x0000_0000
TIMER6_PWMINTSTS0	TMR67_BA+0x88	R/W	Timer6 PWM Interrupt Status Register 0	0x0000_0000
TIMER7_PWMINTSTS0	TMR67_BA+0x188	R/W	Timer7 PWM Interrupt Status Register 0	0x0000_0000
TIMER8_PWMINTSTS0	TMR89_BA+0x88	R/W	Timer8 PWM Interrupt Status Register 0	0x0000_0000
TIMER9_PWMINTSTS0	TMR89_BA+0x188	R/W	Timer9 PWM Interrupt Status Register 0	0x0000_0000
TIMER10_PWMINTSTS0	TMR1011_BA+0x88	R/W	Timer10 PWM Interrupt Status Register 0	0x0000_0000
TIMER11_PWMINTSTS0	TMR1011_BA+0x188	R/W	Timer11 PWM Interrupt Status Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**CMPDIF**	**CMPUIF**	**PIF**	**ZIF**

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3]	**CMPDIF**	**PWM Compare Down Count Interrupt Flag** This bit is set by hardware when TIMERx_PWM counter in down count direction and reaches CMP. **Note 1:** If CMP equal to PERIOD, there is no CMPDIF flag in down count type. **Note 2:** This bit is cleared by writing 1 to it.
[2]	**CMPUIF**	**PWM Compare Up Count Interrupt Flag** This bit is set by hardware when TIMERx_PWM counter in up count direction and reaches CMP. **Note 1:** If CMP equal to PERIOD, there is no CMPUIF flag in up count type and up-down count type. **Note 2:** This bit is cleared by writing 1 to it.
[1]	**PIF**	**PWM Period Point Interrupt Flag**

		This bit is set by hardware when TIMERx_PWM counter reaches PERIOD. **Note 1:** In up-down count type, PIF flag means the center point flag of current PWM period. **Note 2:** This bit is cleared by writing 1 to it.
[0]	**ZIF**	**PWM Zero Point Interrupt Flag** This bit is set by hardware when TIMERx_PWM counter reaches 0. **Note:** This bit is cleared by writing 1 to it.

Timer PWM Interrupt Status Register 1 (TIMERx PWMINTSTS1)

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMINTSTS1	TMR01_BA+0x8C	R/W	Timer0 PWM Interrupt Status Register 1	0x0000_0000
TIMER1_PWMINTSTS1	TMR01_BA+0x18C	R/W	Timer1 PWM Interrupt Status Register 1	0x0000_0000
TIMER2_PWMINTSTS1	TMR23_BA+0x8C	R/W	Timer2 PWM Interrupt Status Register 1	0x0000_0000
TIMER3_PWMINTSTS1	TMR23_BA+0x18C	R/W	Timer3 PWM Interrupt Status Register 1	0x0000_0000
TIMER4_PWMINTSTS1	TMR45_BA+0x8C	R/W	Timer4 PWM Interrupt Status Register 1	0x0000_0000
TIMER5_PWMINTSTS1	TMR45_BA+0x18C	R/W	Timer5 PWM Interrupt Status Register 1	0x0000_0000
TIMER6_PWMINTSTS1	TMR67_BA+0x8C	R/W	Timer6 PWM Interrupt Status Register 1	0x0000_0000
TIMER7_PWMINTSTS1	TMR67_BA+0x18C	R/W	Timer7 PWM Interrupt Status Register 1	0x0000_0000
TIMER8_PWMINTSTS1	TMR89_BA+0x8C	R/W	Timer8 PWM Interrupt Status Register 1	0x0000_0000
TIMER9_PWMINTSTS1	TMR89_BA+0x18C	R/W	Timer9 PWM Interrupt Status Register 1	0x0000_0000
TIMER10_PWMINTSTS1	TMR1011_BA+0x8C	R/W	Timer10 PWM Interrupt Status Register 1	0x0000_0000
TIMER11_PWMINTSTS1	TMR1011_BA+0x18C	R/W	Timer11 PWM Interrupt Status Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						BRKLSTS1	BRKLSTS0
23	22	21	20	19	18	17	16
Reserved						BRKESTS1	BRKESTS0
15	14	13	12	11	10	9	8
Reserved						BRKLIF1	BRKLIF0
7	6	5	4	3	2	1	0
Reserved						BRKEIF1	BRKEIF0

Bits	Description	
[31:26]	**Reserved**	Reserved.
[25]	**BRKLSTS1**	**Level-detect Brake Status of PWMx_CH1 (Read Only)** 0 = PWMx_CH1 level-detect brake state is released. 1 = PWMx_CH1 at level-detect brake state. **Note:** If TIMERx_PWM level-detect brake source has released, both PWMx_CH0 and PWMx_CH1 will release brake state when current PWM period finished and resume PWMx_CH0 and PWMx_CH1 output waveform start from next full PWM period.
[24]	**BRKLSTS0**	**Level-detect Brake Status of PWMx_CH0 (Read Only)** 0 = PWMx_CH0 level-detect brake state is released. 1 = PWMx_CH0 at level-detect brake state. **Note:** If TIMERx_PWM level-detect brake source has released, both PWMx_CH0 and PWMx_CH1 will release brake state when current PWM period finished and resume PWMx_CH0 and PWMx_CH1 output

		waveform start from next full PWM period.
[23:18]	**Reserved**	Reserved.
[17]	**BRKESTS1**	**Edge-detect Brake Status of PWMx_CH1 (Read Only)** 0 = PWMx_CH1 edge-detect brake state is released. 1 = PWMx_CH1 at edge-detect brake state. **Note:** User can set BRKEIF1 1 to clear BRKEIF1 flag and PWMx_CH1 will release brake state when current PWM period finished and resume PWMx_CH1 output waveform start from next full PWM period.
[16]	**BRKESTS0**	**Edge -detect Brake Status of PWMx_CH0 (Read Only)** 0 = PWMx_CH0 edge-detect brake state is released. 1 = PWMx_CH0 at edge-detect brake state. **Note:** User can set BRKEIF0 1 to clear BRKEIF0 flag and PWMx_CH0 will release brake state when current PWM period finished and resume PWMx_CH0 output waveform start from next full PWM period.
[15:10]	**Reserved**	Reserved.
[9]	**BRKLIF1**	**Level-detect Brake Interrupt Flag on PWMx_CH1 (Write Protect)** 0 = PWMx_CH1 level-detect brake event do not happened. 1 = PWMx_CH1 level-detect brake event happened. **Note 1:** This bit is cleared by writing 1 to it. **Note 2:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register
[8]	**BRKLIF0**	**Level-detect Brake Interrupt Flag on PWMx_CH0 (Write Protect)** 0 = PWMx_CH0 level-detect brake event do not happened. 1 = PWMx_CH0 level-detect brake event happened. **Note 1:** This bit is cleared by writing 1 to it. **Note 2:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register
[7:2]	**Reserved**	Reserved.
[1]	**BRKEIF1**	**Edge-detect Brake Interrupt Flag PWMx_CH1 (Write Protect)** 0 = PWMx_CH1 edge-detect brake event do not happened. 1 = PWMx_CH1 edge-detect brake event happened. **Note 1:** This bit is cleared by writing 1 to it. **Note 2:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register
[0]	**BRKEIF0**	**Edge-detect Brake Interrupt Flag on PWMx_CH0 (Write Protect)** 0 = PWMx_CH0 edge-detect brake event do not happened. 1 = PWMx_CH0 edge-detect brake event happened. **Note 1:** This bit is cleared by writing 1 to it. **Note 2:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register

Timer PWM EADC Trigger Control Register (TIMERx PWMEADCTS)

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMEADCTS	TMR01_BA+0x90	R/W	Timer0 PWM EADC Trigger Source Select Register	0x0000_0000
TIMER1_PWMEADCTS	TMR01_BA+0x190	R/W	Timer1 PWM EADC Trigger Source Select Register	0x0000_0000
TIMER2_PWMEADCTS	TMR23_BA+0x90	R/W	Timer2 PWM EADC Trigger Source Select Register	0x0000_0000
TIMER3_PWMEADCTS	TMR23_BA+0x190	R/W	Timer3 PWM EADC Trigger Source Select Register	0x0000_0000
TIMER4_PWMEADCTS	TMR45_BA+0x90	R/W	Timer4 PWM EADC Trigger Source Select Register	0x0000_0000
TIMER5_PWMEADCTS	TMR45_BA+0x190	R/W	Timer5 PWM EADC Trigger Source Select Register	0x0000_0000
TIMER6_PWMEADCTS	TMR67_BA+0x90	R/W	Timer6 PWM EADC Trigger Source Select Register	0x0000_0000
TIMER7_PWMEADCTS	TMR67_BA+0x190	R/W	Timer7 PWM EADC Trigger Source Select Register	0x0000_0000
TIMER8_PWMEADCTS	TMR89_BA+0x90	R/W	Timer8 PWM EADC Trigger Source Select Register	0x0000_0000
TIMER9_PWMEADCTS	TMR89_BA+0x190	R/W	Timer9 PWM EADC Trigger Source Select Register	0x0000_0000
TIMER10_PWMEADCTS	TMR1011_BA+0x90	R/W	Timer10 PWM EADC Trigger Source Select Register	0x0000_0000
TIMER11_PWMEADCTS	TMR1011_BA+0x190	R/W	Timer11 PWM EADC Trigger Source Select Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
TRGEN	Reserved			TRGSEL			

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**TRGEN**	**PWM Counter Event Trigger EADC Conversion Enable Bit** 0 = PWM counter event trigger EADC conversion Disabled. 1 = PWM counter event trigger EADC conversion Enabled.
[6:3]	**Reserved**	Reserved.
[2:0]	**TRGSEL**	**PWM Counter Event Source Select to Trigger EADC Conversion** 000 = Trigger EADC conversion at zero point (ZIF). 001 = Trigger EADC conversion at period point (PIF). 010 = Trigger EADC conversion at zero or period point (ZIF or PIF). 011 = Trigger EADC conversion at compare up count point (CMPUIF).

		100 = Trigger EADC conversion at compare down count point (CMPDIF). Others = Reserved.

Timer PWM Synchronous Control Register (TIMERx PWMSCTL)

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMSCTL	TMR01_BA+0x94	R/W	Timer0 PWM Synchronous Control Register	0x0000_0000
TIMER1_PWMSCTL	TMR01_BA+0x194	R/W	Timer1 PWM Synchronous Control Register	0x0000_0000
TIMER2_PWMSCTL	TMR23_BA+0x94	R/W	Timer2 PWM Synchronous Control Register	0x0000_0000
TIMER3_PWMSCTL	TMR23_BA+0x194	R/W	Timer3 PWM Synchronous Control Register	0x0000_0000
TIMER4_PWMSCTL	TMR45_BA+0x94	R/W	Timer4 PWM Synchronous Control Register	0x0000_0000
TIMER5_PWMSCTL	TMR45_BA+0x194	R/W	Timer5 PWM Synchronous Control Register	0x0000_0000
TIMER6_PWMSCTL	TMR67_BA+0x94	R/W	Timer6 PWM Synchronous Control Register	0x0000_0000
TIMER7_PWMSCTL	TMR67_BA+0x194	R/W	Timer7 PWM Synchronous Control Register	0x0000_0000
TIMER8_PWMSCTL	TMR89_BA+0x94	R/W	Timer8 PWM Synchronous Control Register	0x0000_0000
TIMER9_PWMSCTL	TMR89_BA+0x194	R/W	Timer9 PWM Synchronous Control Register	0x0000_0000
TIMER10_PWMSCTL	TMR1011_BA+0x94	R/W	Timer10 PWM Synchronous Control Register	0x0000_0000
TIMER11_PWMSCTL	TMR1011_BA+0x194	R/W	Timer11 PWM Synchronous Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					SYNCSRC		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						SYNCMODE	

Bits	Description	
[31:11]	**Reserved**	Reserved.
[10:8]	**SYNCSRC**	**PWM Synchronous Counter Start/Clear Source Select** 000 = Counter synchronous start/clear by trigger TIMER0_PWMSTRG STRGEN. 001 = Counter synchronous start/clear by trigger TIMER2_PWMSTRG STRGEN. 010 = Counter synchronous start/clear by trigger TIMER4_PWMSTRG STRGEN. 011 = Counter synchronous start/clear by trigger TIMER6_PWMSTRG STRGEN. 100 = Counter synchronous start/clear by trigger TIMER8_PWMSTRG STRGEN. 101 = Counter synchronous start/clear by trigger TIMER10_PWMSTRG STRGEN. **Note 1:** If TIMER0/1/2/3 PWM counter synchronous source are from TIMER0, TIMER0_PWMSCTL[8], TIMER1_PWMSCTL[8], TIMER2_PWMSCTL[8] and TIMER3_PWMSCTL[8] should be 0. **Note 2:** If TIMER0/1/ PWM counter synchronous source are from TIMER0, TIMER0_PWMSCTL[8] and TIMER1_PWMSCTL[8] should be set 0, and TIMER2/3/ PWM counter synchronous source are from

		TIMER2, TIME2_PWMSCTL[8] and TIMER3_PWMSCTL[8] should be set 1.
[7:2]	**Reserved**	Reserved.
[1:0]	**SYNCMODE**	**PWM Synchronous Mode Enable Select** 00 = PWM synchronous function Disabled. 01 = PWM synchronous counter start function Enabled. 10 = Reserved. 11 = PWM synchronous counter clear function Enabled.

Timer PWM Synchronous Trigger Register (TIMERx PWMSTRG)

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMSTRG	TMR01_BA+0x98	W	Timer0 PWM Synchronous Trigger Register	0x0000_0000
TIMER2_PWMSTRG	TMR23_BA+0x98	W	Timer2 PWM Synchronous Trigger Register	0x0000_0000
TIMER4_PWMSTRG	TMR45_BA+0x98	W	Timer4 PWM Synchronous Trigger Register	0x0000_0000
TIMER6_PWMSTRG	TMR67_BA+0x98	W	Timer6 PWM Synchronous Trigger Register	0x0000_0000
TIMER8_PWMSTRG	TMR89_BA+0x98	W	Timer8 PWM Synchronous Trigger Register	0x0000_0000
TIMER10_PWMSTRG	TMR1011_BA+0x98	W	Timer10 PWM Synchronous Trigger Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**STRGEN**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**STRGEN**	**PWM Counter Synchronous Trigger Enable Bit (Write Only)** PMW counter synchronous function is used to make selected PWM channels (include TIMER0/1/2/3 PWM, TIMER0/1 PWM and TIMER2/3 PWM) start counting or clear counter at the same time according to TIMERx_PWMSCTL setting. **Note:** This bit is only available in TIMER0 and TIMER2.

Timer PWM Status Register (TIMERx PWMSTATUS)

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMSTATUS	TMR01_BA+0x9C	R/W	Timer0 PWM Status Register	0x0000_0000
TIMER1_PWMSTATUS	TMR01_BA+0x19C	R/W	Timer1 PWM Status Register	0x0000_0000
TIMER2_PWMSTATUS	TMR23_BA+0x9C	R/W	Timer2 PWM Status Register	0x0000_0000
TIMER3_PWMSTATUS	TMR23_BA+0x19C	R/W	Timer3 PWM Status Register	0x0000_0000
TIMER4_PWMSTATUS	TMR45_BA+0x9C	R/W	Timer4 PWM Status Register	0x0000_0000
TIMER5_PWMSTATUS	TMR45_BA+0x19C	R/W	Timer5 PWM Status Register	0x0000_0000
TIMER6_PWMSTATUS	TMR67_BA+0x9C	R/W	Timer6 PWM Status Register	0x0000_0000
TIMER7_PWMSTATUS	TMR67_BA+0x19C	R/W	Timer7 PWM Status Register	0x0000_0000
TIMER8_PWMSTATUS	TMR89_BA+0x9C	R/W	Timer8 PWM Status Register	0x0000_0000
TIMER9_PWMSTATUS	TMR89_BA+0x19C	R/W	Timer9 PWM Status Register	0x0000_0000
TIMER10_PWMSTATUS	TMR1011_BA+0x9C	R/W	Timer10 PWM Status Register	0x0000_0000
TIMER11_PWMSTATUS	TMR1011_BA+0x19C	R/W	Timer11 PWM Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							EADCTRGF
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							CNTMAXF

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**EADCTRGF**	**Trigger EADC Start Conversion Flag** 0 = PWM counter event trigger EADC start conversion is not occurred. 1 = PWM counter event trigger EADC start conversion has occurred. **Note:** This bit is cleared by writing 1 to it.
[15:1]	**Reserved**	Reserved.
[0]	**CNTMAXF**	**PWM Counter Equal to 0xFFFF Flag** 0 = The PWM counter value never reached its maximum value 0xFFFF. 1 = The PWM counter value has reached its maximum value. **Note:** This bit is cleared by writing 1 to it.

Timer PWM Period Buffer Register (TIMERx PWMPBUF)

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMPBUF	TMR01_BA+0xA0	R	Timer0 PWM Period Buffer Register	0x0000_0000
TIMER1_PWMPBUF	TMR01_BA+0x1A0	R	Timer1 PWM Period Buffer Register	0x0000_0000
TIMER2_PWMPBUF	TMR23_BA+0xA0	R	Timer2 PWM Period Buffer Register	0x0000_0000
TIMER3_PWMPBUF	TMR23_BA+0x1A0	R	Timer3 PWM Period Buffer Register	0x0000_0000
TIMER4_PWMPBUF	TMR45_BA+0xA0	R	Timer4 PWM Period Buffer Register	0x0000_0000
TIMER5_PWMPBUF	TMR45_BA+0x1A0	R	Timer5 PWM Period Buffer Register	0x0000_0000
TIMER6_PWMPBUF	TMR67_BA+0xA0	R	Timer6 PWM Period Buffer Register	0x0000_0000
TIMER7_PWMPBUF	TMR67_BA+0x1A0	R	Timer7 PWM Period Buffer Register	0x0000_0000
TIMER8_PWMPBUF	TMR89_BA+0xA0	R	Timer8 PWM Period Buffer Register	0x0000_0000
TIMER9_PWMPBUF	TMR89_BA+0x1A0	R	Timer9 PWM Period Buffer Register	0x0000_0000
TIMER10_PWMPBUF	TMR1011_BA+0xA0	R	Timer10 PWM Period Buffer Register	0x0000_0000
TIMER11_PWMPBUF	TMR1011_BA+0x1A0	R	Timer11 PWM Period Buffer Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PBUF							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PBUF							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**PBUF**	**PWM Period Buffer Register (Read Only)** Used as PERIOD active register.

Timer PWM Comparator Buffer Register (TIMERx PWMCMPBUF)

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMCMPBUF	TMR01_BA+0xA4	R	Timer0 PWM Comparator Buffer Register	0x0000_0000
TIMER1_PWMCMPBUF	TMR01_BA+0x1A4	R	Timer1 PWM Comparator Buffer Register	0x0000_0000
TIMER2_PWMCMPBUF	TMR23_BA+0xA4	R	Timer2 PWM Comparator Buffer Register	0x0000_0000
TIMER3_PWMCMPBUF	TMR23_BA+0x1A4	R	Timer3 PWM Comparator Buffer Register	0x0000_0000
TIMER4_PWMCMPBUF	TMR45_BA+0xA4	R	Timer4 PWM Comparator Buffer Register	0x0000_0000
TIMER5_PWMCMPBUF	TMR45_BA+0x1A4	R	Timer5 PWM Comparator Buffer Register	0x0000_0000
TIMER6_PWMCMPBUF	TMR67_BA+0xA4	R	Timer6 PWM Comparator Buffer Register	0x0000_0000
TIMER7_PWMCMPBUF	TMR67_BA+0x1A4	R	Timer7 PWM Comparator Buffer Register	0x0000_0000
TIMER8_PWMCMPBUF	TMR89_BA+0xA4	R	Timer8 PWM Comparator Buffer Register	0x0000_0000
TIMER9_PWMCMPBUF	TMR89_BA+0x1A4	R	Timer9 PWM Comparator Buffer Register	0x0000_0000
TIMER10_PWMCMPBUF	TMR1011_BA+0xA4	R	Timer10 PWM Comparator Buffer Register	0x0000_0000
TIMER11_PWMCMPBUF	TMR1011_BA+0x1A4	R	Timer11 PWM Comparator Buffer Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CMPBUF							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CMPBUF							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**CMPBUF**	**PWM Comparator Buffer Register (Read Only)** Used as CMP active register.

6.18 EPWM Generator and Capture Timer (EPWM)

6.18.1 Overview

The chip provides three EPWM generators — EPWM0, EPWM1 and EPWM2. Each EPWM supports 6 channels of EPWM output or input capture. There is a 12-bit prescaler to supports flexible clock to the 16-bit EPWM counter with 16-bit comparator. The EPWM counter supports up, down and up-down counter types. EPWM uses comparator compared with counter to generate events. These events use to generate EPWM pulse, interrupt and trigger signal for EADC to start conversion.

The EPWM generator supports two standard EPWM output modes: Independent mode and Complementary mode, they have difference architecture. There are two output functions based on standard output modes: Group function and Synchronous function. Group function can be enabled under Independent mode or complementary mode. Synchronous function only enabled under complementary mode. Complementary mode has two comparators to generate various EPWM pulse with 12-bit dead-time generator and another free trigger comparator to generate trigger signal for EADC. For EPWM output control unit, it supports polarity output, independent pin mask and brake functions.

The EPWM generator also supports input capture function. It supports latch EPWM counter value to corresponding register when input channel has a rising transition, falling transition or both transition happened. Capture function also supports PDMA to transfer captured data to memory.

6.18.2 Features

6.18.2.1 EPWM Function Features

- Supports maximum clock frequency up to maximum PLL frequency
- Supports up to three EPWM modules, each module provides 6 output channels
- Supports independent mode for EPWM output/Capture input channel
- Supports complementary mode for 3 complementary paired EPWM output channel
  - Dead-time insertion with 12-bit resolution
  - Synchronous function for phase control
  - Two compared values during one period
- Supports 12-bit prescaler from 1 to 4096
- Supports 16-bit resolution EPWM counter
  - Up, down and up/down counter operation type
- Supports one-shot or auto-reload counter operation mode
- Supports group function
- Supports mask function and tri-state enable for each EPWM pin
- Supports trigger EADC on the following events:
  - EPWM counter matches 0, period value or compared value
  - EPWM counter matches free trigger comparator compared value (only for EADC)
  - Supports EPWM trigger EADC event prescaler feature
- Supports EPWM output accumulator stop counter mode
- Supports Fault Detect Function.
- Supports brake function
  - Brake source from pin and system safety events (clock failed, LVDIF detection and

CPU lockup).
  - Noise filter for brake source from pin
  - Edge detect brake source to control brake state until brake interrupt cleared
  - Level detect brake source to auto recover function after brake condition removed
- Supports synchronous function
- Supports interrupt on the following events:
  - EPWM counter matches 0, period value or compared value
  - Brake condition happened

6.18.2.2 Capture Function Features

- Supports up to 12 capture input channels with 16-bit resolution
- Supports rising or falling capture condition
- Supports input rising/falling capture interrupt
- Supports rising/falling capture with counter reload option
- Supports PDMA transfer function for all EPWM channels

6.18.3 Block Diagram

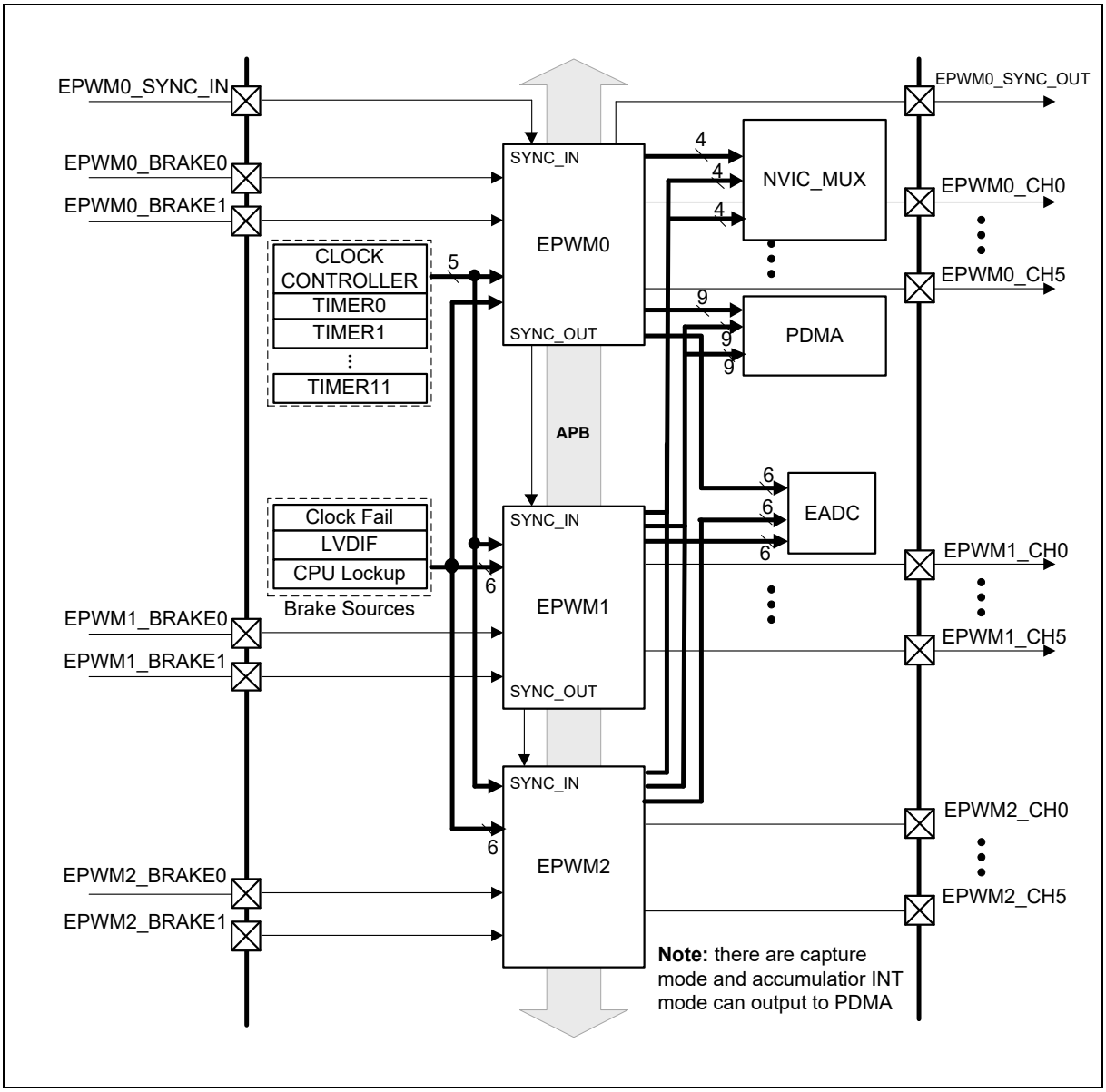
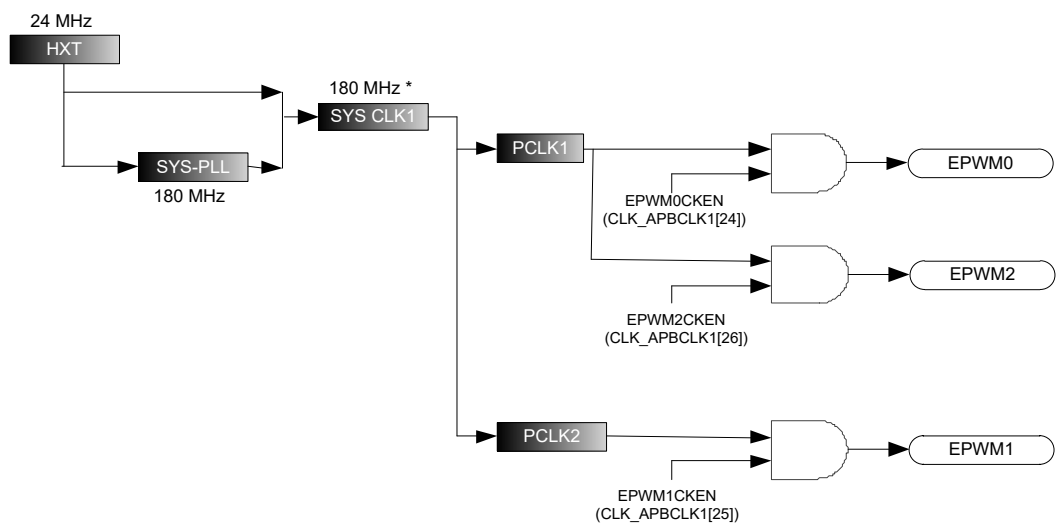


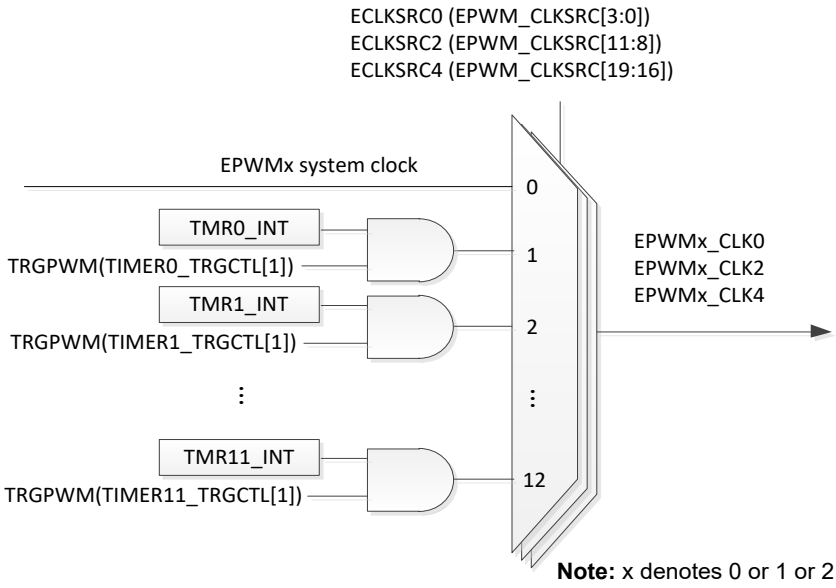
Figure 6.18-1 EPWM Generator Overview Block Diagram

The EPWM0 Clock frequency will be set equal to PCLK1 frequency, EPWM1 Clock frequency will be set equal to PCLK2 frequency and EPWM2 Clock frequency will be set equal to PCLK1 frequency as shown in Figure 6.18-2.. Each EPWM generator has three clock source inputs, each clock source can be selected from EPWM Clock or twelve TIMER trigger EPWM outputs as Figure 6.18-3 by ECLKSRC0 (EPWM_CLKSRC[3:0]) for EPWM_CLK0, ECLKSRC2 (EPWM_CLKSRC[11:8]) for EPWM_CLK2 and ECLKSRC4 (EPWM_CLKSRC[19:16]) for EPWM_CLK4. If the clock source of EPWM counter is selected from TIMERn interrupt events, the TRGPWM (TIMERn_TRGCTL[1], n=0,1..11) bit must be set as 1.



Note *: SYS CLK1 clock source is from SYS-PLL in this condition

Figure 6.18-2 EPWM Clock Source Control



Note: x denotes 0 or 1 or 2

Figure 6.18-3 EPWM Clock Source Control

Figure 6.18-4 and Figure 6.18-5 illustrate the architecture of EPWM independent mode and complementary mode. No matter independent mode or complementary mode, paired channels' (EPWM_CH0 and EPWM_CH1, EPWM_CH2 and EPWM_CH3, EPWM_CH4 and EPWM_CH5) counters both come from the same clock source and prescaler. When the counter counts to 0, PERIOD (EPWM_PERIODn[15:0]) or equal to comparator, events will be generated. These events are passed to corresponding generators to generate EPWM pulse, interrupt signal and trigger signal for EADC to start conversion. Output control is used to change EPWM pulse output state; brake function in output

control also generates interrupt events. In complementary mode, synchronize function is available and even channel uses odd channel comparator to generate events, free trigger comparator events only used to generate trigger EADC signals.

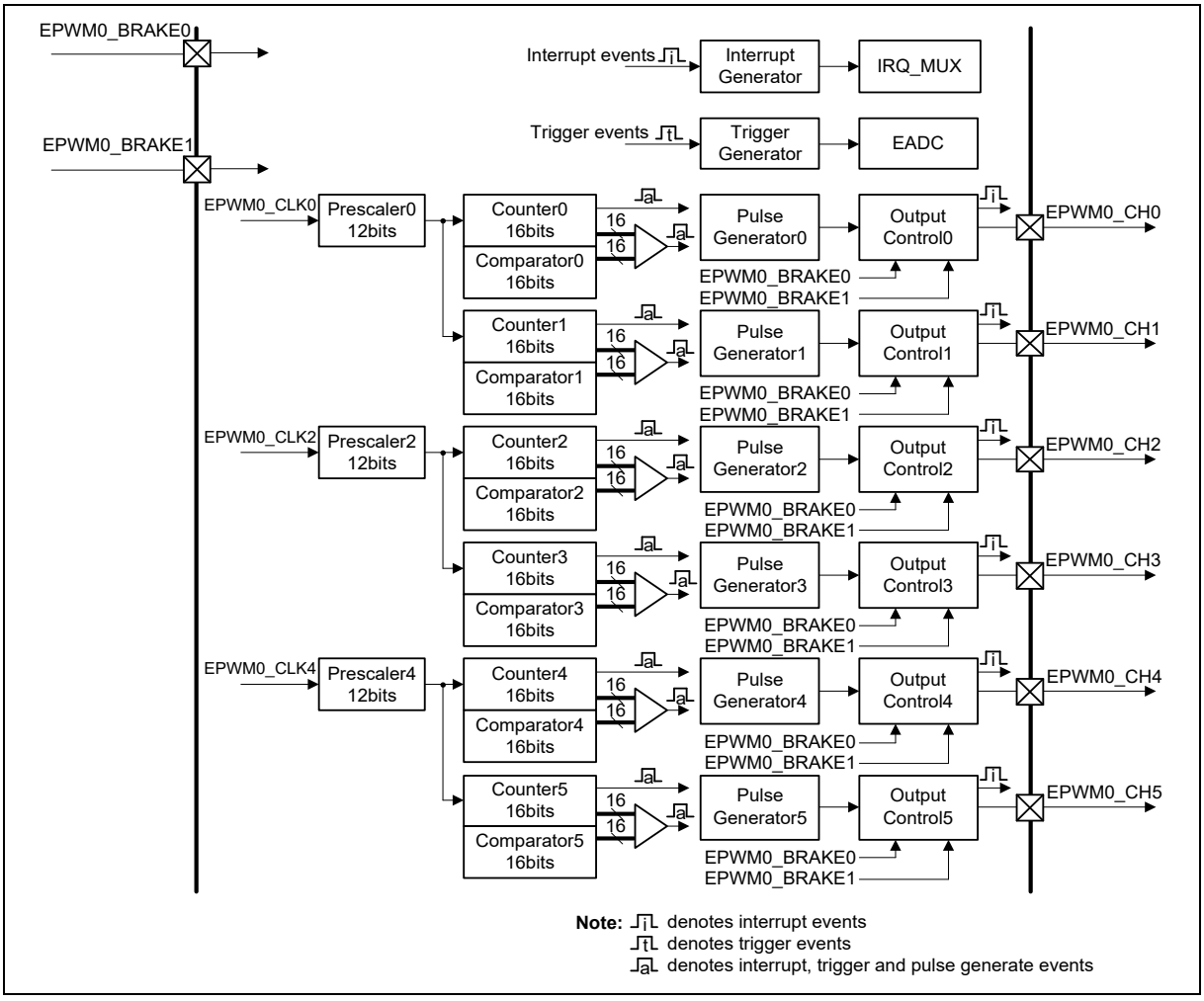


Figure 6.18-4 EPWM Independent Mode Architecture Diagram

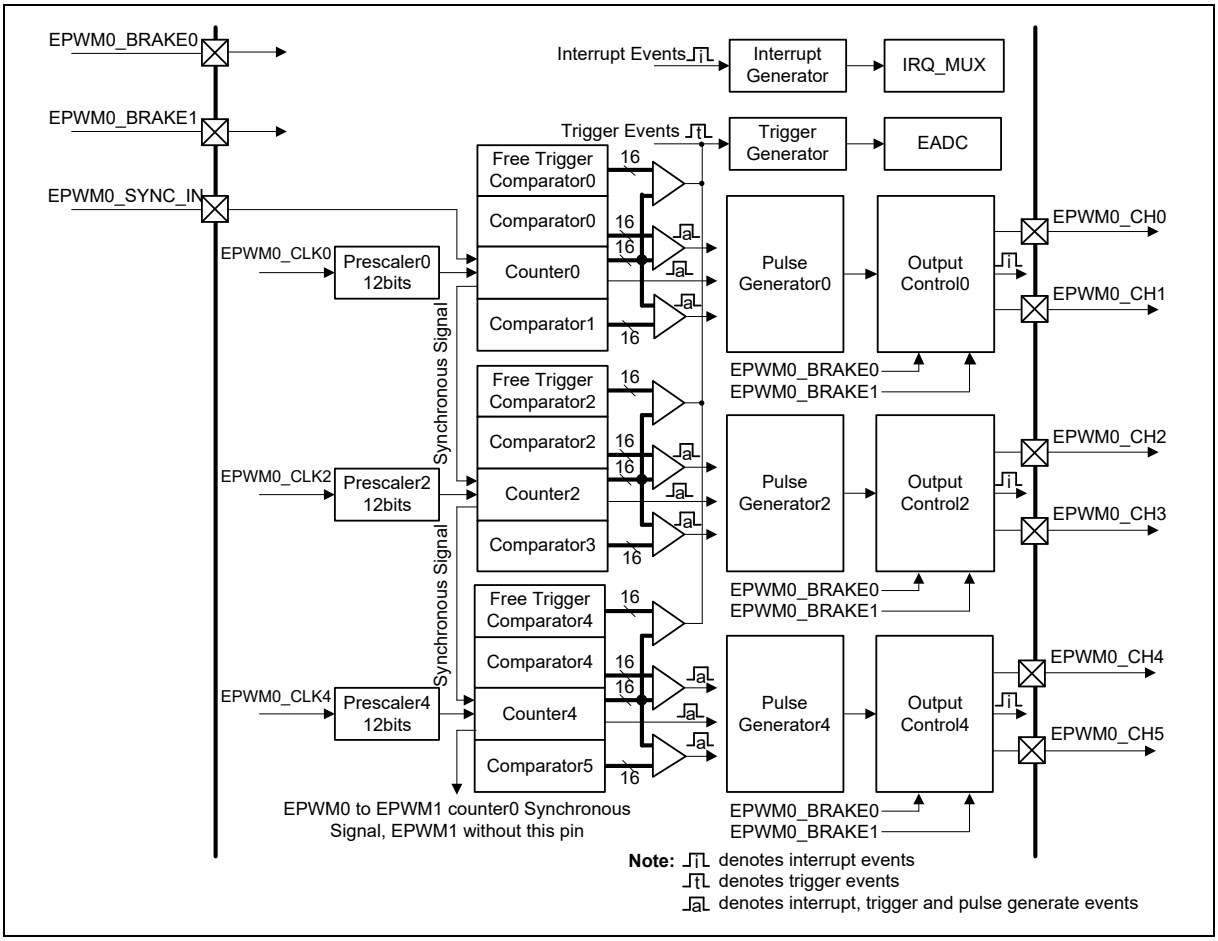


Figure 6.18-5 EPWM Complementary Mode Architecture Diagram

6.18.4 Basic Configuration

6.18.4.1 EPWM0 Basic Configuration

- Clock Source Configuration
  - Enable EPWM0 peripheral clock in EPWM0CKEN (CLK_APBCLK1[24])
- Reset Configuration
  - Reset EPWM0 in EPWM0RST (SYS_IPRST2[16])
- Pin Configuration

Group	Pin Name	GPIO	MFP
EPWM0	EPWM0_BRAKE0	PD.12, PI.6, PD.8	MFP1
	EPWM0_BRAKE1	PD.13, PI.7, PD.9	MFP1
	EPWM0_CH0	PG.0, PI.0, PL.6	MFP1
		PD.6	MFP8
	EPWM0_CH1	PL.15, PI.1, PL.7	MFP1
		PD.7	MFP8
	EPWM0_CH2	PL.14, PA.15, PI.2, PL.8	MFP1

		PD.8	MFP8
		PL.12	MFP12
	EPWM0_CH3	PG.1, PI.3, PL.9	MFP1
		PD.9	MFP8
		PL.13	MFP12
	EPWM0_CH4	PG.2, PI.4, PL.10	MFP1
		PL.8	MFP6
		PD.10	MFP8
	EPWM0_CH5	PG.3, PI.5, PL.11	MFP1
		PL.10	MFP6
		PD.11	MFP8
	EPWM0_SYNC_IN	PL.12, PD.14, PK.0, PD.6	MFP1
	EPWM0_SYNC_OUT	PL.13, PD.15, PK.1, PD.7	MFP1

6.18.4.2 EPWM1 Basic Configuration

- Clock Source Configuration
  - Enable EPWM1 peripheral clock in EPWM1CKEN (CLK_APBCLK1[25])
- Reset Configuration
  - Reset EPWM1 in EPWM1RST (SYS_IPRST2[17])
- Pin Configuration

Group	Pin Name	GPIO	MFP
EPWM1	EPWM1_BRAKE0	PM.14, PD.10, PJ.0	MFP1
	EPWM1_BRAKE1	PM.15, PD.11, PJ.1	MFP1
	EPWM1_CH0	PG.4, PK.2, PL.0, PK.8	MFP1
	EPWM1_CH1	PG.5, PK.3, PL.1, PK.5	MFP1
	EPWM1_CH2	PG.6, PJ.12, PL.2, PM.10, PK.6	MFP1
	EPWM1_CH3	PG.7, PJ.13, PL.3, PM.11, PK.7	MFP1
	EPWM1_CH4	PM.12, PJ.14, PL.4, PJ.2, PG.8	MFP1
		PL.9	MFP6
	EPWM1_CH5	PM.13, PJ.15, PL.5, PJ.3, PG.9	MFP1
		PL.11	MFP6

6.18.4.3 EPWM2 Basic Configuration

- Clock Source Configuration
  - Enable EPWM2 peripheral clock in EPWM2CKEN (CLK_APBCLK1[26])
- Reset Configuration
  - Reset EPWM2 in EPWM2RST (SYS_IPRST3[20])

- Pin Configuration

Group	Pin Name	GPIO	MFP
EPWM2	EPWM2_BRAKE0	PF.14, PB.8	MFP1
	EPWM2_BRAKE1	PB.11	MFP1
	EPWM2_CH0	PK.12, PB.12	MFP1
	EPWM2_CH1	PK.13, PB.13	MFP1
	EPWM2_CH2	PB.14, PK.14	MFP1
	EPWM2_CH3	PB.15, PK.15	MFP1
	EPWM2_CH4	PN.15, PB.9	MFP1
	EPWM2_CH5	PB.10	MFP1

6.18.5 Functional Description

6.18.5.1 EPWM Prescaler

The EPWM prescaler is used to divide clock source, prescaler counting CLKPSC +1 times, EPWM counter only count once. The prescale double buffer is setting by CLKPSC (EPWM_CLKPSCn[11:0], n = 0, 2, 4) bits. Figure 6.18-6 is an example of EPWM channel 0 prescale waveform. The prescale counter will reload CLKPSC at the beginning of the next prescale counter down-count.

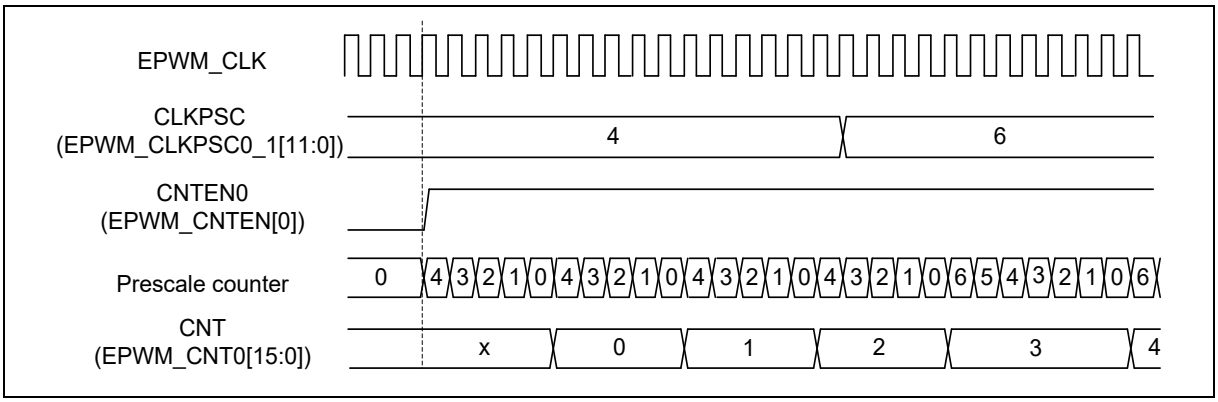


Figure 6.18-6 EPWM_CH0 Prescaler Waveform in Up Counter Type

6.18.5.2 EPWM Counter

The EPWM supports three counter types operation: Up Counter, Down Counter and Up-Down Counter types.

For EPWM channel0, CNT(EPWM_CNT0[15:0]) can be cleared to 0x00 by CNTCLR0 (EPWM_CNTCLR[0]). CNT will be cleared when the prescale counter counts to 0, and CNTCLR0 will be set 0 by hardware automatically.

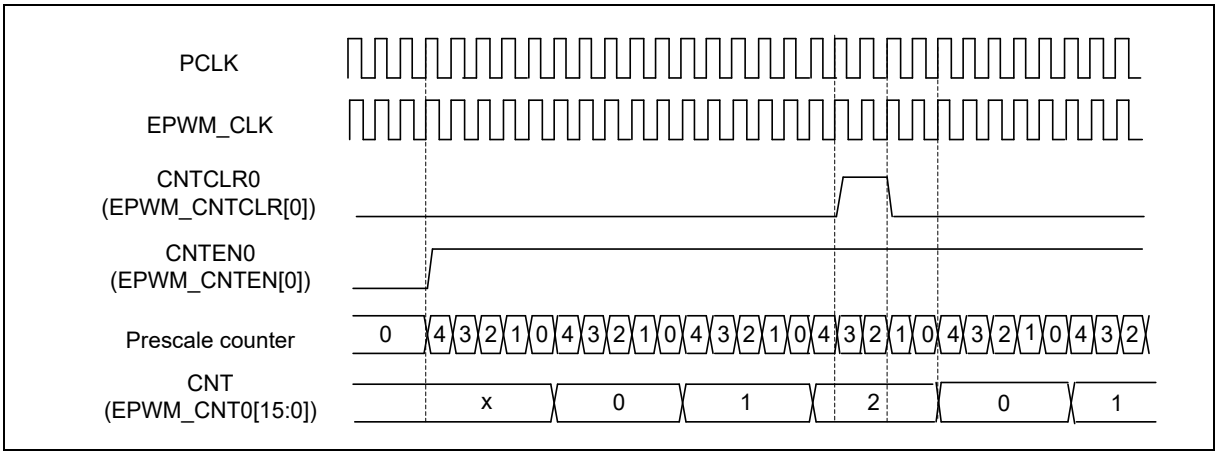


Figure 6.18-7 EPWM Counter Waveform when Setting Clear Counter

6.18.5.3 Up Counter Type

When EPWM counter is set to up counter type, CNTTYPEn (EPWM_CTL1[2n+1:2n], n = 0,1..5) is 0x0, it starts up-counting from 0 to PERIOD (EPWM_PERIODn[15:0], where n denotes channel number) to complete a EPWM period. The current counter value can be read from CNT (EPWM_CNTn[15:0]) bits. The EPWM generates zero point event when the counter counts to 0 and prescale counts to 0. EPWM generates period point event when the counter counts to PERIOD and prescale counts to 0. Figure 6.18-8 shows an example of up counter, wherein

EPWM period time = (PERIOD+1) *(CLKPSC+1)* EPWMx_CLK.

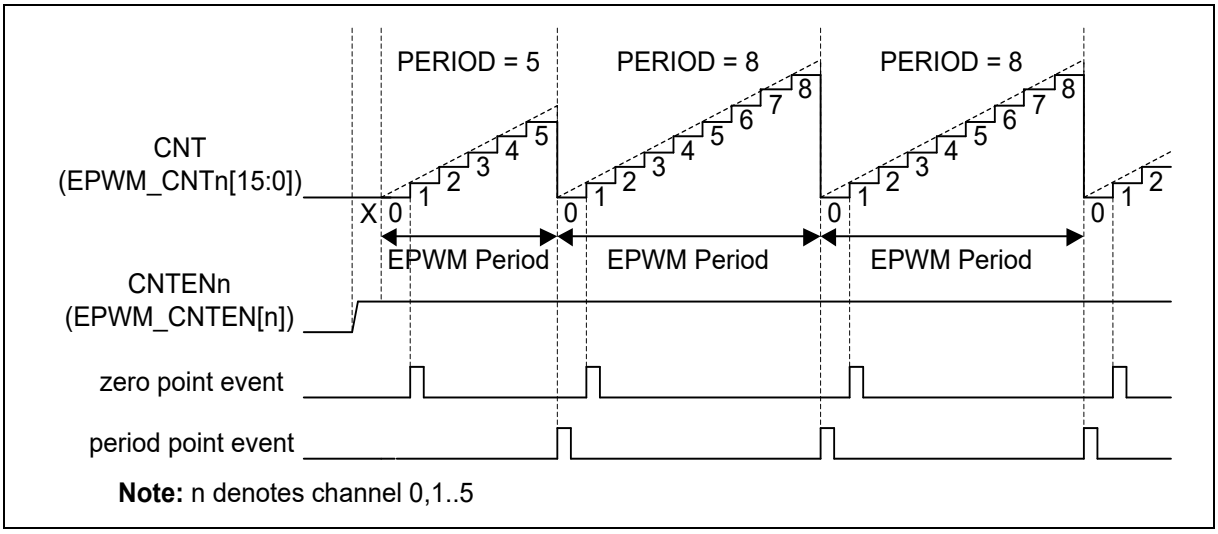


Figure 6.18-8 EPWM Up Counter Type

6.18.5.4 Down Counter Type

When EPWM counter is set to down counter type, CNTTYPEn (EPWM_CTL1[2n+1:2n], n = 0,1..5) is 0x1, it starts down-counting from PERIOD to 0 to complete a EPWM period. The current counter value can be read from CNT (EPWM_CNTn[15:0]) bits. EPWM generates zero point event when the counter counts to 0 and prescale counts to 0. EPWM generates period point event when the counter counts to PERIOD and prescale counts to 0. Figure 6.18-9 shows an example of down counter, wherein

EPWM period time = (PERIOD+1) * (CLKPSC+1) * EPWMx_CLK.

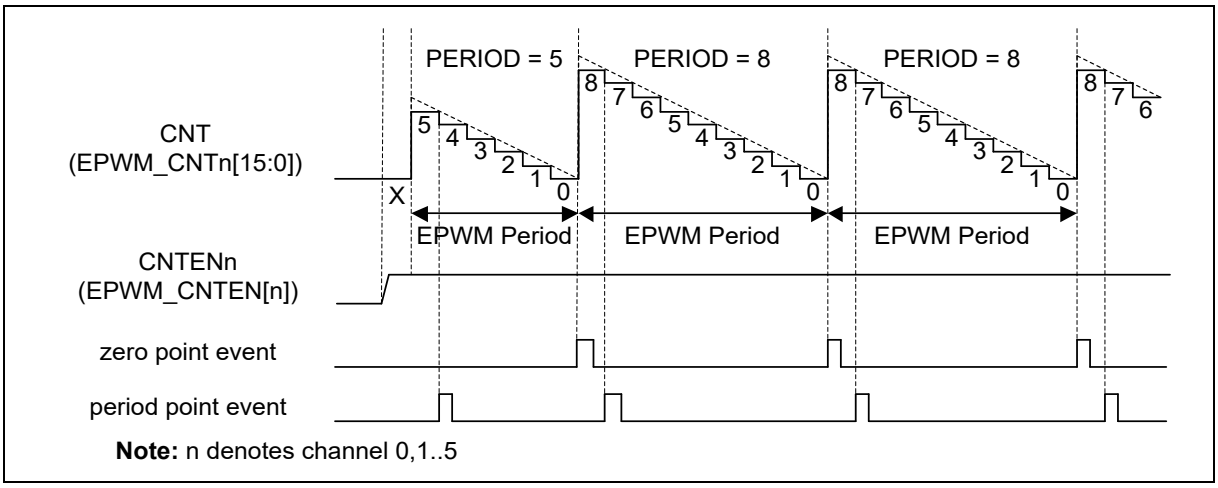


Figure 6.18-9 EPWM Down Counter Type

6.18.5.5 Up-Down Counter Type

When EPWM counter is set to up-down count type, CNTTYPEn (EPWM_CTL1[2n+1:2n], n = 0,1..5) is 0x2, it starts counting-up from 0 to PERIOD and then starts counting down to 0 to complete a EPWM period. The current counter value can be read from CNT (EPWM_CNTn[15:0]) bits. EPWM generates zero point event when the counter counts to 0 and prescale counts to 0. EPWM generates center point event which is equal to period point event when the counter counts to PERIOD. Figure 6.18-10 shows an example of up-down counter, wherein

EPWM period time =(2*PERIOD) * (CLKPSC+1) * EPWMx_CLK.

The DIRF (EPWM_CNTn[16]) bit is counter direction indicator flag, where high is up counting, and low is down counting.

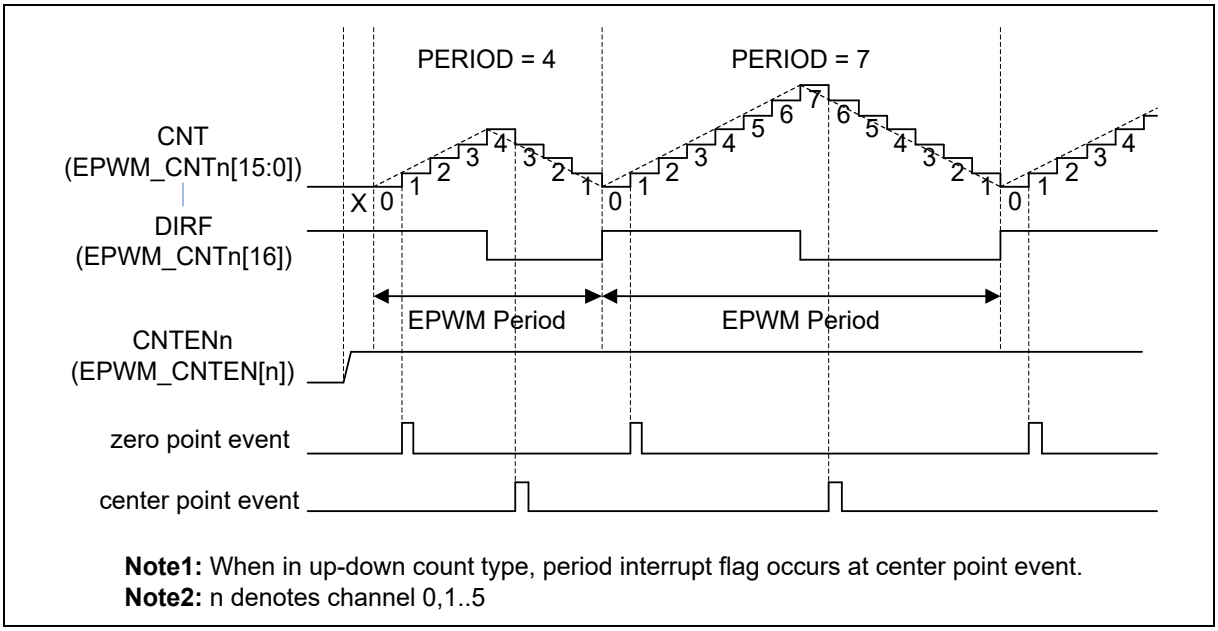
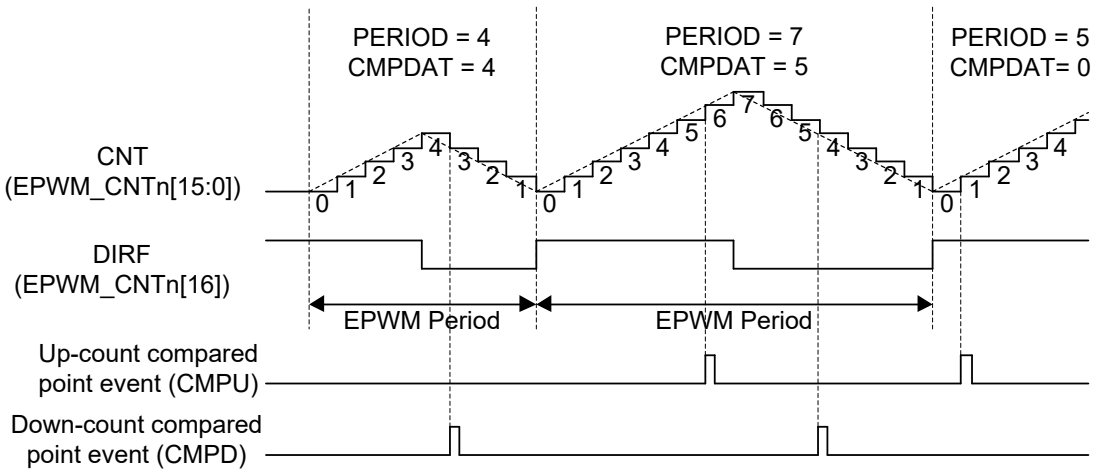


Figure 6.18-10 EPWM Up-Down Counter Type

6.18.5.6 EPWM Comparator

There are two kinds of comparator registers: one is CMPDATn(n = 0,1..5), and the other is FTCMPDATn_m(n = 0,2,4, m = 1,3,5) register. CMPDATn is a basic comparator register of EPWM channel n; In Independent mode each channel only has one comparator, the value of CMPDATn register is continuously compared to the corresponding channel's counter value. In Complementary mode each paired channels has two comparators, and the value of CMPDATn and CMPDATm (n = 0,2,4, m = 1,3,5) registers are continuously compared to the complementary even channel's counter value, because of odd channel's counter is useless. For example, channel 0 and channel 1 are complementary channels, in Complementary mode, channel 1's comparator is continuously compared to channel 0's counter, but not channel 1's. When the counter is equal to value of CMPDAT0 register, EPWM generates a compared point event and uses the event to generate EPWM pulse, interrupt or uses to trigger EADC. In up-down counter type, two events will be generated in an EPWM period as shown in Figure 6.18-11. The CMPU is up count compared point event and CMPD is down count compared point event.



Note1: No CMPU event occurred when CMPDAT equals to PERIOD.
Note2: n denotes channel 0,1..5

Figure 6.18-11 EPWM Compared point Events in Up-Down Counter Type

FTCMPDAT is a free trigger comparator register. Each complementary paired channel only supports one free trigger comparator. The value of FTCMPDATn_m(n = 0,2,4, m = 1,3,5) register is continuously compared to even channel's counter value. When counter is equal to the value of FTCMPDAT register, FTCMDn (EPWM_FTCI[10:8], n=0,2,4) indicator is set in down count type and FTCMUn (EPWM_FTCI[2:0], n=0,2,4) indicator is set in up count type. In addition, EPWM generates an event and only uses to trigger EADC.

6.18.5.7 EPWM Double Buffering

The double buffering uses double buffers to separate software writing and hardware action operation timing. There are four loading modes for loading values to buffer: period loading mode, immediately loading mode, window loading mode and center loading mode. After registers are modified through software, hardware will load register value to the buffer register according to the loading mode timing. The hardware action is based on the buffer value. This can prevent asynchronously operation problem due to software and hardware asynchronism.

The EPWM provides PBUF (EPWM_PBUFn[15:0]) as the active PERIOD buffer register, CMPBUF (EPWM_CMPBUFn[15:0]) as the active CMPDAT buffer register, FTCMPBUF (EPWM_FTCMPBUFn_m[15:0]) as the active FTCMPDAT buffer register and CPSCBUF (EPWM_CPSCBUFn_m[15:0]) as the active CLKPSC buffer register. The concept of double buffering is used in loading modes, which are described in the following sections. For example, as shown Figure 6.18-12, in period loading mode, writing PERIOD, CMPDAT and FTCMPDAT registers through software, EPWM will load new values to their buffer PBUF (EPWM_PBUFn[15:0]), CMPBUF (EPWM_CMPBUFn[15:0]) and FTCMPBUF (EPWM_FTCBUF[15:0]) at start of the next period without affecting the current period counter operation. FTCMPU denotes up-count free trigger compared point event and FTCMPD denotes down-count free trigger compared event.

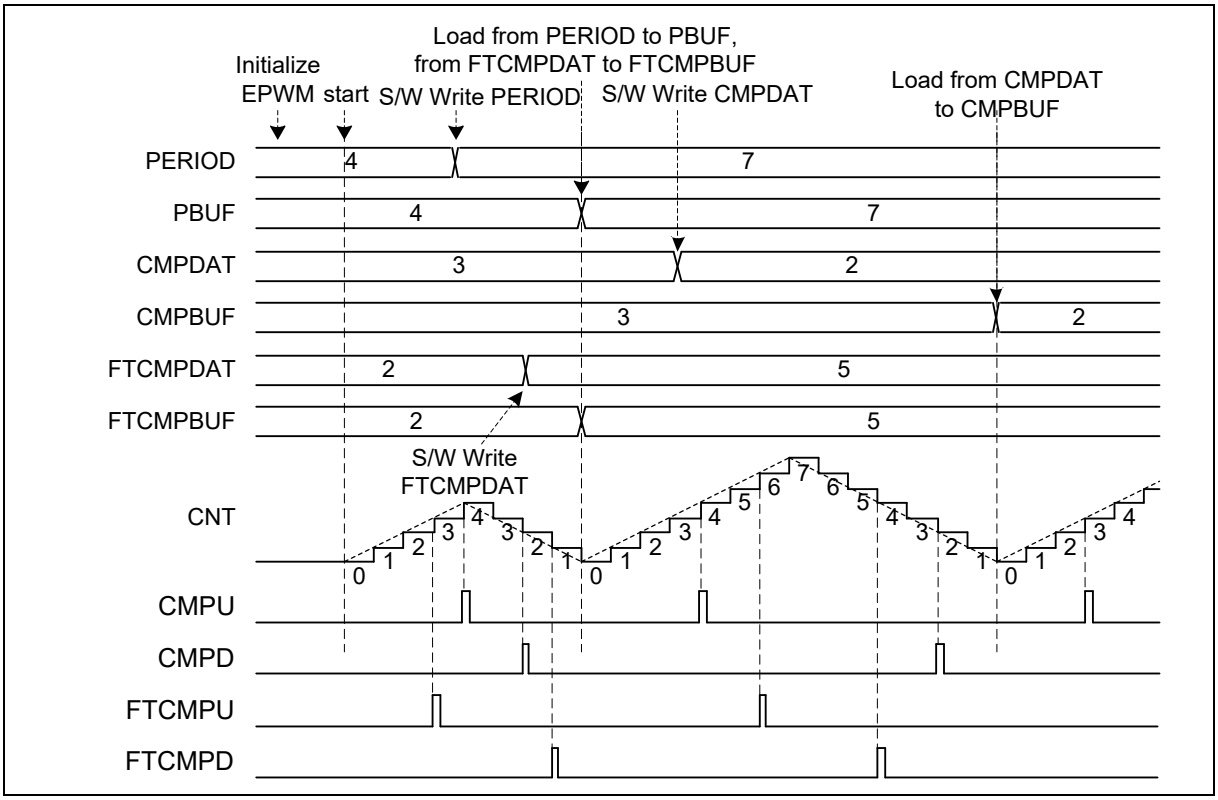


Figure 6.18-12 EPWM Double Buffering Illustration

6.18.5.8 Period Loading Mode

When the immediately loading mode, window loading mode and center loading mode are disabled that IMMLDENn bits, WINLDENn bits and CTRLDn bits of EPWM_CTL0 register are set to 0, EPWM operates in period Loading mode. In period Loading mode, CLKPSC(EPWM_CLKPSCn_m[11:0]), PERIOD(EPWM_PERIODn[15:0]) and CMP(EPWM_CMPDATn[15:0]) will all be loaded to their active CPSCBUF, PBUF and CMPBUF registers while each period is completed. For example, after EPWM counter up counts from 0 to PERIOD in the up-counter operation or down counts from PERIOD to 0 in the down-counter operation or counts up from 0 to PERIOD and then counts down to 0 in the up-down counter operation.

Figure 6.18-13 shows period loading timing of up-count operation, where PERIOD DATA0 denotes the initial data of PERIOD, PERIOD DATA1 denotes the first updated PERIOD data by software and so on. CMPDAT also follows this rule. The following describes steps sequence of Figure 6.18-13. User can know the PERIOD and CMPDAT update condition, by watching EPWM period and CMPU event.

1. Software writes CMPDAT DATA1 to CMPDAT at point 1.
2. Hardware loads CMPDAT DATA1 to CMPBUF at the end of EPWM period at point 2.
3. Software writes PERIOD DATA1 to PERIOD at point 3.
4. Hardware loads PERIOD DATA1 to PBUF at the end of EPWM period at point 4.
5. Software writes PERIOD DATA2 to PERIOD at point 5.
6. Hardware loads PERIOD DATA2 to PBUF at the end of EPWM period at point 6.

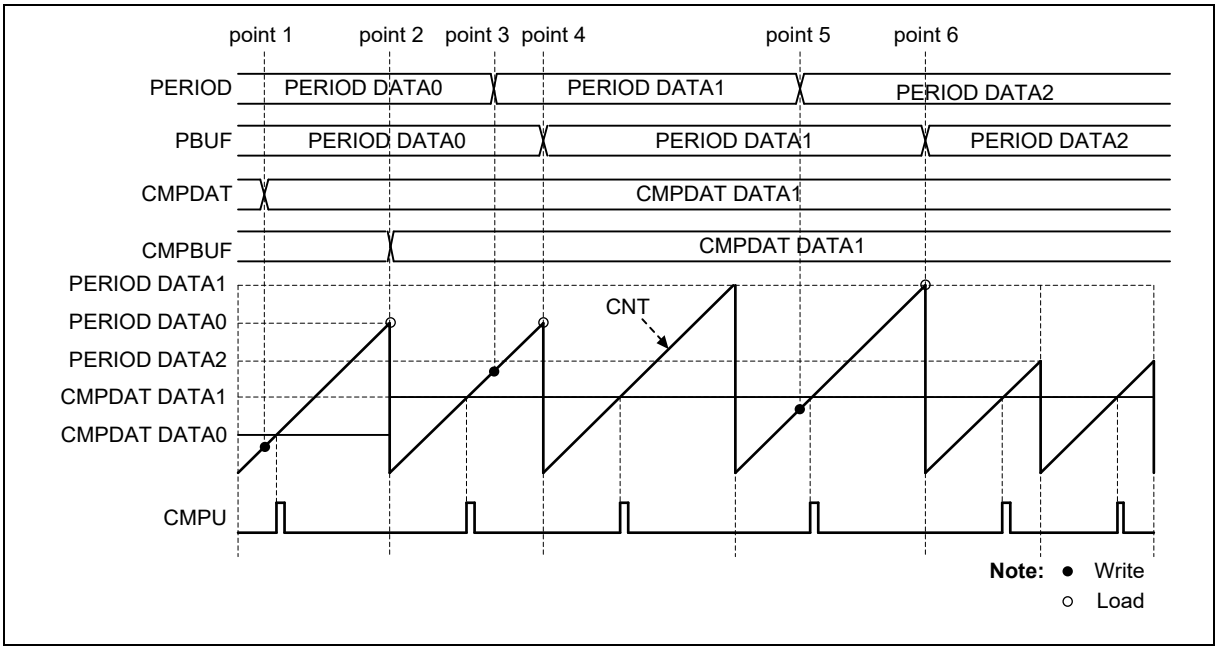


Figure 6.18-13 Period Loading in Up-Count Mode

6.18.5.9 Immediately Loading Mode

If the IMMLDENn (EPWM_CTL0[21:16]) bit is set to 1, EPWM operates at immediately loading mode. In immediately loading mode, when user updates CLKPSC(EPWM_CLKPSCn_m[11:0]), PERIOD(EPWM_PERIODn[15:0]) or CMP(EPWM_CMPDATn[15:0]), PERIOD or CMPDAT will be load to active CPSCBUF (EPWM_CPSCBUFn_m[15:0]), PBUF (EPWM_PBUFn[15:0]) or CMPBUF (EPWM_CMPBUFn[15:0]) after current counter count is completed. If the updated PERIOD value is less than current counter value, counter will counts to 0xFFFF, when counter counts to 0xFFFF and prescale counts to 0, the flag CNTMAXF(EPWMx_STATUS[5:0]) will raise, and then counter will count wraparound. Immediately loading mode has the highest priority. If IMMLDENn has been set, other loading mode for channel n will become invalid. Figure 6.18-14 shows an example and its steps sequence is described below.

1. Software writes CMPDAT DATA1 and hardware immediately loading CMPDAT DATA1 to CMPBUF at point 1.
2. Software writes PERIOD DATA1 which is greater than current counter value at point 2; counter will continue counting until equal to PERIOD DATA1 to finish a period loading.
3. Software writes PERIOD DATA2 which is less than the current counter value at point 3; counter will continue counting to its maximum value 0xFFFF and count wraparound from 0 to PERIOD DATA2 to finish this period loading.

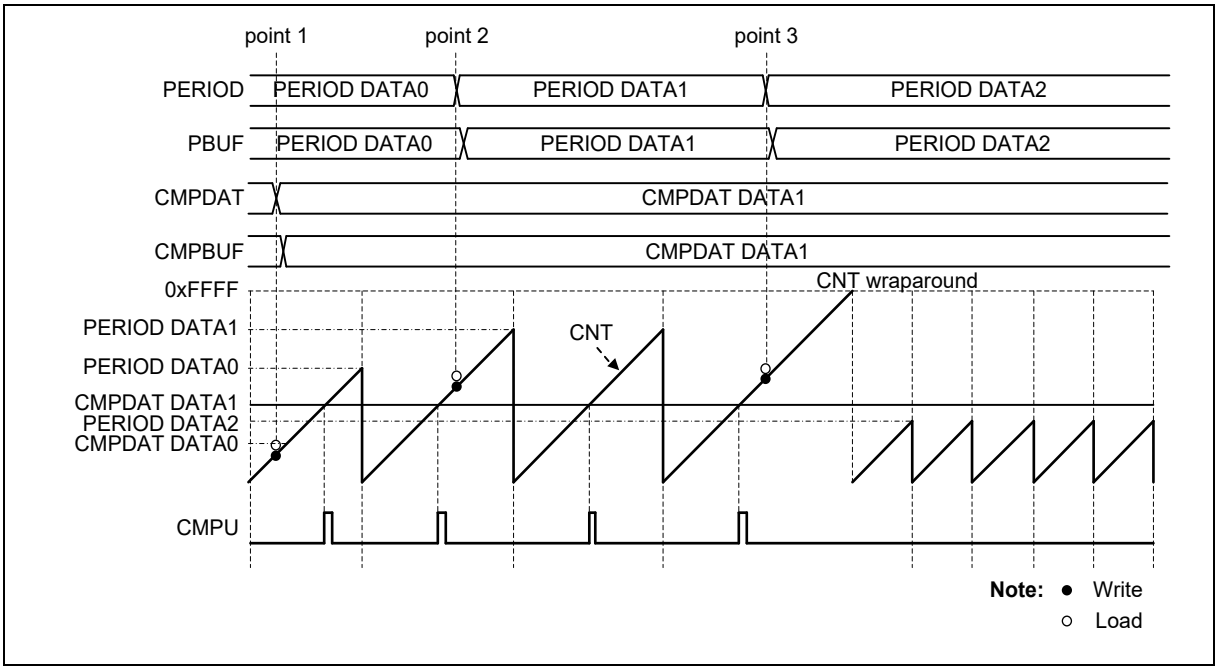


Figure 6.18-14 Immediately Loading in Up-Count Mode

6.18.5.10 Window Loading Mode

When the WINLDENn (EPWM_CTL0[13:8]) bit is set to 1, EPWM operates at window loading mode. In Window loading mode, CLKPSC(EPWM_CLKPSCn_m[11:0]), PERIOD(EPWM_PERIODn[15:0]) and CMP(EPWM_CMPDATn[15:0]) will all be loaded to their active CPSCBUF, PBUF and CMPBUF registers while each period is completed, but CMPBUF loading are valid only when load window is opened. Every channel n's load window is opened by setting the corresponding LOADn (EPWM_LOAD[5:0]) to 1, and hardware will close the window at the end of EPWM period. Figure 6.18-15 shows an example and its steps sequence is described below.

1. Software writes CMPDAT DATA1 at point 1, and the load window is not opened at this period so CMPDAT will not load to CMPBUF.
2. Software writes LOAD to open the load window at point2.
3. Software writes CLKPSC DATA1 and PERIOD DATA1 at point 3.
4. At point 4, load window has been opened, hardware loads CLKPSC DATA1, PERIOD DATA1 and CMPDAT DATA1 to their buffer and closes the load window at the end of EPWM period.
5. Software writes CLKPSC DATA2 and PERIOD DATA2 at point 5.
6. Hardware loads CLKPSC DATA2 and PERIOD DATA2 to their buffer at the end of EPWM period at point 6.
7. Software writes CLKPSC DATA3 and PERIOD DATA3 at point 7.
8. Software writes LOAD to open the load window at point8.
9. Hardware loads CLKPSC DATA3 and PERIOD DATA3 to their buffer and closes the load window at the end of EPWM period at point 9.

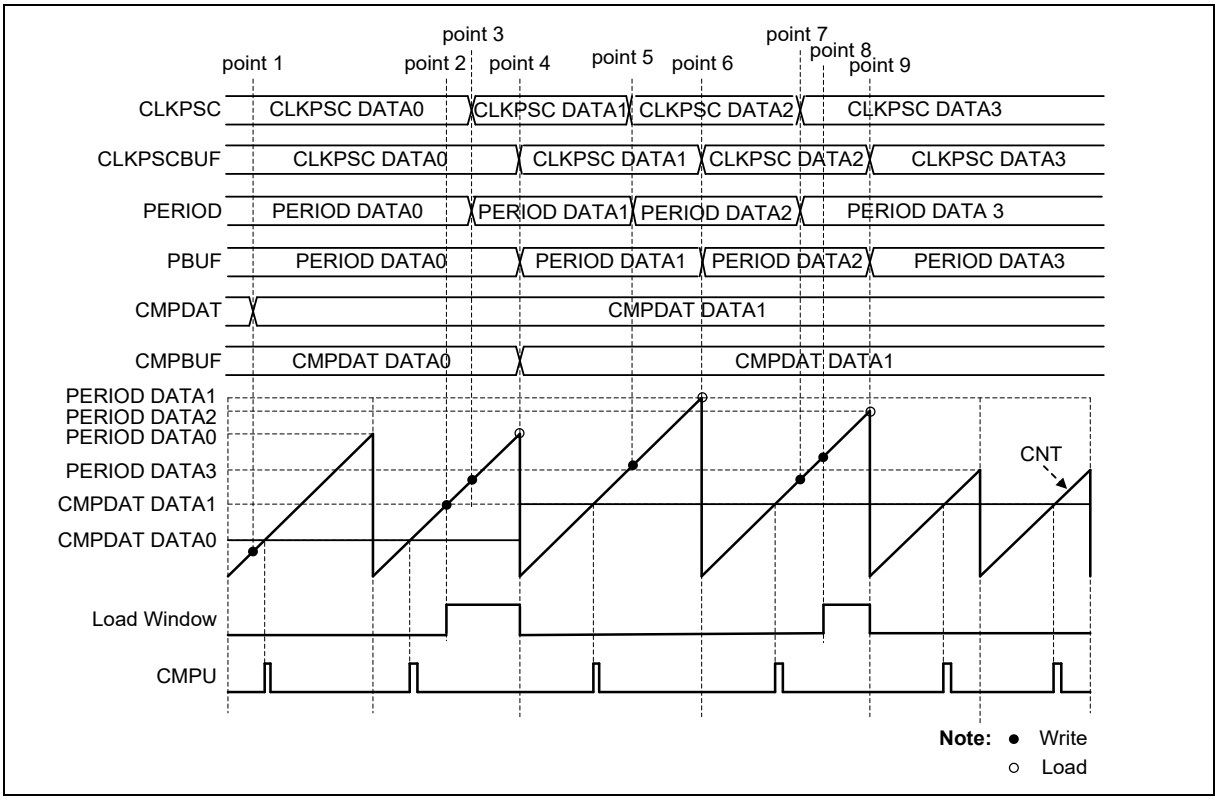


Figure 6.18-15 Window Loading in Up-Count Mode

6.18.5.11 Center Loading Mode

When the CTRLDn (EPWM_CTL0[5:0]) bit is set to 1 and EPWM counter is set to up-down count type, CNTTYPEn (EPWM_CTL1[2n+1:2n], n = 0,1..5) is 0x2, EPWM operates at center loading mode. In center loading mode, CMP(EPWM_CMPDATn[15:0]) will be loaded to active CMPBUF register in center of each period, that is, counter counts to PERIOD. CLKPSC(EPWM_CLKPSCn_m[11:0]) and PERIOD(EPWM_PERIODn[15:0]) will all load to their active CPSCBUF and PBUF registers while each period is completed. Center loading mode can work with window loading mode, the CMP(EPWM_CMPDATn[15:0]) will load to active CMPBUF register in center of each period, but it is valid only at the interval of load window. Figure 6.18-16 shows an example and its steps sequence is described below.

1. Software writes CMPDAT DATA1 at point 1.
2. Hardware loads CMPDAT DATA1 to CMPBUF at center of EPWM period at point 2.
3. Software writes PERIOD DATA1 at point 3.
4. Hardware loads PERIOD DATA1 to PBUF at the end of EPWM period at point 4.
5. Software writes CMPDAT DATA2 at point 5.
6. Hardware loads CMPDAT DATA2 to CMPBUF at center of EPWM period at point 6.
7. Software writes PERIOD DATA2 at point 7.
8. Hardware loads PERIOD DATA2 to PBUF at the end of EPWM period at point 8.

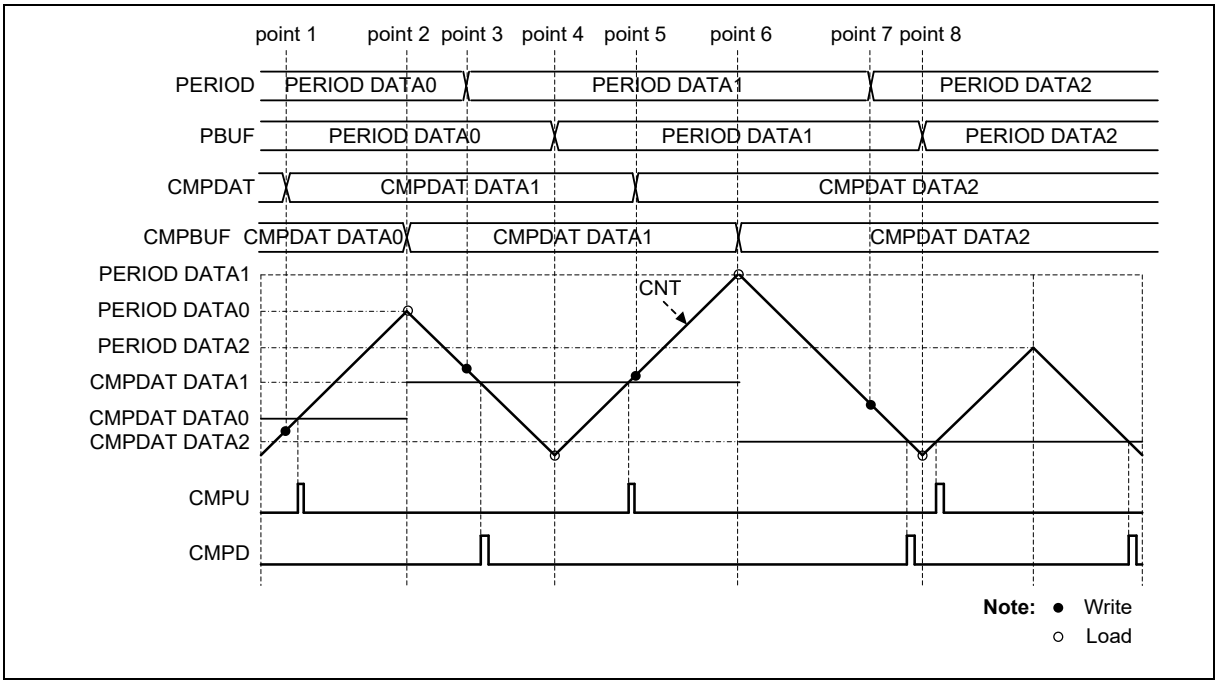


Figure 6.18-16 Center Loading in Up-Down-Count Mode

6.18.5.12 EPWM Counter Operation Mode

The EPWM counter supports two operation modes: One-shot mode and Auto-reload mode. EPWM counter will operate in One-shot mode if CNTMODEn (EPWM_CTL1[21:16]) bit is set to 1, and operate in Auto-reload mode if set to 0.

In One-shot mode, CMPDAT and PERIOD registers should be written first and then set CNTENn (EPWM_CNTEN[5:0]) bit as 1 to enable EPWM prescaler and counter start running. After EPWM counter counted a period, counter value will keep 0.

User can re-start next one-shot by writing new value to CMP(EPWM_CMPDATn[15:0]) bits. If one-shot counter still running, to update CMPDAT register will cause next one-shot as continuous one-shot. Besides, to write CMPDAT register twice under continuous one-shot operation, latest value in CMPDAT register is valid at next one-shot period and only generate one-shot pulse once. Moreover, if user wants to clear counter within one-shot operation and starts next one-shot, user should monitor counter value to check counter has cleared and then writes CMPDAT register. Figure 6.18-17 is an example and following is steps sequence.

1. Software writes PERIOD DATA1 and hardware immediately loading PERIOD DATA1 to PBUF at point 1.
2. Software writes CMPDAT DATA1 which is equal to CMPDAT DATA0 at point 2 and hardware immediately loading CMPDAT DATA1 to CMPBUF, this event also trigger one-shot.
3. Software writes CMPDAT DATA2 and re-trigger next one-shot (continuous one-shot) at point 3.
4. Software writes CMPDAT DATA3 to cover CMPDAT DATA2 and re-trigger next one-shot at point 4.
5. Period loading CMPDAT DATA3 to CMPBUF at point 5.
6. There are no new CMPDAT write in the previous period, and the counter value is kept as 0 at point 6.

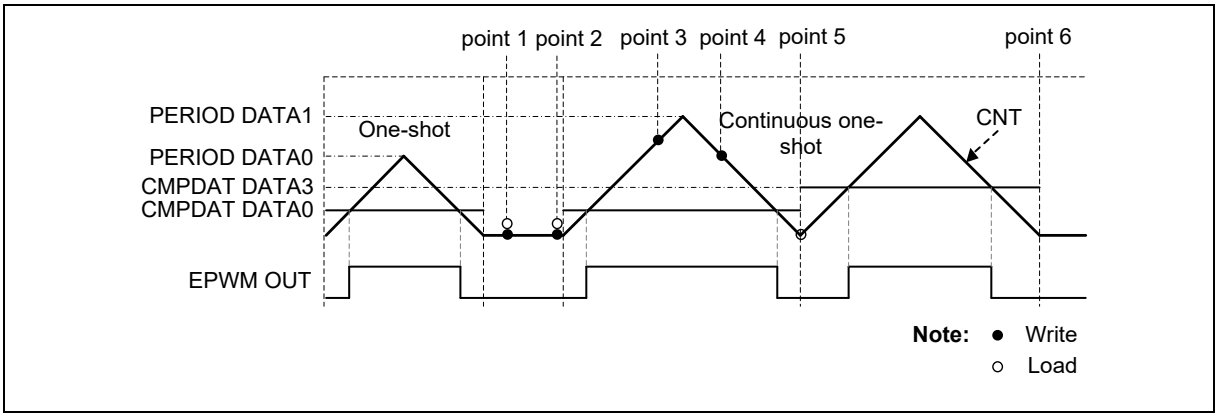


Figure 6.18-17 EPWM One-shot Mode Output Waveform

In Auto-reload mode, CMPDAT and PERIOD registers should be written first and then the CNTENn(EPWM_CNTEN[n]) bit is set to 1 to enable EPWM prescaler and start to run counter. The value of CLKPSC(EPWM_CLKPSCn_m[11:0]), PERIOD(EPWM_PERIODn[15:0]) and CMP(EPWM_CMPDATn[15:0]) will auto reload to their active buffer according different loading mode. If PERIOD(EPWM_PERIODn[15:0]) is set to 0, EPWM counter will be set to 0.

6.18.5.13 EPWM Pulse Generator

The EPWM pulse generator uses counter and comparator events to generate EPWM pulse. The events are: zero point, period point in up counter type and down counter type, center point in up-down counter type and counter equal to comparator point in three types. As to up-down counter type, there are two counter equal comparator points, one at up count and the other at down count. Besides, Complementary mode has two comparators compared with counter, and thus comparing equal points will become four in up-down counter type and two for up or down counter type.

Each event point can decide EPWM waveform to do nothing (X), set Low (L), set High (H) or toggle (T) by setting the EPWM_WGCTL0 and EPWM_WGCTL1 registers. Using these points can easily generate asymmetric EPWM pulse or variant waveform as shown in Figure 6.18-18. In the figure, EPWM is in complementary mode, there are two comparators n and m to generate EPWM pulse. n denotes even channel number 0, 2, or 4, and m denotes odd channel number 1, 3, or 5. n channel and m channel are complementary paired. Complementary mode uses two channels (CH0 and CH1, CH2 and CH3, or CH4 and CH5) as a pair of EPWM outputs to generate complement paired waveforms. CMPU denotes CNT(EPWM_CNTn[15:0]) is equal to CMP(EPWM_CMPDATn[15:0]) when counting up. CMPD denotes CNT bits is equal to CMP bits when counting down.

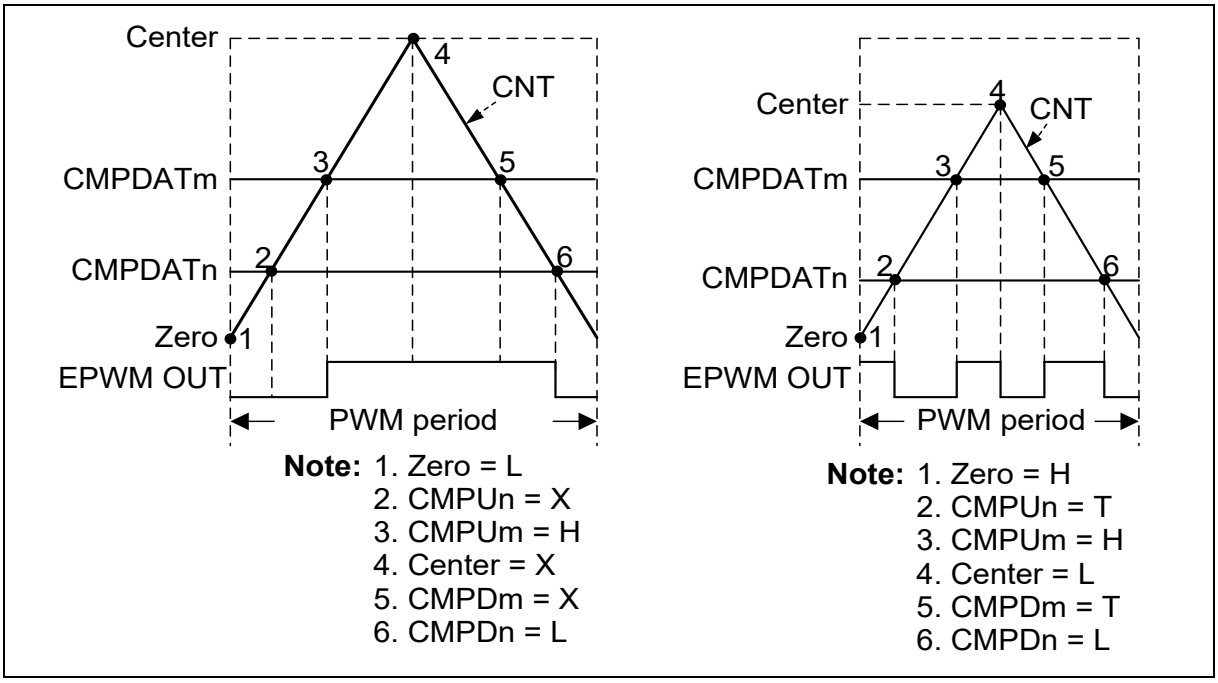


Figure 6.18-18 EPWM Pulse Generation

The generation events may sometimes set to the same value, as the reason, events priority between different counter types include up counter type (Table 6.18-1), down counter type (Table 6.18-2) and up-down counter type (Table 6.18-3). By using event priority, user can easily generate 0% to 100% duty pulse as shown in Figure 6.18-19.

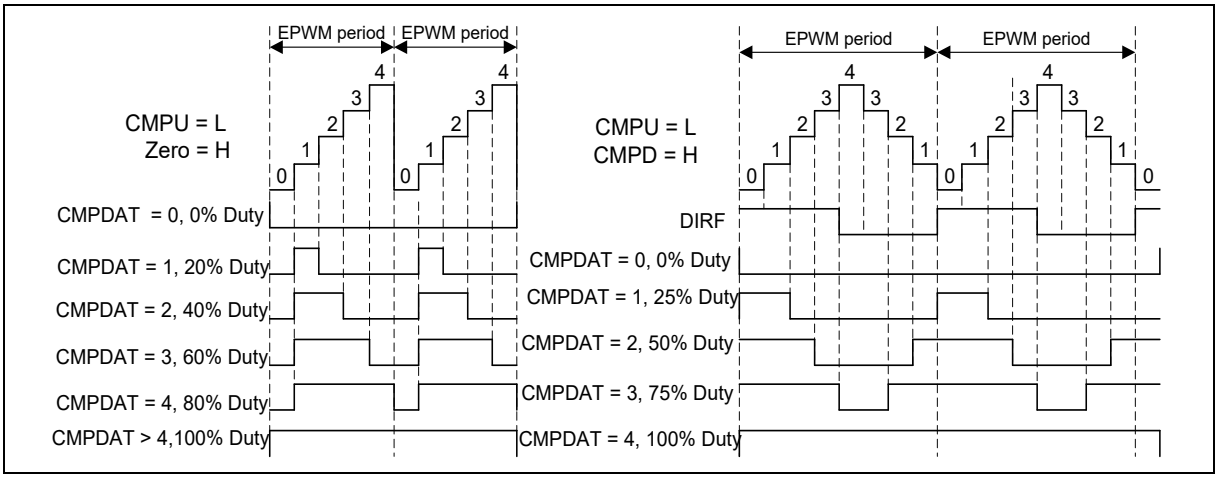


Figure 6.18-19 EPWM 0% to 100% Pulse Generation

Priority	Up Event
1 (Highest)	Period event (CNT = PERIOD)
2	Compare up event of odd channel (CNT = CMPUm)
3	Compare up event of even channel (CNT = CMPUn)
4 (Lowest)	Zero event (CNT = 0)

Table 6.18-1 EPWM Pulse Generation Event Priority for Up-Counter

Priority	Down Event
1 (Highest)	Zero event (CNT = 0)
2	Compare down event of odd channel (CNT = CMPDm)
3	Compare down event of even channel (CNT = CMPDn)
4 (Lowest)	Period event (CNT = PERIOD)

Table 6.18-2 EPWM Pulse Generation Event Priority for Down-Counter

Priority	Up Event	Down Event
1 (Highest)	Compare up event of odd channel (CNT = CMPUm	Compare down event of odd channel (CNT = CMPDm)
2	Compare up event of even channel (CNT = CMPUn)	Compare down event of even channel (CNT = CMPDn)
3	Zero event (CNT = 0)	Period (center) event (CNT = PERIOD)
4	Compare down event of odd channel (CNT = CMPDm)	Compare up event of odd channel (CNT = CMPUm)
5 (Lowest)	Compare down event of even channel (CNT = CMPDn)	Compare up event of even channel (CNT = CMPUn)

Table 6.18-3 EPWM Pulse Generation Event Priority for Up-Down-Counter

6.18.5.14 EPWM Output Mode

The EPWM supports two output modes: Independent mode that may be applied to DC motor system, Complementary mode with dead-time insertion that may be used in the application of AC induction motor and permanent magnet synchronous motor.

6.18.5.15 Independent mode

By default, the EPWM is operating in independent mode, independent mode is enabled when channel n corresponding OUTMODEn (EPWM_CTL1[26:24]) bit is set to 0. In this mode six EPWM channels: EPWM_CH0, EPWM_CH1, EPWM_CH2, EPWM_CH3, EPWM_CH4 and EPWM_CH5 are running off its own period and duty as shown in Figure 6.18-20.

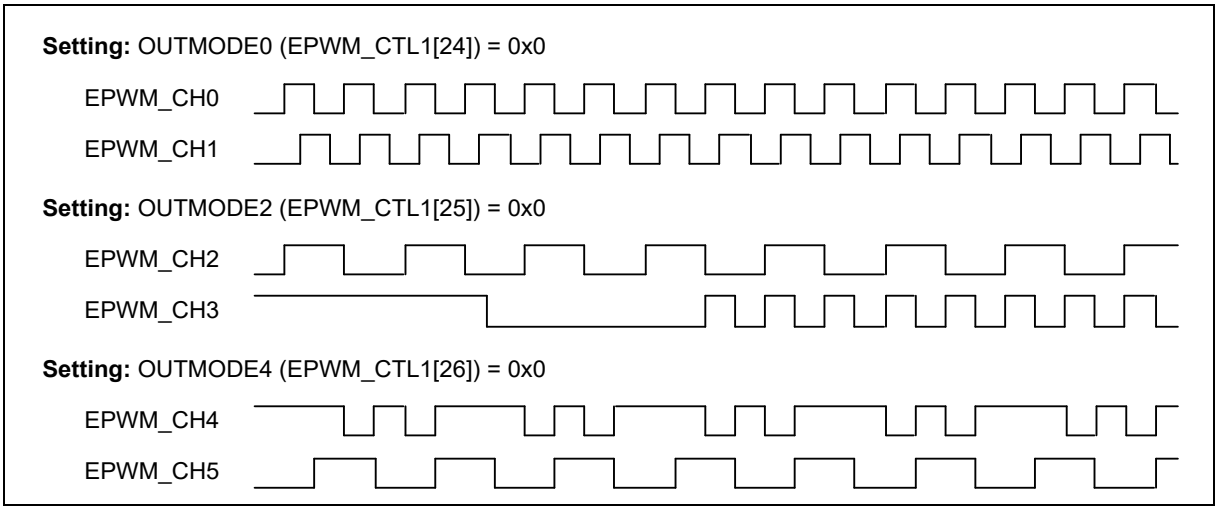


Figure 6.18-20 EPWM Independent Mode Waveform

6.18.5.16 Complementary Mode

Complementary mode is enabled when the pair channel corresponding OUTMODEn (EPWM_CTL1[26:24]) bit is set to 1. In this mode there are three EPWM generators utilized for complementary mode, with a total of 3 EPWM output paired pins in this module. In Complimentary modes, the internal odd EPWM signal must always be the complement of the corresponding even EPWM signal. EPWM_CH1 will be the complement of EPWM_CH0. EPWM_CH3 will be the complement of EPWM_CH2 and EPWM_CH5 will be the complement of EPWM_CH4 as shown in Figure 6.18-21.

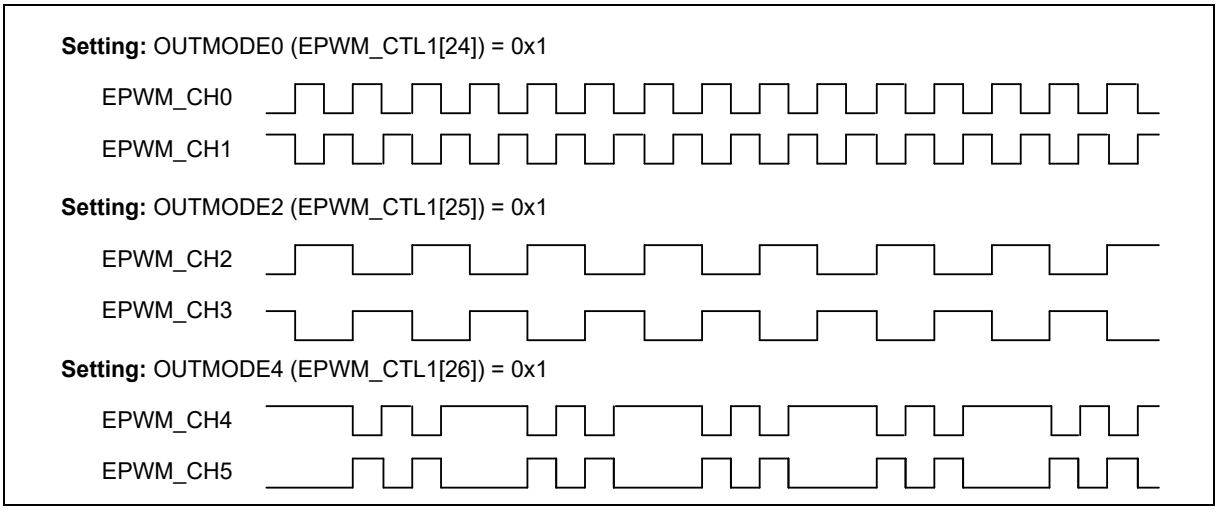


Figure 6.18-21 EPWM Complementary Mode Waveform

6.18.5.17 EPWM Output Function

Based on the output mode, there are two output functions: group and synchronous functions for advanced output control. Group function, forces the EPWM_CH2 and EPWM_CH4 synchronous with EPWM_CH0 generator and forces the EPWM_CH3 and EPWM_CH5 synchronous with EPWM_CH1, may simplify updating duty control in DC and BLDC motor applications. Besides, Synchronous function makes any channel of EPWM0 and EPWM1 in phase, user can control phase value and direction.

6.18.5.18 Group Function

Group function is enabled when GROUPEN (EPWM_CTL0[24]) is set to 1, no matter in independent or

complementary mode. This control allows all even EPWM channels output to be controllable by EPWM_PERIOD0 and EPWM_CMPDAT0 registers and all odd EPWM channels output to be controllable by EPWM_PERIOD1 and EPWM_CMPDAT1 registers. That is, user only needs to set EPWM_CH0 to get EPWM_CH0, EPWM_CH2 and EPWM_CH4 output the same pulse, and set EPWM_CH1 to get EPWM_CH1, EPWM_CH3 and EPWM_CH5 output the same pulse, as shown in Figure 6.18-22. When operating group function, OUTMODE0, OUTMODE2 and OUTMODE4 bits of CTL1 register must all be set to 0 for independent mode or all set to 1 for complementary mode.

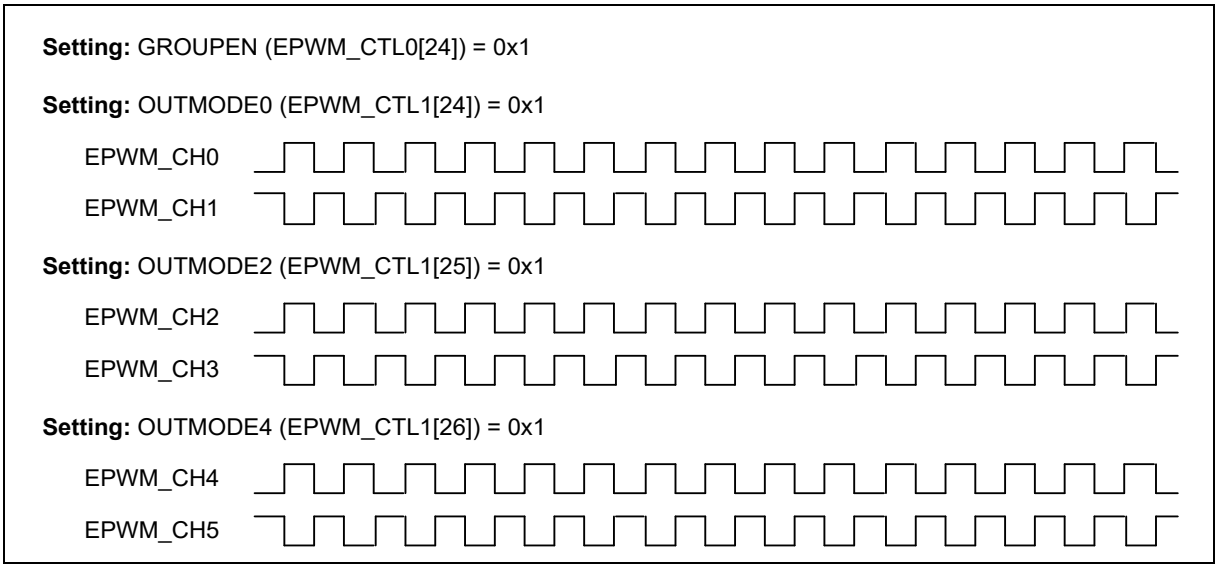


Figure 6.18-22 EPWM Group Function Waveform

6.18.5.19 Synchronous Function

Synchronous function can only be enabled when complementary mode is enabled. Figure 6.18-24 is counter synchronous function block diagram. Every counter of EPWM pairs has a SYNC_IN and a SYNC_OUT signals. The SYNC_IN signal for the first EPWM0 pair counter comes from EPWM0_SYNC_IN pin, and the others come from the SYNC_OUT signal of the previous EPWM pair counter. The input signal from EPWM0_SYNC_IN pin will be filtered by a 3-bit noise filter as Figure 6.18-23. In addition, it can be inversed by setting the bit SINPINV (EPWM_SYNC[23]) to realize the polarity setup for the input signal. The noise filter sampling clock can be selected by setting bits SFLTCSEL (EPWM_SYNC[19:17]) to fit different noise properties. Moreover, by setting the bits SFLTCNT (EPWM_SYNC[22:20]), user can define by how many sampling clock cycles a filter will recognize the effective edge of the SYNC_IN signal. Configuring the SNFLTEN (EPWM_SYNC[16]) will enable the noise filter function. By default, it is disabled.

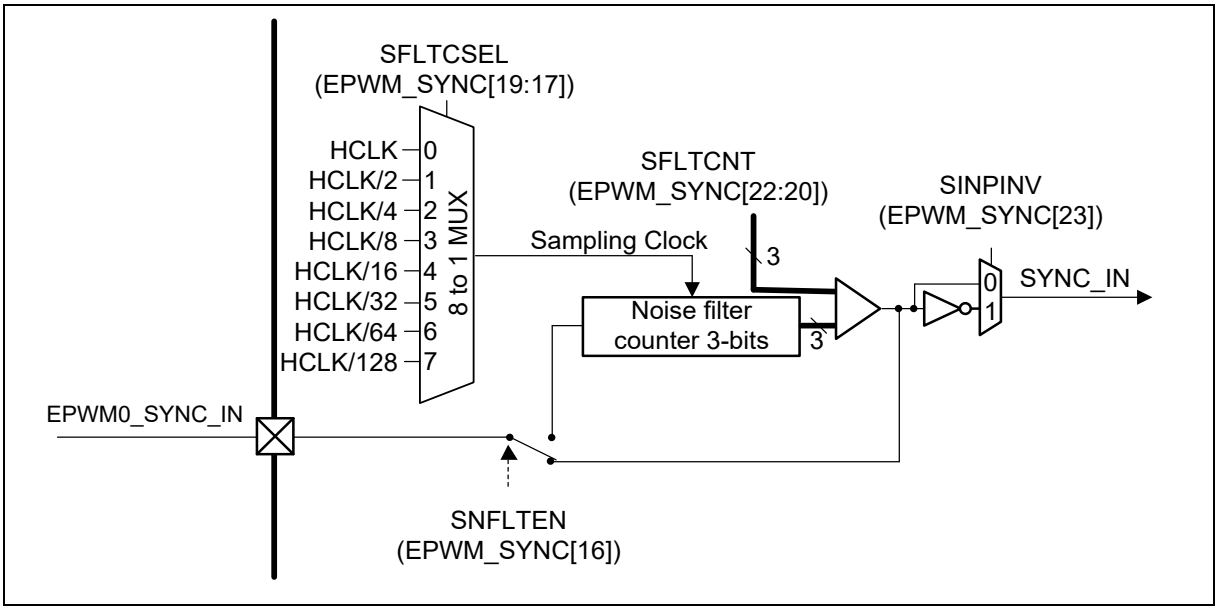


Figure 6.18-23 EPWM SYNC_IN Noise Filter Block Diagram

User can uses SINSRCn (EPWM_SYNC[13:8]) bits to select the synchronize source. When SINSRCn bits is set to 0, user can generate SYNC_IN signal for the next counter's synchronization when EPWM0_SYNC_IN pin is high or setting SWSYNCn (EPWM_SWSYNC[2:0]) to 1. Synchronizing source can also be selected as CNT = 0 or CNT = EPWM_CMPDATm register (if being the up-down counter type, it will synchronize twice in a EPWM period) to trigger a sync event or to disable SYNC_OUT signal.

When the PHSENn (EPWM_SYNC[2:0]) is enabled and the synchronous source has a happening event, the counter will load a value from the PHS (EPWM_PHSn_m[15:0]) register. This method synchronizes counters to different phase in the same time. In the up-down counter type, user can set the value in PHSDIRn (EPWM_SYNC[26:24]) to control the counter direction after synchronization. Although the Synchronous function can synchronize channels in phase, it cannot work from the beginning of EPWM enable. To start EPWM counters in the same time, user has to set the EPWM Synchronous Start Control Register (EPWM_SSCTL[5:0]) to enable the channel counters which are planned to start counting together, and select the SSRC(EPWM_SSCTL[9:8]) to choose the Synchronous Start source, followed by setting the EPWM Synchronous Start Trigger Register CNTSEN (EPWM_SSTRG[0]).

For applications, please do not uses Group and Synchronous function simultaneously because the Synchronous function will be inactive.

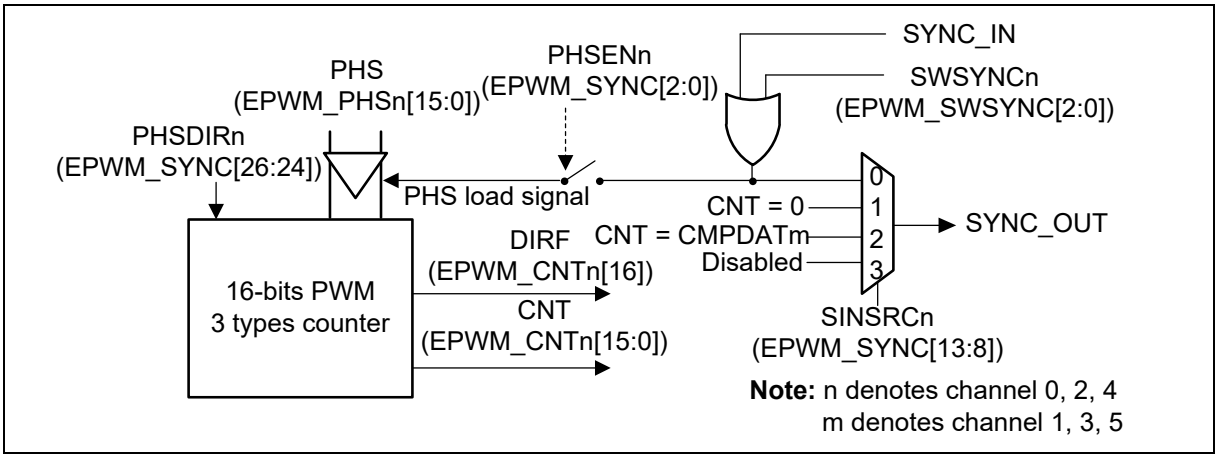


Figure 6.18-24 EPWM Counter Synchronous Function Block Diagram

Figure 6.18-25 is an example of the synchronous function in the up-down counter type. In the example, synchronizing source comes from the external EPWM SYNC_IN signal. At the beginning, the output waveform of EPWM_CH0, EPWM_CH2 and EPWM_CH4 are in the same phase. Then at Point A, the EPWM SYNC input signal comes as a sync event, resulting in phase shifts and counting direction changes for all of the counters. To realize the altered counter behaviors before the sync event coming, user has to setup the corresponding phase value in the PHS of(EPWM_PHSn_m[15:0]) as well as the counting direction in the PHSDIRn (EPWM_SYNC[26:24]). In this case, one third of phase shifts are made. by setting the corresponding channel n's counter counting direction after synchronizing, as illustrated around the left side of Figure 6.18-25.

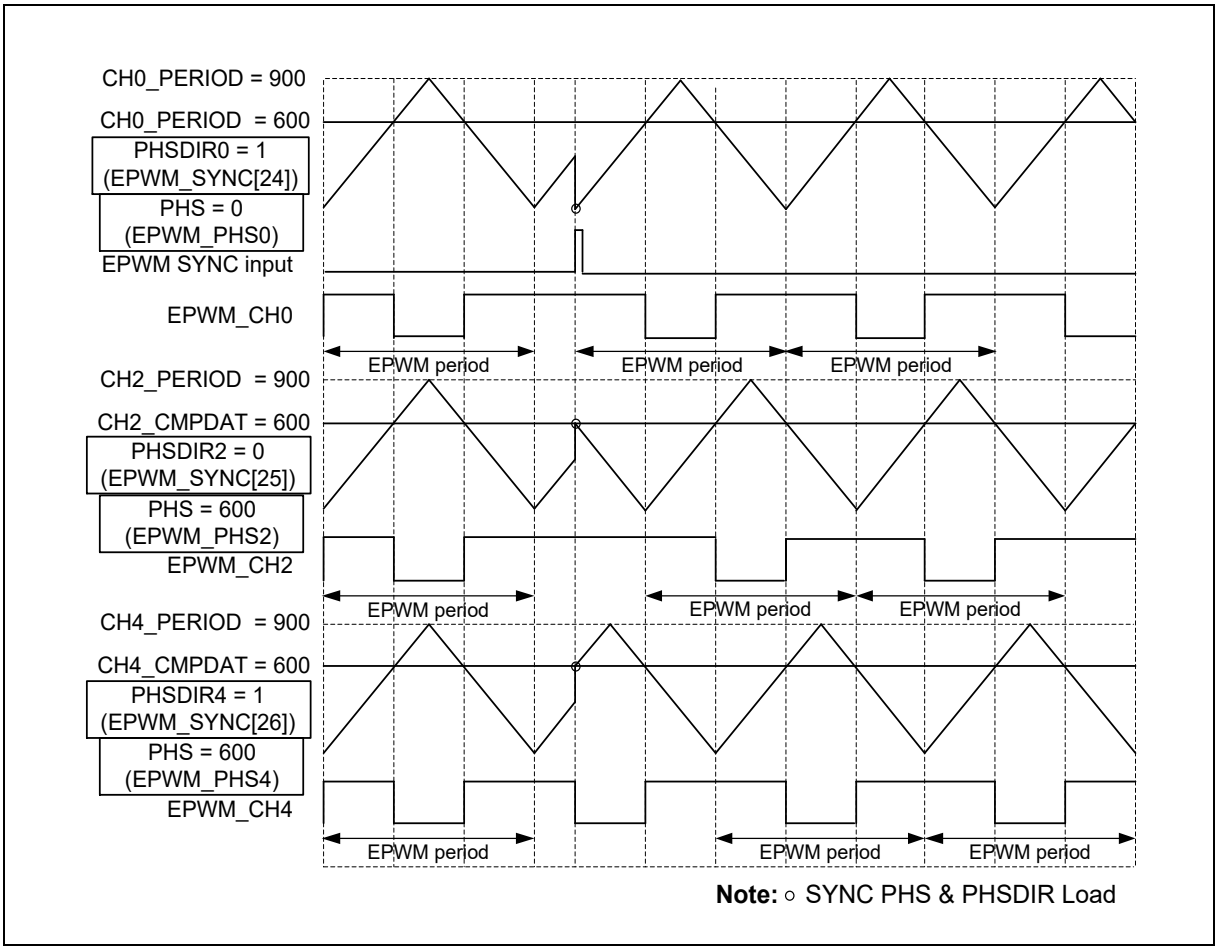


Figure 6.18-25 EPWM Synchronous Function with Synchronize Source from SYNC_IN Signal

6.18.5.20 EPWM Output Control

After EPWM pulse generation, there are four to six steps to control the output of EPWM channels. In independent mode, there are Mask, Brake, Pin Polarity and Output Enable four steps as shown in Figure 6.18-26. In complementary mode, it needs two more steps to precede these four steps, Complementary channels and Dead-Time Insertion as shown in Figure 6.18-27.

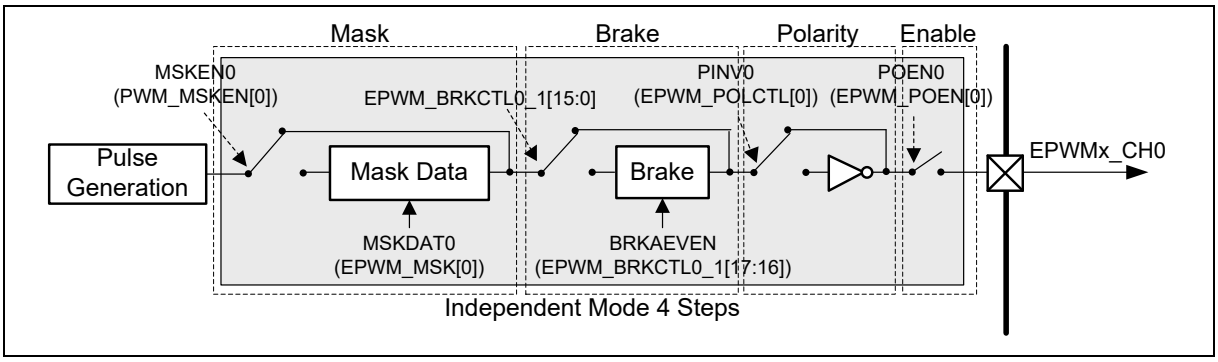


Figure 6.18-26 EPWMx_CH0 Output Control in Independent Mode

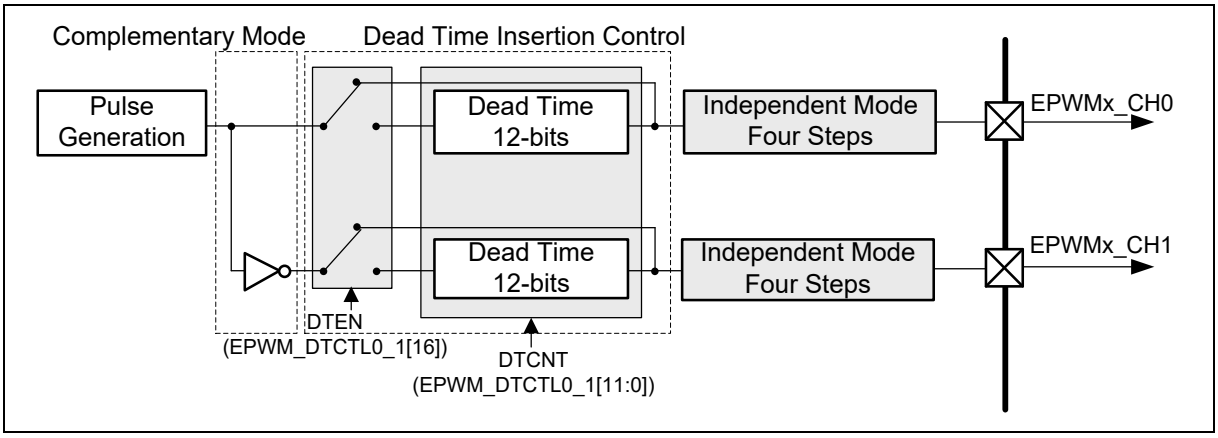


Figure 6.18-27 EPWMx_CH0 and EPWMx_CH1 Output Control in Complementary Mode

6.18.5.21 Dead-Time Insertion

In the complementary application, the complement channels may drive the external devices like power switches. The dead-time generator inserts a low level period called “dead-time” between complementary outputs to drive these devices safely and to prevent system or devices from the burn-out damage. Hence the dead-time control is a crucial mechanism to the proper operation of the complementary system. By setting corresponding channel n DTEN (EPWM_DTCTLn_m[16]) bit to enable dead-time function and DTCNT (EPWM_DTCTLn_m[11:0]) to control dead-time period, the dead-time can be calculated from the following formula:

Dead-time = (DTCNT (EPWM_DTCTLn[11:0])+1) * EPWMx_CLK period

Dead-time insertion clock source can be selected from prescaler output by setting DTCKSEL (EPWM_DTCTLn_m[24]) to 1. By default, clock source comes from EPWM_CLK, which is prescaler input. Then the dead-time can be calculated from the following formula:

Dead-time = (DTCNT (EPWM_DTCTLn[11:0])+1) *

(CLKPSC (EPWM_CLKPSCn [11:0])+1)*EPWMx_CLK period

Please note that the EPWM_DTCTLn_m are write-protected registers.

Figure 6.18-28 indicates the dead-time insertion for one pair of EPWM signals.

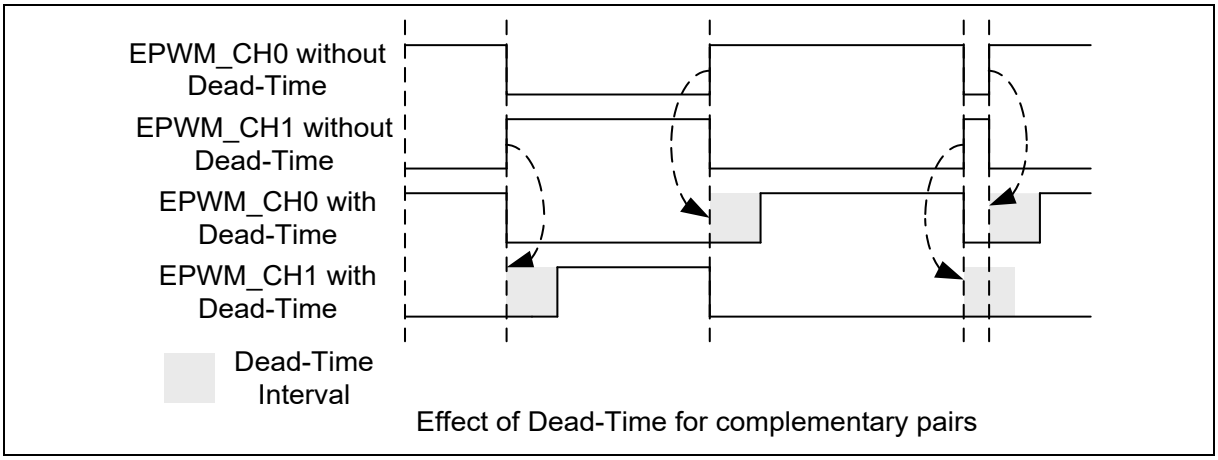


Figure 6.18-28 Dead-Time Insertion

6.18.5.22 EPWM Mask Output Function

Each of the EPWM channel output value can be manually overridden with the settings in the EPWM Mask Enable Control Register (EPWM_MSKEN) and the EPWM Masked Data Register (EPWM_MSK) With these settings, the EPWM channel outputs can be assigned to specified logic states independent of the duty cycle comparison units. The EPWM mask bits are useful when controlling various types of Electrically Commutated Motor (ECM) like a BLDC motor. The EPWM_MSKEN register contains six bits, MSKENn(EPWM_MSKEN[5:0]). If the MASKENn is set to active-high, the EPWM channel n output will be overridden. The EPWM_MSK register contains six bits, MSKDATn(EPWM_MSK[5:0]). The bit value of the MSKDATn determines the state value of the EPWM channel n output when the channel is overridden. Figure 6.18-29 shows an example of how EPWM mask control can be used for the override feature.

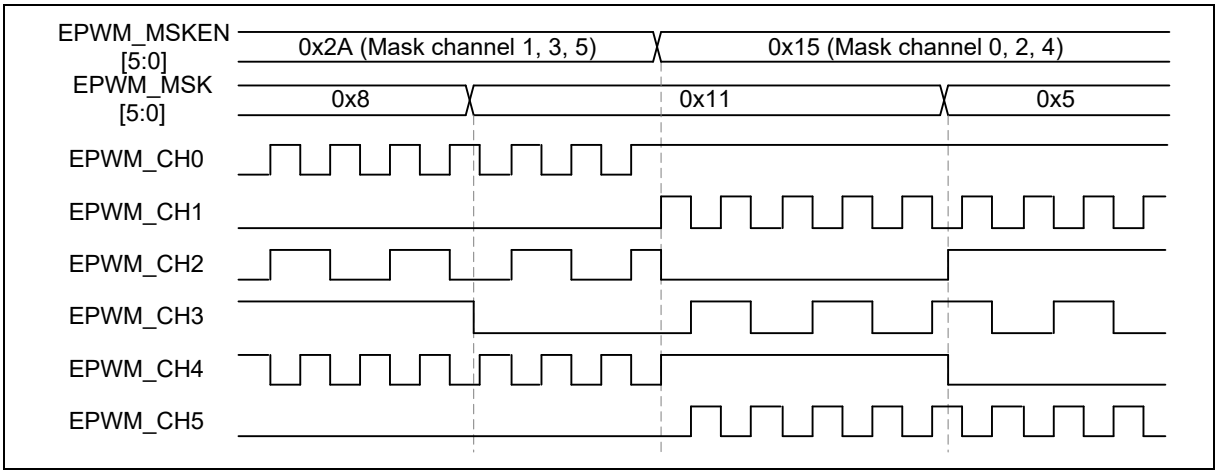


Figure 6.18-29 Illustration of Mask Control Waveform

6.18.5.23 EPWM Brake

Each EPWM module has two external input brake control signals. User can select active brake pin source is from EPWMx_BRAKEy pin by BKxSRC bits of EPWM_BNF register(x=0,1,2 y=0,1). The external signals will be filtered by a 3-bit noise filter. User can enable the noise filter function by BRKxNFEN bits of EPWM_BNF register, and noise filter sampling clock can be selected by setting BRKxNFSEL bits of EPWM_BNF register to fit different noise properties. Moreover, by setting the BRKxFCNT bits, user can define by how many sampling clock cycles a filter will recognize the effective edge of the brake signal.

In addition, it can be inversed by setting the BRKxPINV (x denotes input external pin 0, 1 or 2) bits of EPWM_BNF register to realize the polarity setup for the brake control signals. Set BRKxPINV bit to 0, brake event will occurred when EPWMx_BRAKEy(x=0,1,2 y=0,1) pin status is from low to high; set BRKxPINV to 1, brake event will occurred when EPWMx_BRAKEy pin status is from high to low.

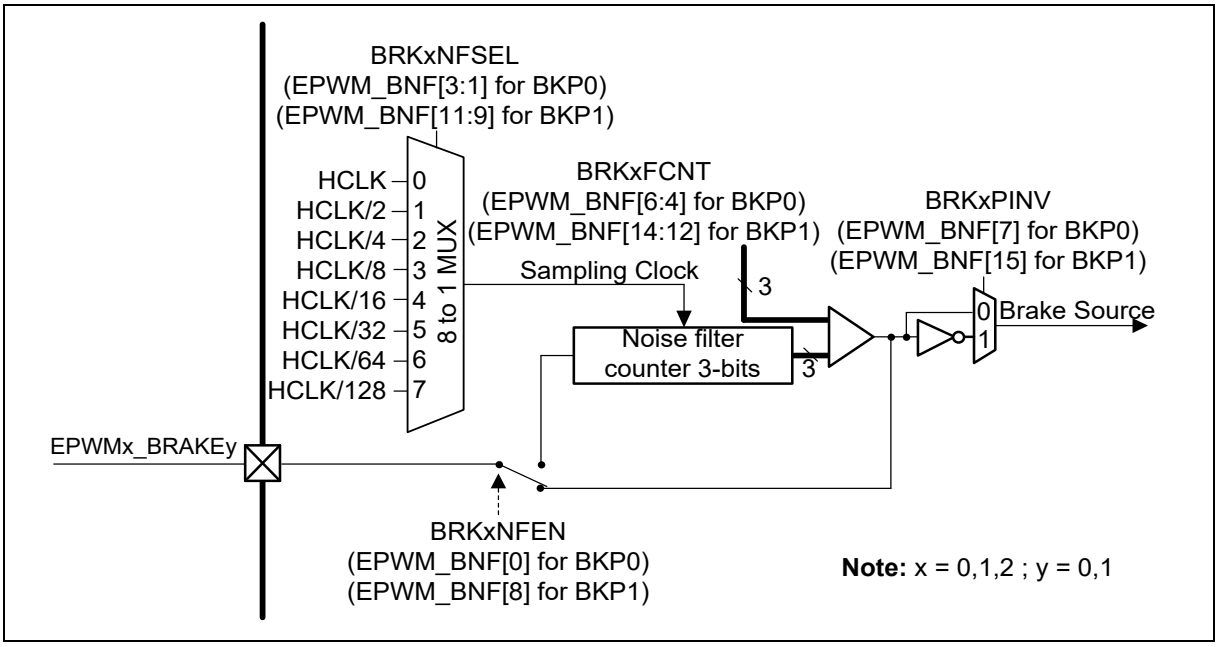


Figure 6.18-30 Brake Noise Filter Block Diagram

For Complementary mode, it is often necessary to set a safe output state to the complement output pairs once the brake event occurs.

Each complementary channel pair shares a EPWM brake function, as show Figure 6.18-31. To control paired channels to output safety state, user can setup BRKAEVEN (EPWM_BRKCTL0_1[17:16]) for even channels and BRKAODD (EPWM_BRKCTL0_1[19:18]) for odd channels when the fault brake event happens. There are two brake detectors: Edge detector and Level detector. When the edge detector detects the brake signal and BRKEIENn_m (EPWM_INTEN1[2:0]) is enabled, the brake function generates BRK_INT. This interrupt needs software to clear, and the BRKESTSn (EPWM_INTSTS1[21:16]) brake state will keep until the next EPWM period starts after the interrupt cleared. The brake function can also operate in another way through the level detector. Once the level detector detects the brake signal and the BRKLIENn_m (EPWM_INTEN1[10:8]) is also enabled, the brake function will generate BRK_INT, but BRKLSTSn (EPWM_INTSTS1[29:24]) brake state will auto recovery to normal output while level brake source recovery to high level and pass through “Low Level Detection” at the EPWM waveform period when brake condition removed without clear interrupt.

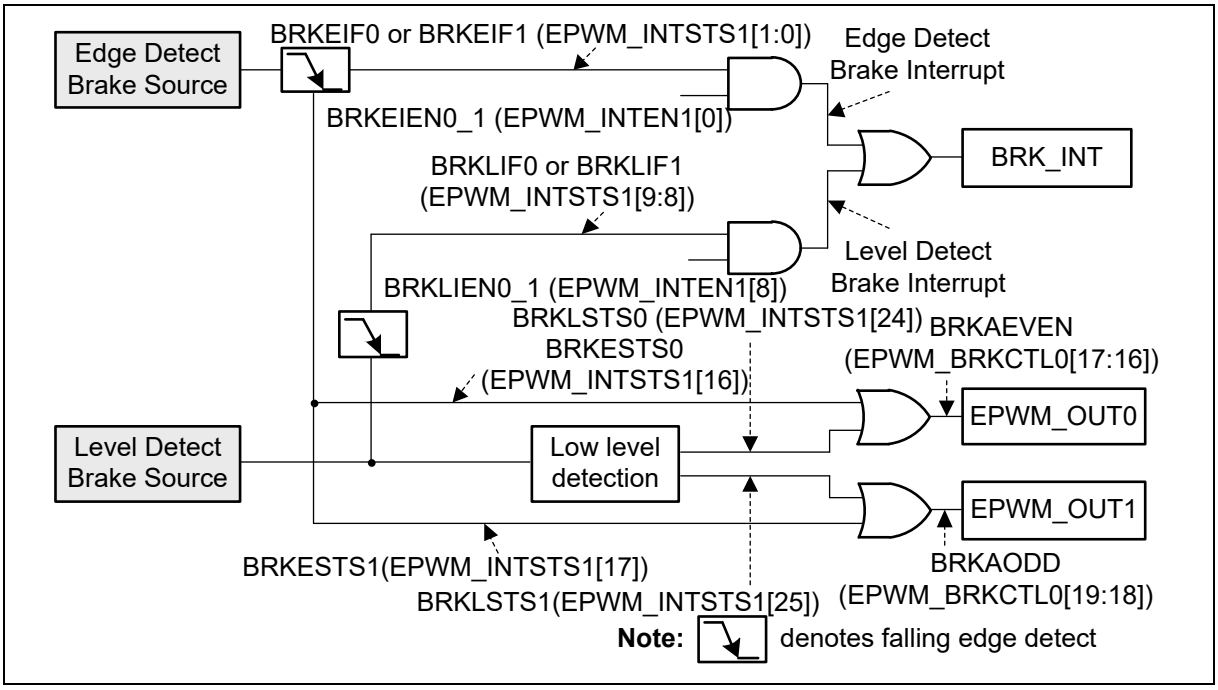


Figure 6.18-31 Brake Block Diagram for EPWMx_CH0 and EPWMx_CH1 Pair

Figure 6.18-32 illustrates the edge detector waveform for EPWMx_CH0 and EPWMx_CH1 pair. In this case, the edge detect brake source occurred twice for the brake events. When the event occurs, both of the BRKEIF0 and BRKEIF1 flags are set and BRKESTS0 and BRKESTS1 bits are also set to indicate brake state of EPWMx_CH0 and EPWMx_CH1. For the first occurring event, software writes 1 to clear the BRKEIF0 flag. After that, the BRKESTS0 bit is cleared by hardware at the next start of the EPWM period. At the same moment, the EPWMx_CH0 outputs the normal waveform even though the brake event is still occurring. The second event also triggers the same flags, but at this time, software writes 1 to clear the BRKEIF1 flag. Afterward, EPWMx_CH1 outputs normally at the next start of the EPWM period.

As a contrast to the edge detector example, Figure 6.18-33 illustrates the level detector waveform for EPWMx_CH0 and EPWMx_CH1 pair. In this case, the BRKLIF0 and BRKLIF1 flags can only indicate the brake event having occurred. The BRKLSTS0 and BRKLSTS1 brake states will automatically recover at the start of the next EPWM period no matter at what states the BRKLIF0 and BRKLIF1 flags are at that moment.

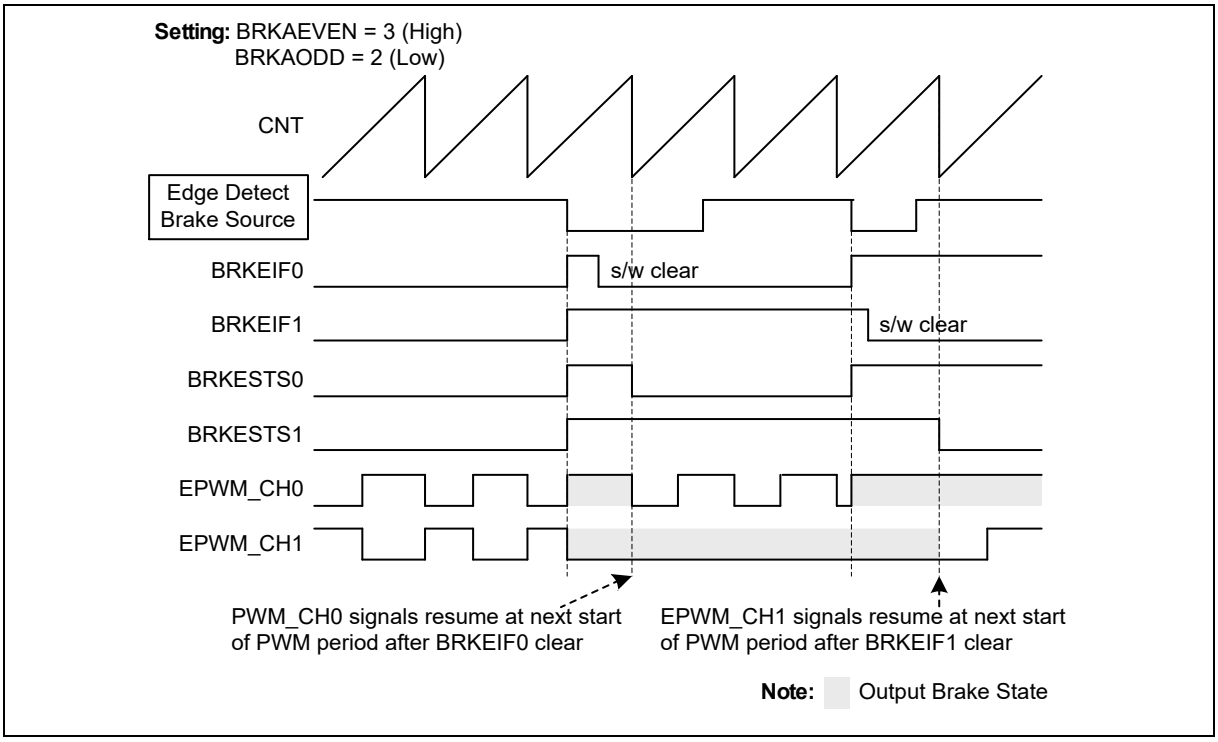


Figure 6.18-32 Edge Detector Waveform for EPWMx_CH0 and EPWMx_CH1 Pair

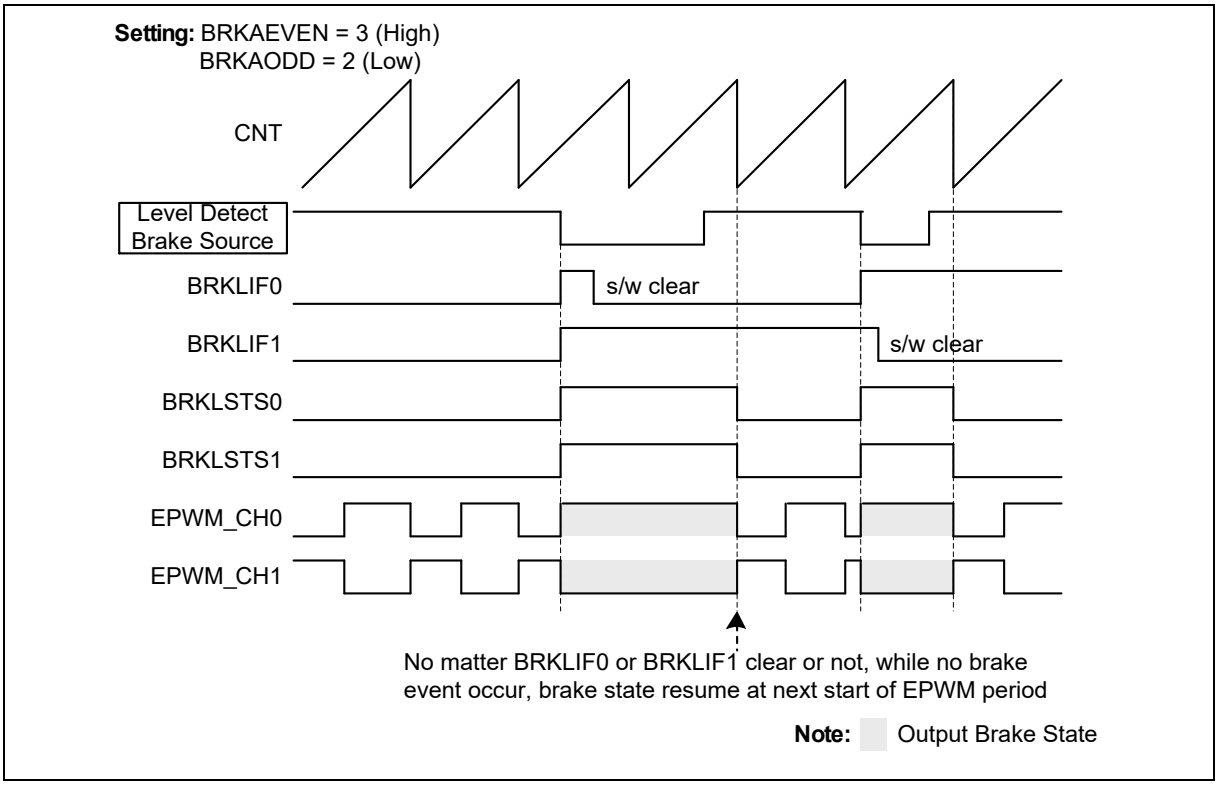


Figure 6.18-33 Level Detector Waveform for EPWMx_CH0 and EPWMx_CH1 Pair

The two kinds of detectors detect the same five brake sources: two from external input signals, one from EADC result monitor (EADCRM), one from system fail and one from software triggered, are shown in

Figure 6.18-34.

Among the above described brake sources, the brake source coming from system fail can still be specified to several different system fail conditions. These conditions include clock fail, Brown-out detect and Core lockup. Figure 6.18-35 shows that by setting corresponding enable bits, the enabled system fail condition can be one of the sources to issue the Brake system fail to the EPWM brake.

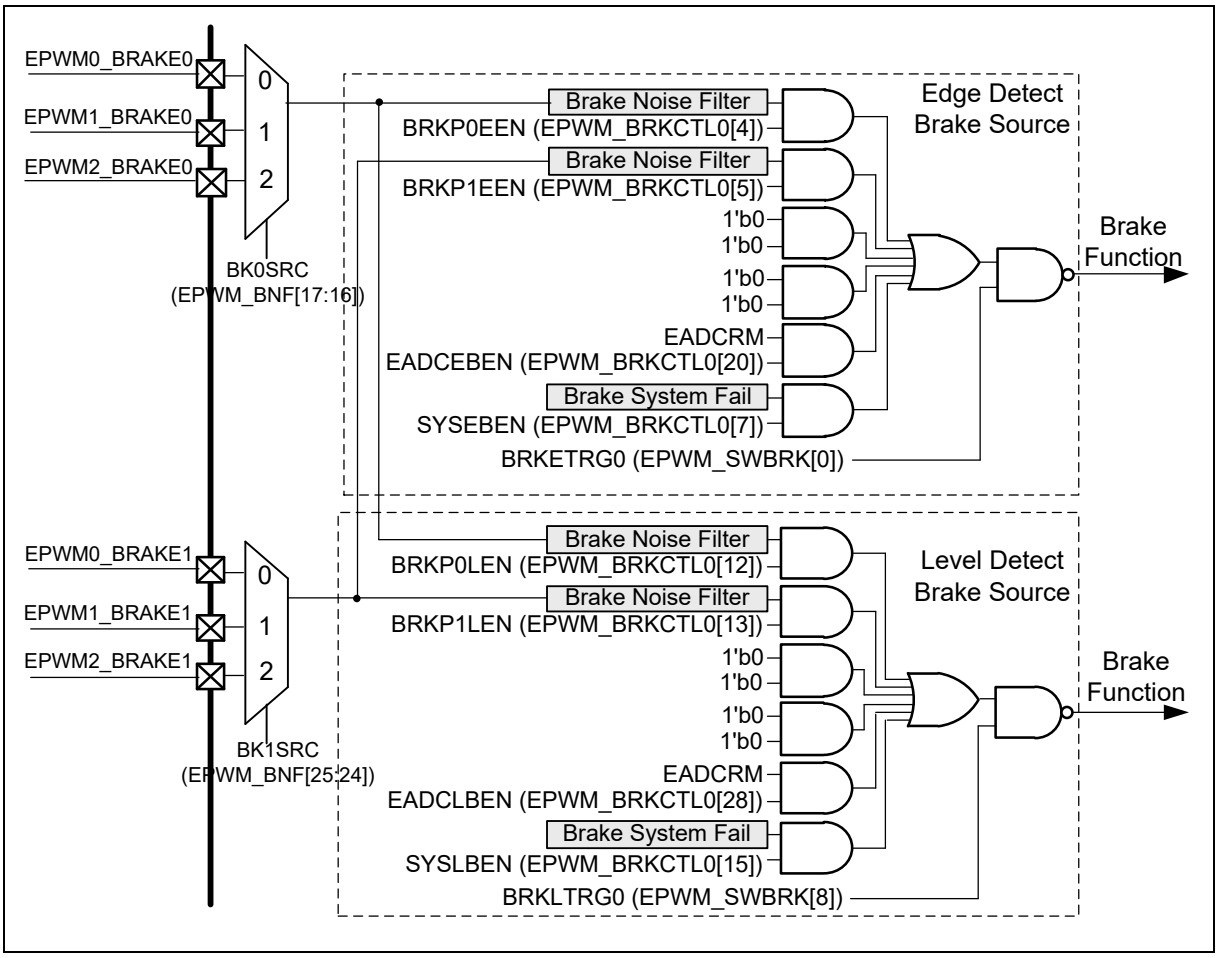


Figure 6.18-34 Brake Source Block Diagram

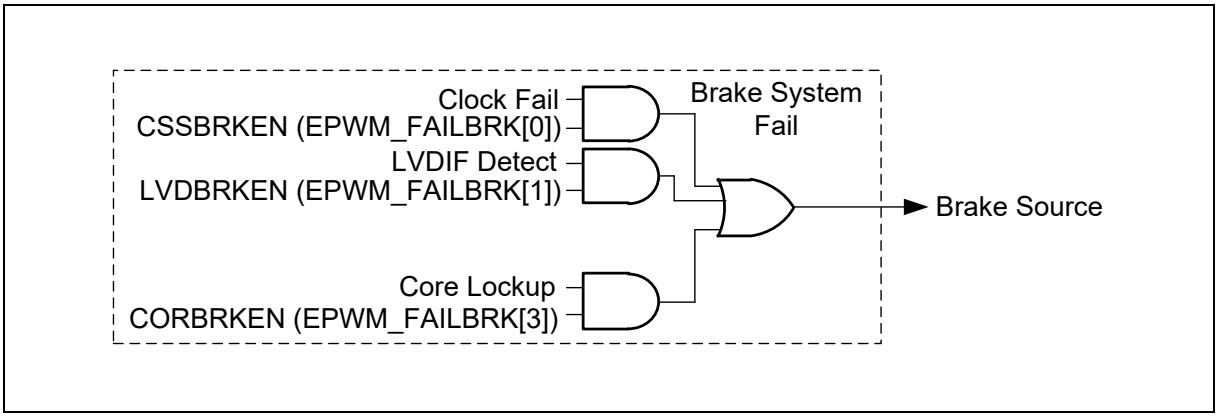
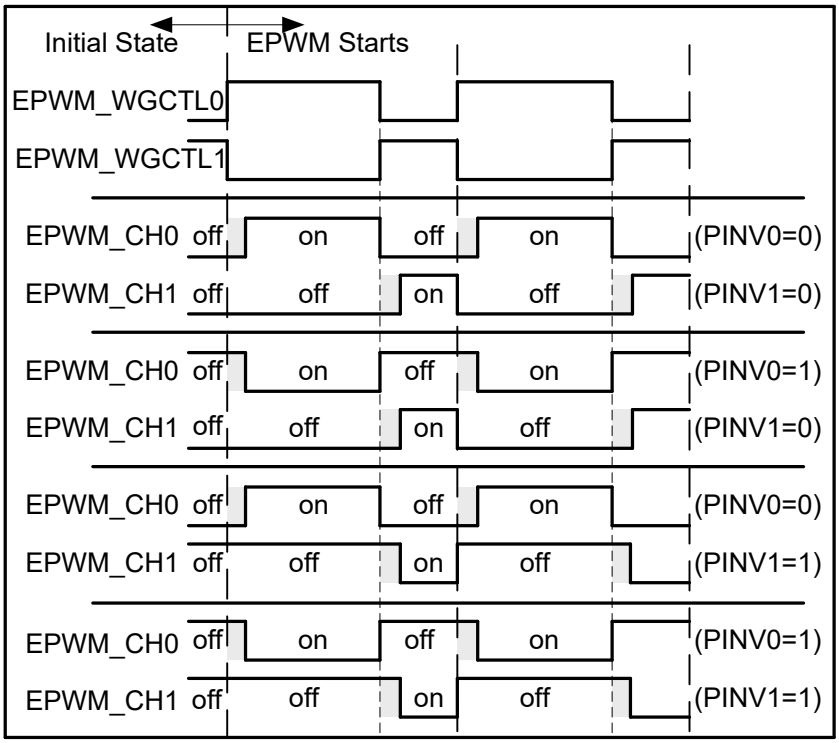


Figure 6.18-35 Brake System Fail Block Diagram

6.18.5.24 Polarity Control

Each EPWM port, from EPWM_CH0 to EPWM_CH5, has an independent polarity control module to configure the polarity of the active state of the EPWM output. By default, the EPWM output is active high. This implies the EPWM OFF state is low and ON state is high. This definition is variable through setting the EPWM Negative Polarity Control Register (EPWM_POLCTL), for each individual EPWM channel. Figure 6.18-36 shows the initial state before EPWM starting with different polarity settings.



Dead-time insertion; It is only effective in complementary mode
Note: PINVx: Negative Polarity control bits; It controls the EPWM output initial state and polarity, x denotes 0 or 1.

Figure 6.18-36 Initial State and Polarity Control with Rising Edge Dead-Time Insertion

6.18.5.25 EPWM Interrupt Generator

Interrupts for each EPWM show in Figure 6.18-38 and Figure 6.18-39.

The 1st EPWM interrupt (EPWM_INT) comes from EPWM complementary pair events. The counter can generate the Zero point Interrupt Flag ZIFn (EPWM_INTSTS0[5:0], n=0,1..5) and the Period point Interrupt Flag PIFn (EPWM_INTSTS0[13:8], n=0,1..5). When EPWM channel n's counter equals to the comparator value stored in EPWM_CMPDATn register, the different interrupt flags will be triggered depending on the counting direction. If the matching occurs at up–count direction, the Up Interrupt Flag CMPUIFn (EPWM_INTSTS0[21:16]) is set and if matching at the opposite direction, the Down Interrupt Flag CMPDIFn (EPWM_INTSTS0[29:24]) is set. If the corresponding interrupt enable bits are set, the trigger events will generates interrupt signals.

EPWM_INT can uses the EPWM_IFAn (n=0~5) register to accumulate the number of times that the interrupt flags have been triggered for each channel. By setting one of IFAEN (EPWM_IFAn[31], n=0~5) bit to 1 to enable accumulator, EPWM_INT will switch interrupt source from every event trigger interrupt

to trigger interrupt once every accumulate times.

By setting the IFASEL (EPWM_IFAn[29:28], n=0~5) bits, user can select one of the 4 interrupt sources to accumulate interrupt flag times for each channel, and the number of times interrupt flags will compare with IFACNT (EPWM_IFAn[15:0], n=0~5) bits. When interrupt accumulator equals IFACNT then set IFAIFn (EPWM_AINTSTS[n], n=0~5) bits as EPWM_INT signal if user enable IFAIENn (EPWM_AINTEN[n], n=0~5) bits. Accumulator interrupt of each channel can also be as request source of PDMA. Figure 6.18-37 is an example of channel 0 using EPWM_IFA0 register to output EPWM_INT once every IFCNT0+1 times interrupt events occurred.

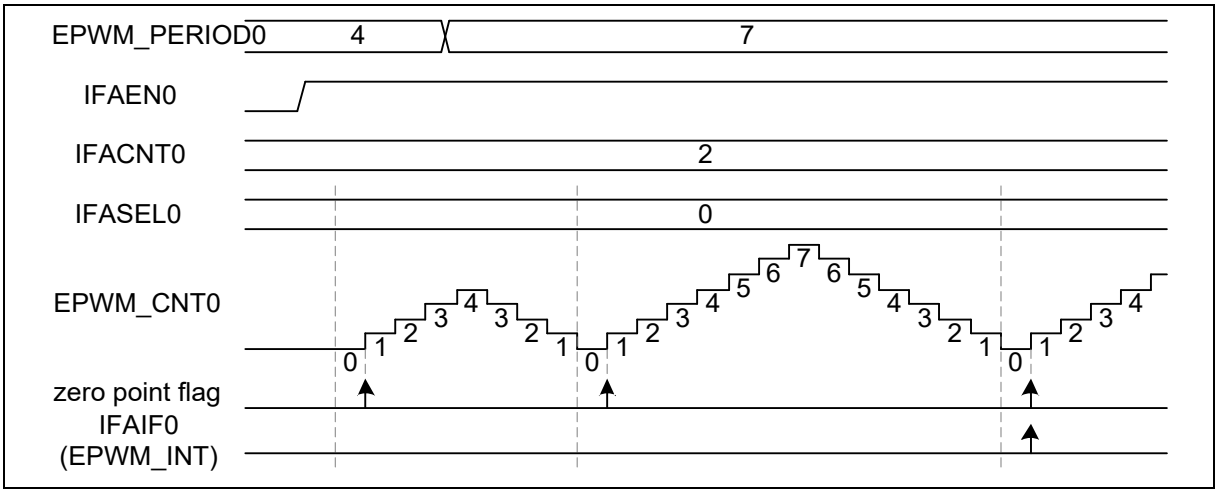


Figure 6.18-37 EPWMx_CH0 Accumulate Interrupt Waveform

The 2nd interrupt is the capture interrupt (CAP_INT). It shares the EPWM_INT vector in NVIC. The CAP_INT can be generated when the CAPRLIFn (EPWM_CAPIF[5:0]) flag is triggered and the Capture Rising Interrupt Enable bit CAPRIENn (EPWM_CAPIEN[5:0]) is set to 1. Or in the falling edge condition, the CAPFLIFn (EPWM_CAPIF[13:8]) flag can be triggered when the Capture Falling Interrupt Enable bit CAPFIENn (EPWM_CAPIEN[13:8]) is set to 1.

The 3rd is the brake interrupt (BRK_INT). The details of the BRK_INT is described in the EPWM Brake section. Figure 6.18-38 demonstrates the architecture of the EPWM interrupts.

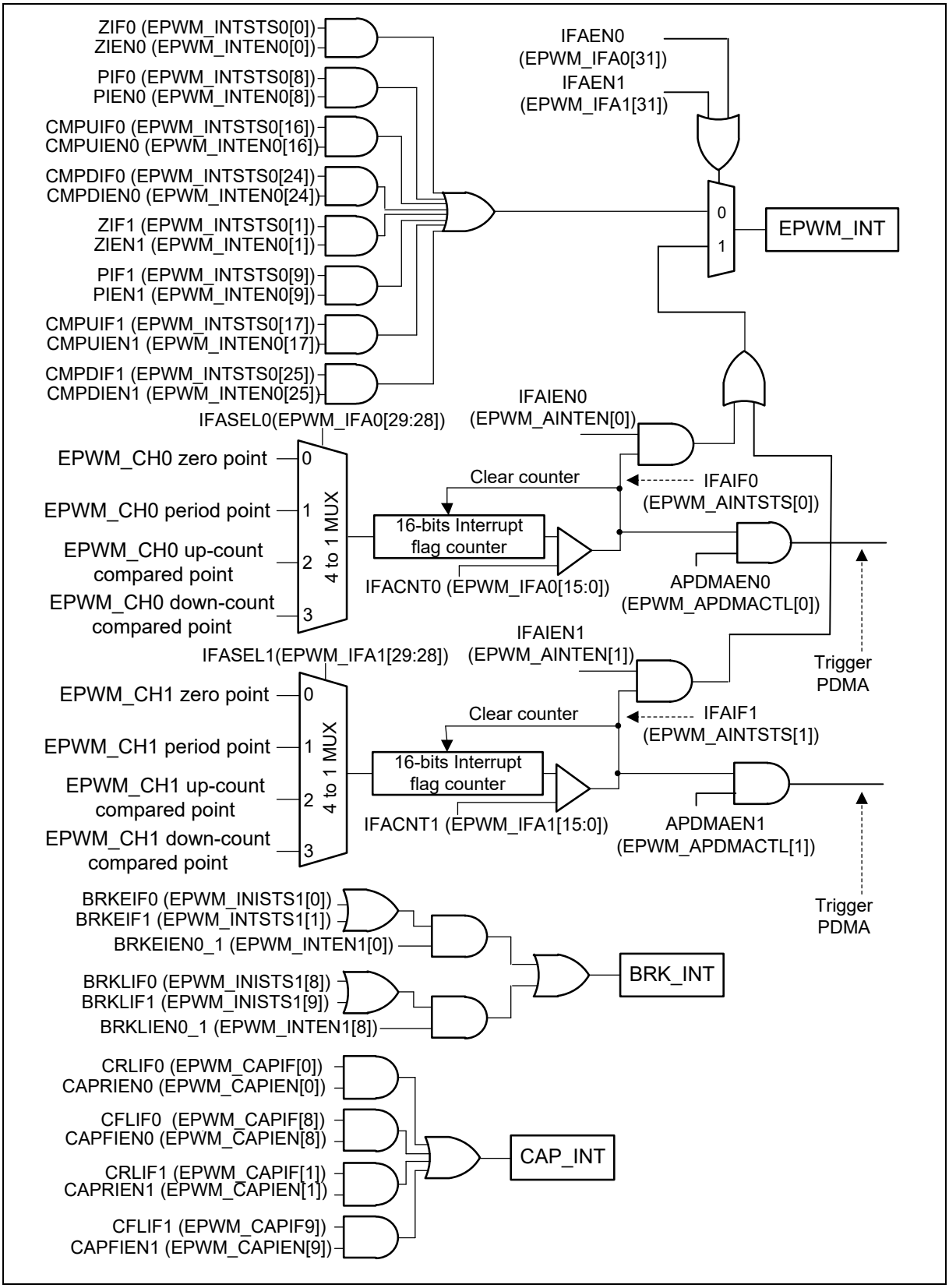


Figure 6.18-38 EPWMx_CH0 and EPWMx_CH1 Pair Interrupt Architecture Diagram

The 4th is Fault Detect Function interrupt (FLT_INT). It shares the EPWM_INT vector in NVIC. Figure 6.18-39 shows the FLT_INT can be generated when FDIENn (EPWM_FDIEN[n], n=0~5) is set to 1 and FDIFn (EPWM_FDSTS[n], n=0~5) is set to 1 by detecting output open short.

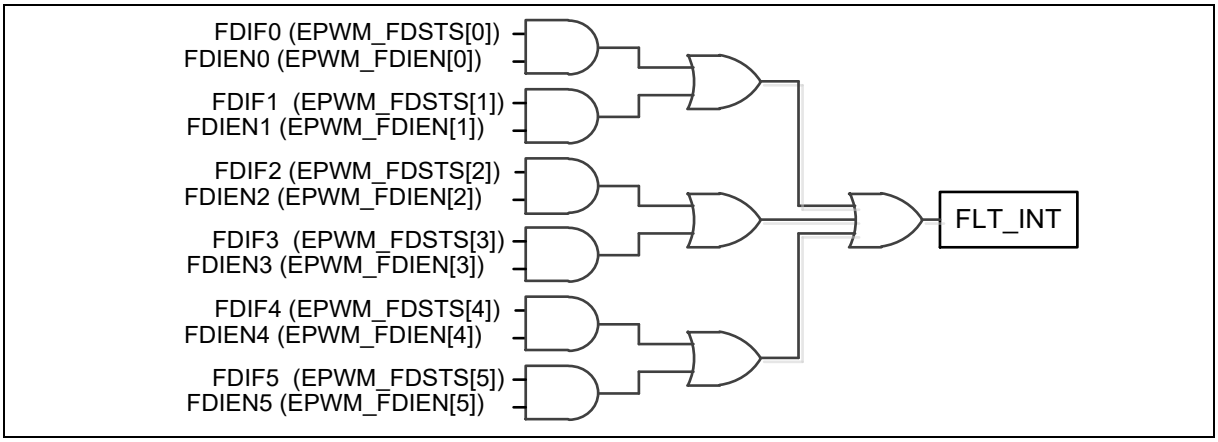


Figure 6.18-39 Fault Detect Function Interrupt Architecture Diagram

6.18.5.26 EPWM Trigger EADC Generator

EPWM can be one of the EADC conversion trigger source. Each EPWM pair channels share the same trigger source. Setting TRGSELn bit of EPWM_EADCTS0 and EPWM_EADCTS1 registers is to select the trigger sources, where TRGSELn bit is TRGSEL0, TRGSEL1, …, and TRGSEL5, which are located in EPWM_EADCTS0[3:0], EPWM_EADCTS0[11:8], EPWM_EADCTS0[19:16], EPWM_EADCTS0[27:24], EPWM_EADCTS1[3:0] and EPWM_EADCTS1[11:8], respectively. Setting TRGENn bit of EPWM_EADCTS0 and EPWM_EADCTS1 registers is to enable the trigger output to EADC, where TRGENn bit is TRGEN0, TRGEN1, …, TRGEN5, which are located in EPWM_EADCTS0[7], EPWM_EADCTS0[15], EPWM_EADCTS0[23], EPWM_EADCTS0[31], EPWM_EADCTS1[7] and EPWM_EADCTS1[15], respectively. The number n (n = 0,1, ..,5) denotes EPWM channel number.

There are 16 EPWM events that can be selected as the trigger source for one pair of channels as shown in Figure 6.18-40. Figure 6.18-41 shows trigger EADC block diagram with prescaler. By setting PSCENn (EPWM_EADCPSCCTL[n], n=0~5) bit to 1, EPWM will trigger EADC when the number of trigger events is equal to EADCPSCn+1 (n=0~5). Reading PSCNTn (n=0~5) can know how many events happened. If PSCENn (n=0~5) is 0, user can also write initial data to PSCNTn (n=0~5) and pre-scale counter will start from this data. If PSCENn (n=0~5) is set from 1 to 0, PSCNTn (n=0~5) will be reset to 0. Figure 6.18-42 is the trigger EADC timing waveform in the up-down counter type.

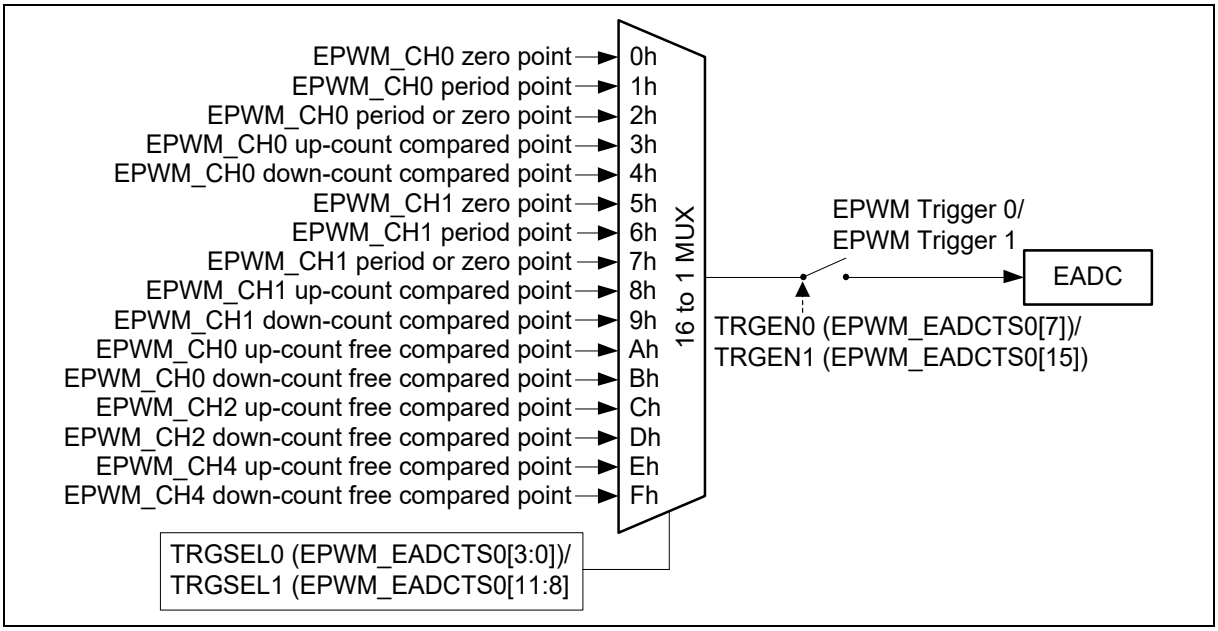


Figure 6.18-40 EPWMx_CH0 and EPWMx_CH1 Pair Trigger EADC Events

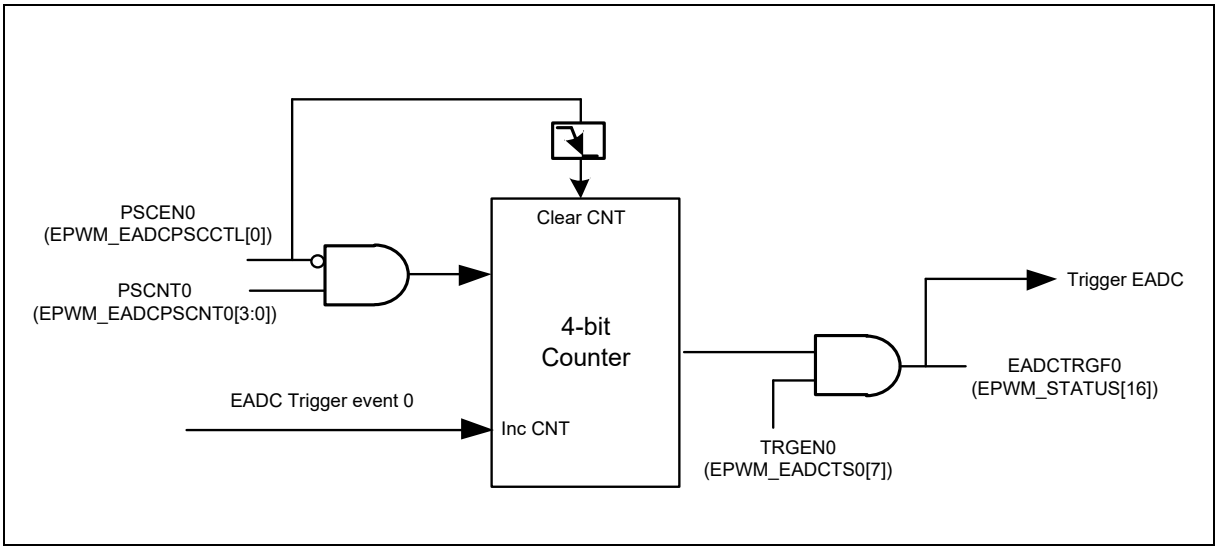


Figure 6.18-41 EPWMx_CH0 Trigger EADC Block Diagram

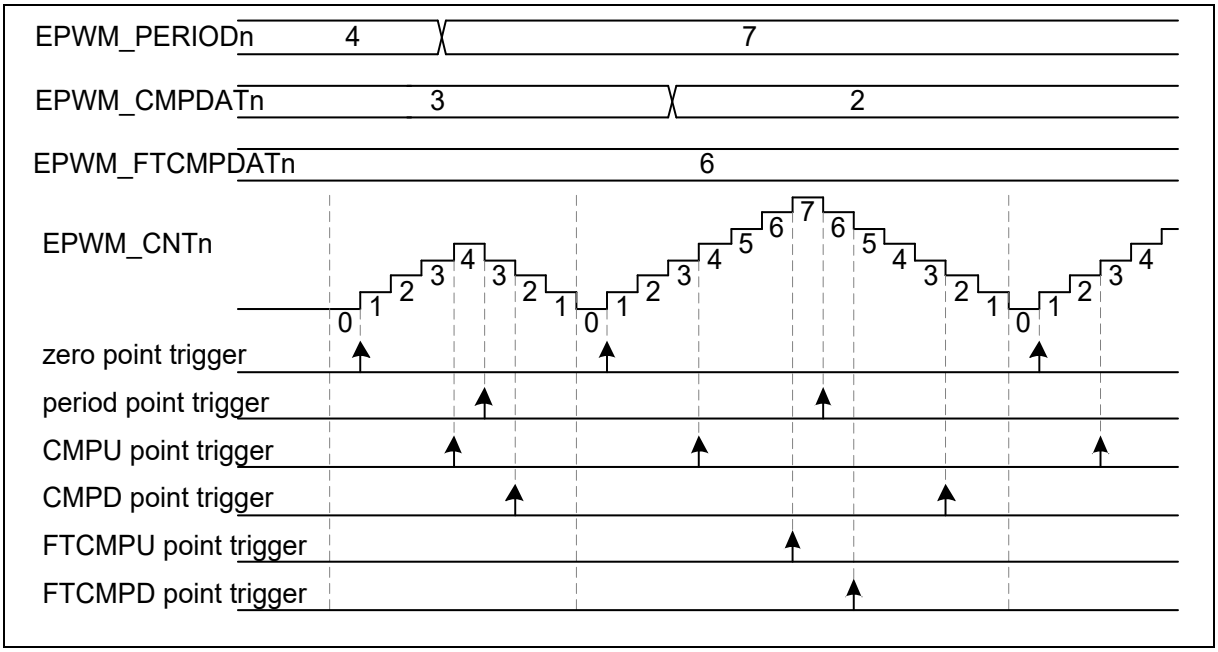


Figure 6.18-42 EPWM Trigger EADC in Up-Down Counter Type Timing Waveform

6.18.5.27 Capture Operation

The channels of the capture input and the EPWM output share the same pin and counter. The counter can operate in up or down counter type. The capture function will always latch the EPWM counter to the RCAPDATn (EPWM_RCAPDATn[15:0]) bits or the FCAPDATn (EPWM_FCAPDATn[15:0]) bits, if the input channel has a rising transition or a falling transition, respectively. The capture function will also generate an interrupt CAP_INT (using EPWM_INT vector) if the rising or falling latch occurs and the corresponding channel n's rising or falling interrupt enable bits are set, where the CAPRIENn (EPWM_CAPIEN[5:0]) bit is for the rising edge and the CAPFIENn (EPWM_CAPIEN[13:8]) bit is for the falling edge. When rising or falling latch occurs, the corresponding EPWM counter may be reloaded with the value of EPWM_PERIODn register, depending on the setting of RCRLDENn or FCRLDENn bits (where RCRLDENn and FCRLDENn are located at EPWM_CAPCTL[21:16] and EPWM_CAPCTL[29:24], respectively). Note that the corresponding GPIO pins must be configured as the capture function by enable the CAPINENn (EPWM_CAPINEN[5:0]) bits for the corresponding capture channel n. Figure 6.18-43 is the capture block diagram of channel 0.

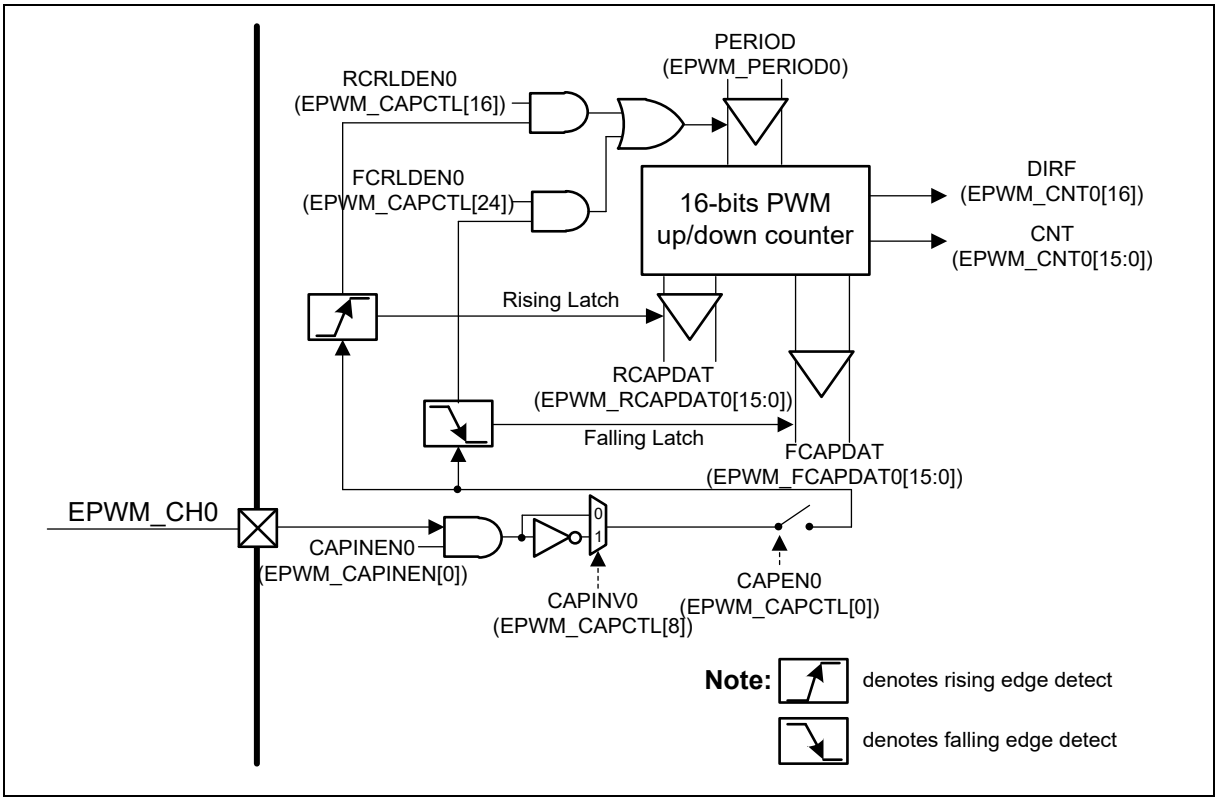


Figure 6.18-43 EPWM_CH0 Capture Block Diagram

Figure 6.18-44 illustrates the capture function timing. In this case, the capture counter is set as EPWM down counter type and the PERIOD is set to 8 so that the counter counts in the down direction, from 8 to 0. When detecting a falling edge at the capture input pin, the capture function latches the counter value to the EPWM_FCAPDATn register. When detecting the rising edge, it latches the counter value to the EPWM_RCAPDATn register. In this timing diagram, when the falling edge is detected at the first time, the capture function will reload the counter value from the PERIOD setting because the FCRLDENn bit is enabled. But at the second time, the falling edge does not result in a reload because of the disabled FCRLDENn bit. In this example, the counter also reloads at the rising edge of the capture input because the RCRLDENn bit is enabled, too.

Moreover, if the case is setup as the up counter type, the counter will reload the value zero and count up to the value PERIOD.

Figure 6.18-44 also illustrates the timing example for the interrupt and interrupt flag generation. When the rising edge at channel n is detected, the corresponding CRLIFn (EPWM_CAPIF[5:0]) bit is set by hardware. Similarly, a falling edge detection at chnnel n causes the corresponding CFLIFn (EPWM_CAPIF[13:8]) bit is set by hardware. CRLIFn and CFLIFn bits can be cleared by software by writing '1'. If the CRLIFn bit is set and the CAPRIENn bit is enabled, the capture function generates an interrupt. If the CFLIFn bit is set and the CAPFIENn bit is enabled, the interrupt also happens.

A condition which is not shown in this figure is: if the rising latch happens again when the CRLIFn bit is already set, the Over run status CRLIFOVn (EPWM_CAPSTS[5:0]) bit will be set to 1 by hardware to indicate the CRLIF flag overrunning. Also, if the falling latch happens again, the same hardware operation occurs for the CFLIF interrupt flag and the Over run status CFLIFOVn (EPWM_CAPSTS[13:8]).

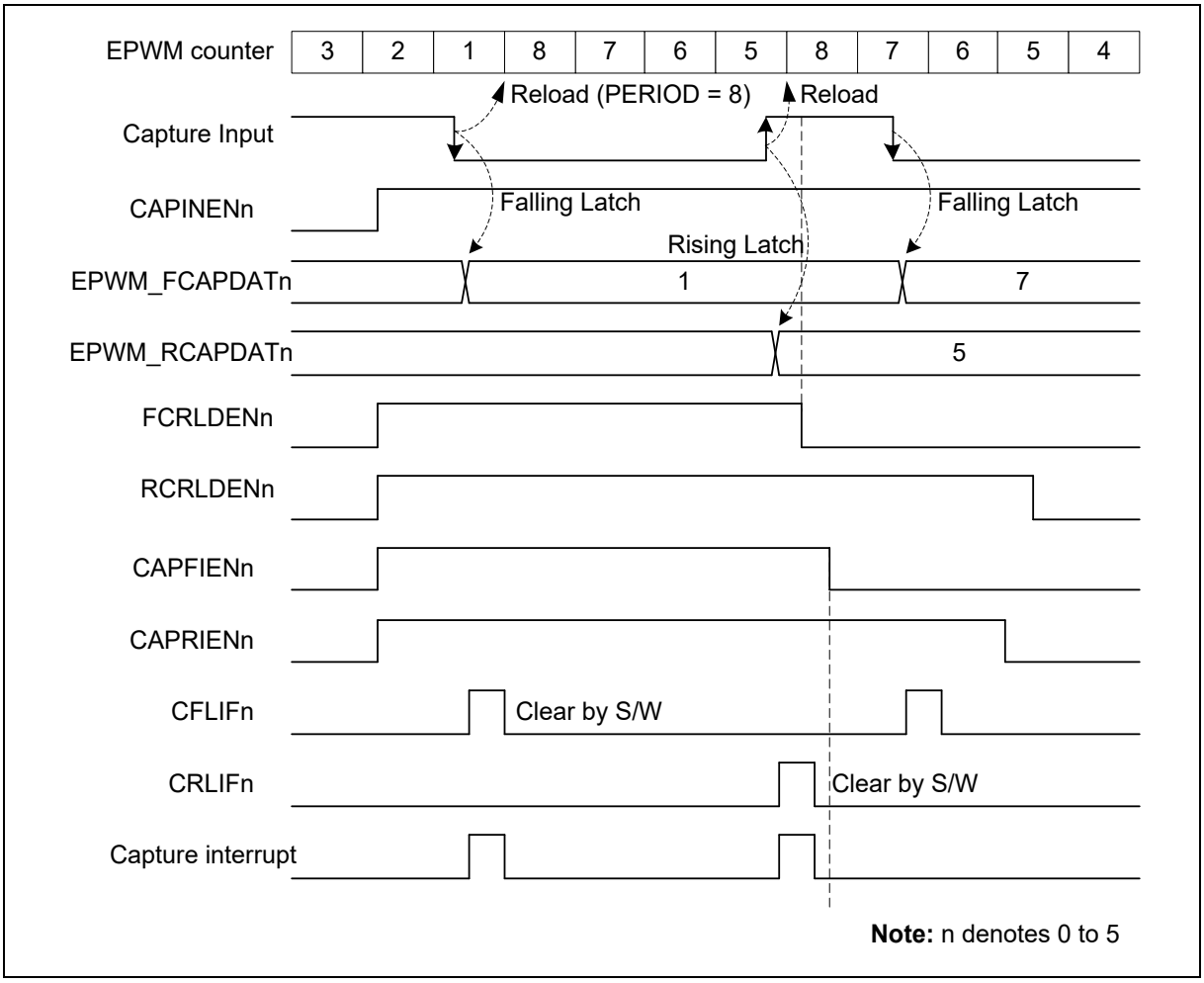


Figure 6.18-44 Capture Operation Waveform

The capture pulse width meeting the following conditions can be calculated according to the formula.

- The capture positive or negative pulse width is shorter than a counter period.
- The counter operates in down counter type.
- The counter can be reloaded by both falling and rising capture events through setting FCRLDENn and RCRLDENn bits of PWM_CAPCTL register to 1.

For the negative pulse case, the channel low pulse width is calculated as (EPWM_PERIODn + 1 - EPWM_RCAPDATn) EPWM counter time, where one EPWM counter time is (CLKPSC+1) * EPWMx_CLK clock time. In Figure 6.18-44, the low pulse width is 8+1-5 = 4 EPWM counter time.

For the positive pulse case, the channel high pulse width is calculated as (EPWM_PERIODn + 1 - EPWM_FCAPDATn) EPWM counter time, where one EPWM counter time is (CLKPSC+1) * EPWMx_CLK clock time. In Figure 6.18-44, the high pulse width is 8+1-7 = 2 EPWM counter time.

6.18.5.28 Capture PDMA Function

The EPWM module supports the PDMA transfer function when operating in the capture mode. When the corresponding PDMA enable bit CHENn_m (CHEN0_1 at EPWM_PDMACTL[0], CHEN2_3 at EPWM_PDMACTL[8] and CHEN4_5 at EPWM_PDMACTL[16], where n and m denote complement pair channels) is set, the capture module will issue a request to PDMA controller when the preceding capture event has happened. The PDMA controller will issue an acknowledgement to the capture module after it has read back the CAPBUF (EPWM_PDMACAPn_m[15:0], n, m denotes complement pair channels)

register in the capture module and has sent the register value to the memory. By setting CAPMODn_m (CAPMOD0_1 at EPWM_PDMACTL[2:1], CAPMOD2_3 at EPWM_PDMACTL[10:9] and CAPMOD4_5 at EPWM_PDMACTL[18:17]) bits, the PDMA can transfer the rising edge captured data or falling edge captured data or both of them to the memory. When using the PDMA to transfer both of the falling and rising edge data, remember to set CAPORDn_m (CAPORD0_1 at EPWM_PDMACTL[3], CAPORD2_3 at EPWM_PDMACTL[11] and CAPORD4_5 at EPWM_PDMACTL[19]) bit to decide the order of the transferred data (falling edge captured is first or rising edge captured first). The complement pair channels share a PDMA channel. Therefore, a selection bit CHSELn_m (CHSEL0_1 (EPWM_PDMACTL[4]), CHSEL2_3 (EPWM_PDMACTL[12]) and CHSEL4_5 (EPWM_PDMACTL[20])) bit is used to decide either channel n or channel m can be serviced by the PDMA channel.

Figure 6.18-45 is capture PDMA waveform. In this case, the CHSEL0_1 (EPWM_PDMACTL[4]) bit is set to 0. Hence the PDMA will service channel 0 for the capture data transfer. CAPMOD0_1 (EPWM_PDMACTL[2:1]) bits are set to 3. That means both of the rising and falling edge captured data will be transferred to the memory. The CAPORD0_1 (EPWM_PDMACTL[1]) bit is set to 1, so the rising edge data will be the first data to transfer and following is the falling edge data to transfer. As shown in Figure 6.18-45, the last assertions of the CRLIF0 and CFLIF0 signal have some overlap. The value of EPWM_RCAPDAT0 register is 11 will be loaded to EPWM_PDMACAP0_1 register to wait for transfer but not the EPWM_FCAPDAT0 value. The EPWM_PDMACAP0_1 register saves the data which will be transferred to the memory by PDMA. The HWDATA in this figure denotes the data which are being transferred by PDMA.

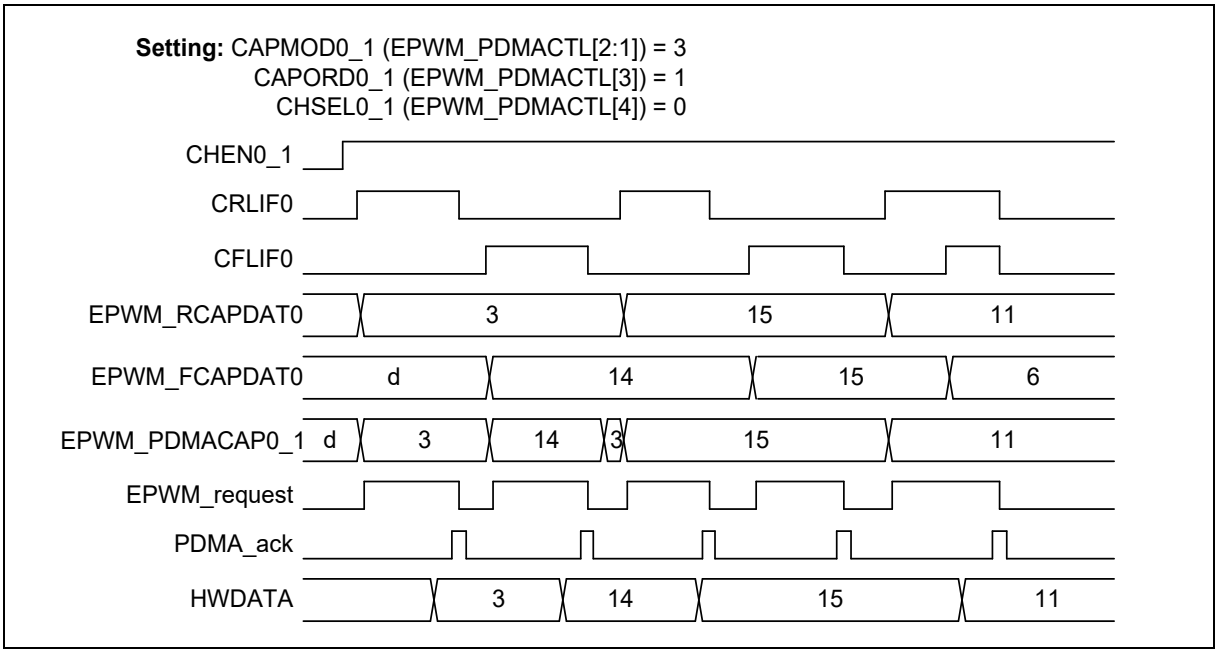


Figure 6.18-45 Capture PDMA Operation Waveform of Channel 0

6.18.5.29 Accumulator PDMA Function

The EPWM module supports the PDMA transfer function when accumulator interrupt happened. Figure 6.18-46 shows accumulator PDMA function architecture. When the corresponding PDMA enable bit APDMAENn (EPWM_APDMACTL[n], n=0~5) is set, accumulator module will send a request to PDMA controller when accumulator interrupt has happened, meaning that IFAIFn (EPWM_AINTSTS[n], n=0~5) is set 1. The PDMA controller will issue an acknowledge to accumulator after it has read memory data and send the data to the particular register (EPWM_PERIOD, etc.). So, user can uses this function to change accumulator interrupt frequency.

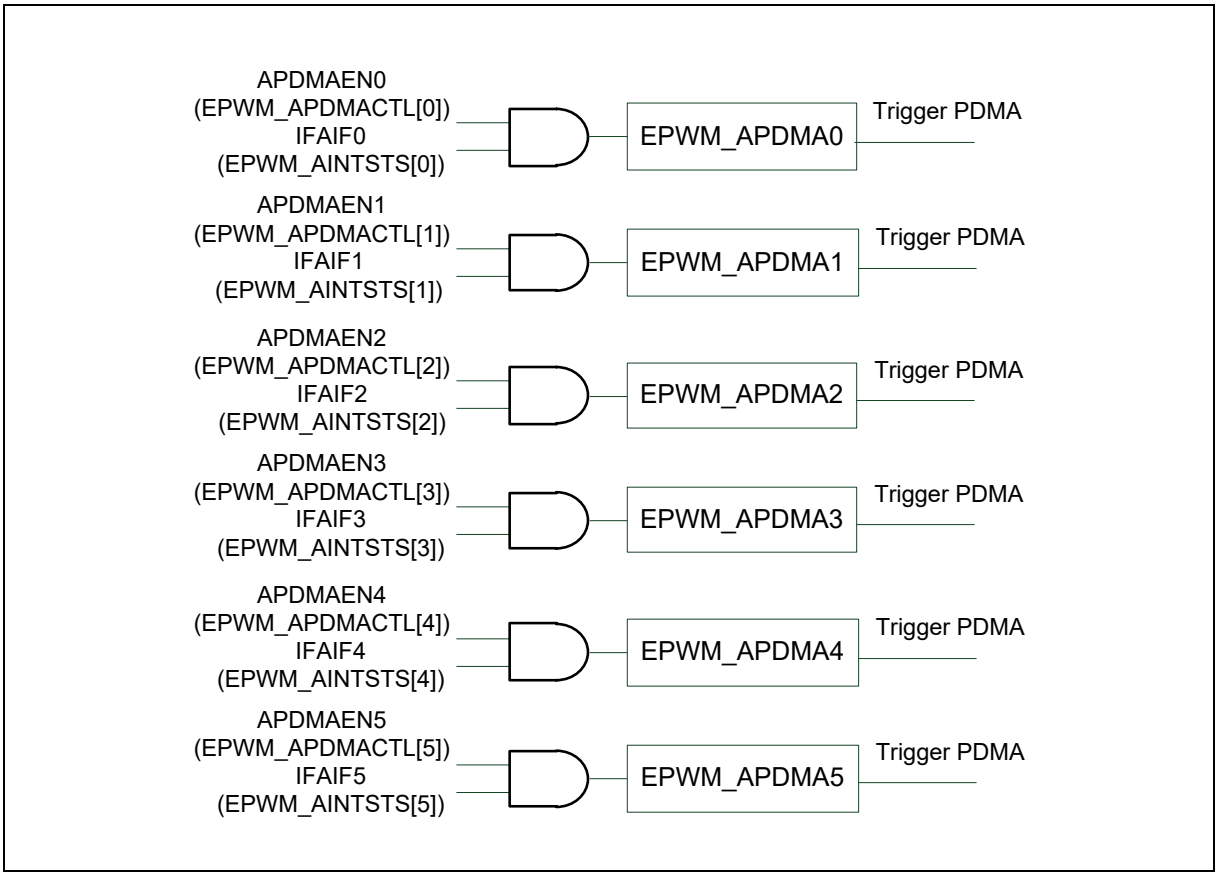


Figure 6.18-46 Accumulator PDMA Function Architecture

6.18.5.30 Accumulator Stop Mode

The EPWM module supports the Accumulator Stop Mode to stop counting when accumulator interrupt happened. Figure 6.18-47 shows Accumulator Stop Mode waveform. When the corresponding STPMOD (EPWM_IFAn[24], n=0~5) is set, accumulator module will disable CNT_ENn (EPWM_CNTEN[n], n=0~5) after the counter finishes the whole period.

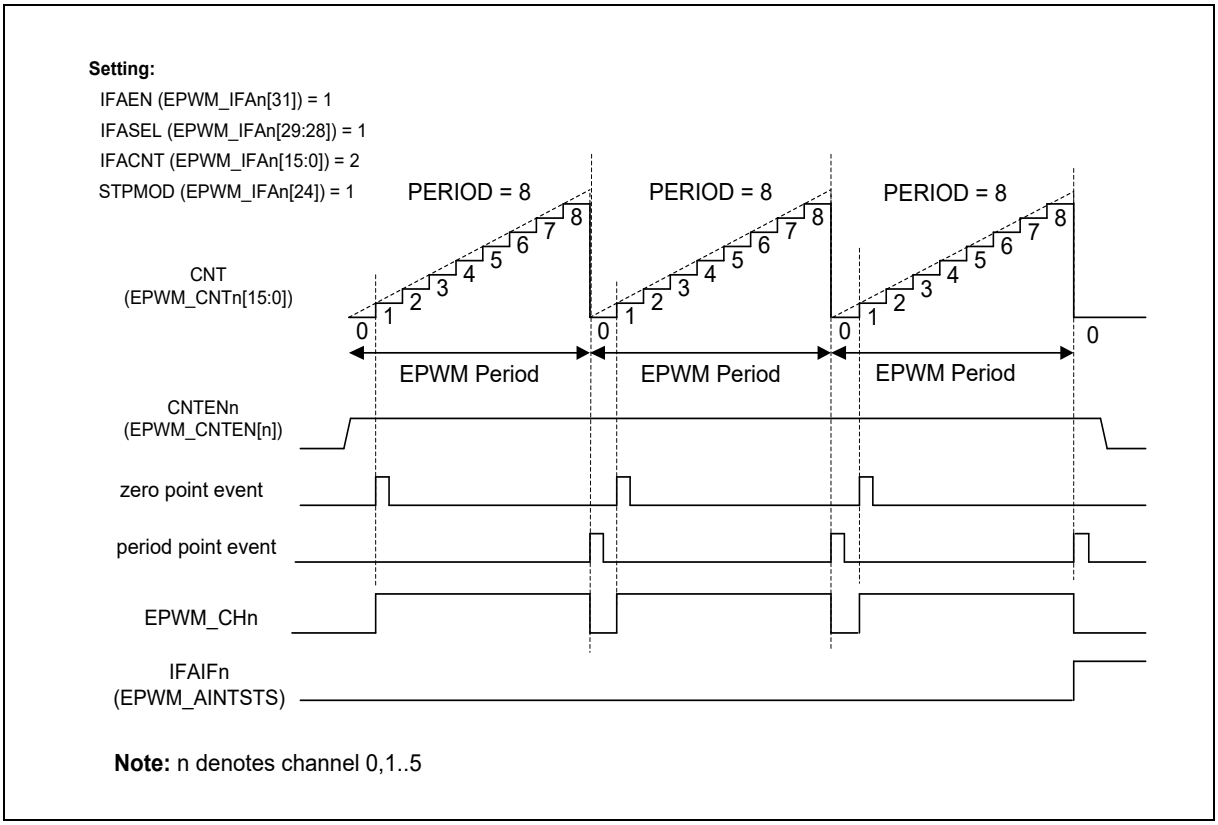


Figure 6.18-47 EPWM Accumulator Stop Mode Waveform

6.18.5.31 Fault Detect Function

The EPWM module supports Fault Detect Function to detect output short condition. Figure 6.18-48 shows fault detect function architecture. When the corresponding channel fault detect enable bit FDENn(EPWM_FDEN[n]) is set to 1, two statements will start detection. One is that EPWM output is from low to high or from high to low. The other is that POENn (EPWM_POEN[n], n=0~5) is from 0 to 1 or from 1 to 0.

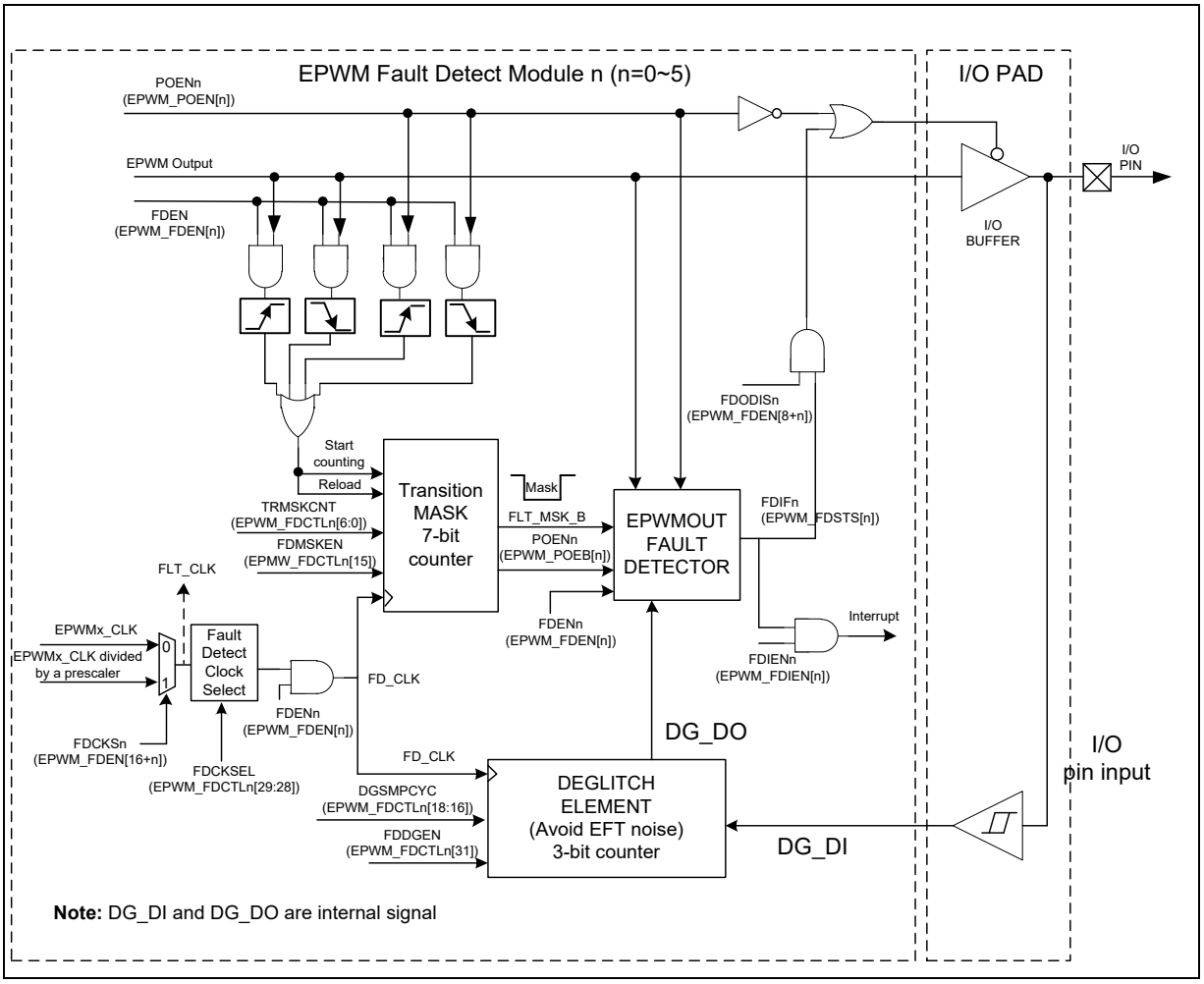


Figure 6.18-48 Fault Detect Function Architecture

Figure 6.18-49 shows Mask Function waveform example. Setting FDMSKEN (EPWM_FDCTLn[15], n=0~5) will enable the MASK Function. The detection will disable in the beginning until mask counter count from 0 to TRMSKCNT (EPWM_FDCTLn[6:0], n=0~5).

Figure 6.18-50 shows Deglitch function waveform example. Setting FDDGEN (EPWM_FDCTLn[31], n=0~5) will enable Deglitch function. The feedback signal will be sampled DGSMPCYC (EPWM_FDCTLn[18:16], n=0~5) +1 times and become valid input to be used for detection. FDCKSn (EPWM_FDEN[n], n=0~5) and FDCKSEL (EPWM_FDCTLn[29:28]) are clock setting for MASK and Deglitch Function. For more details, please see the register description.

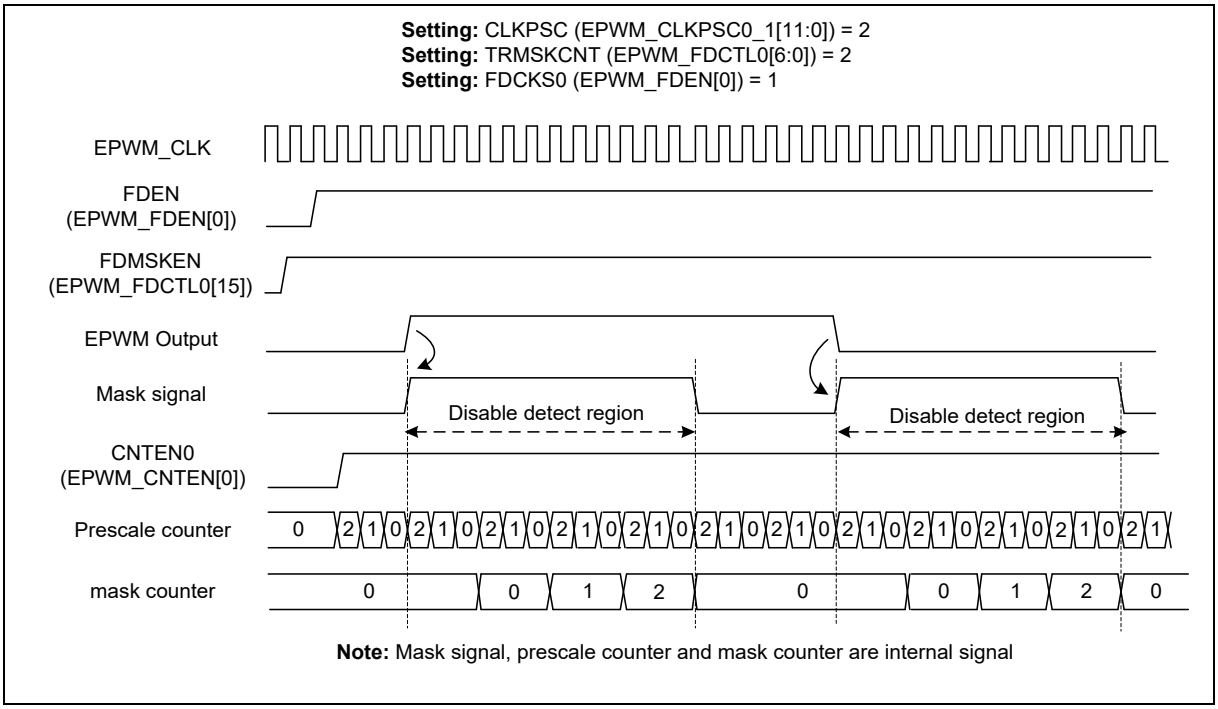


Figure 6.18-49 EPWM Mask Function Waveform

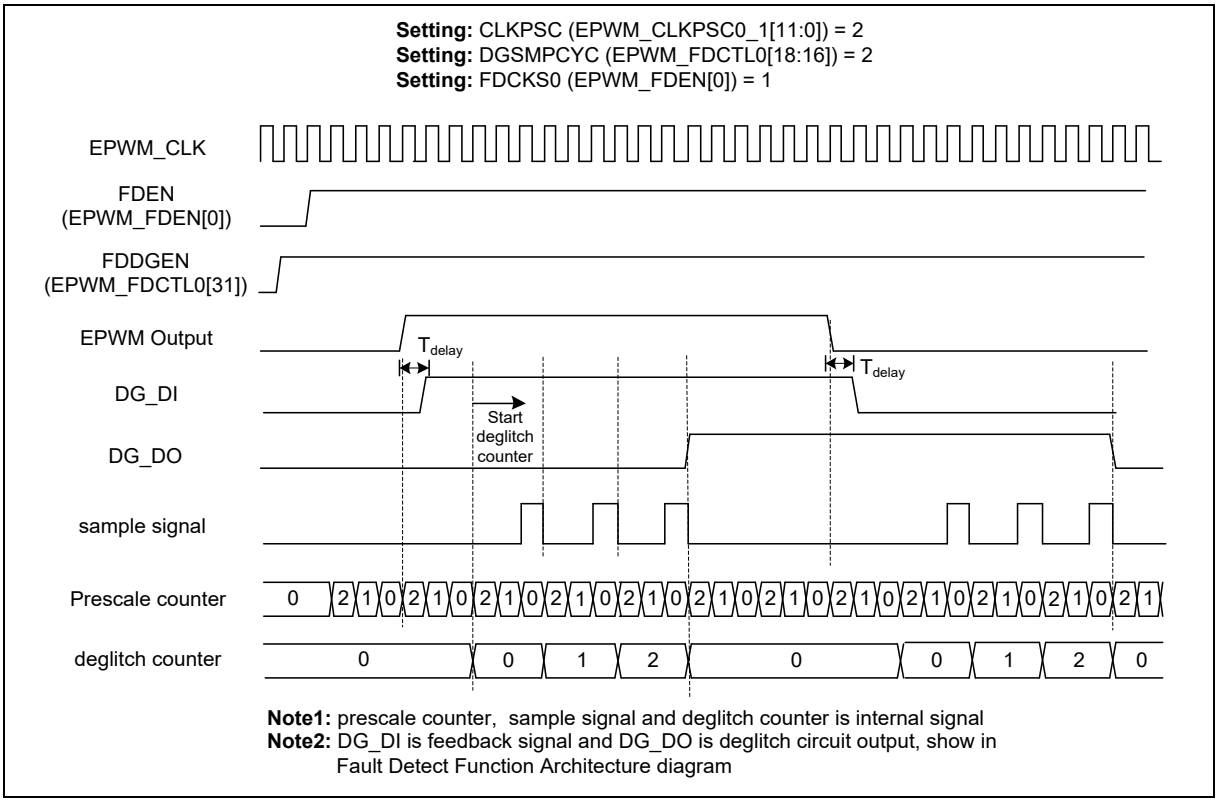


Figure 6.18-50 EPWM Deglitch Function Waveform

6.18.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
EPWM Base Address: **EPWMx_BA = 0x4058_0000+x*0x10000** **x=0,1,2**				
EPWM_CTL0	EPWMx_BA+0x00	R/W	EPWM Control Register 0	0x0000_0000
EPWM_CTL1	EPWMx_BA+0x04	R/W	EPWM Control Register 1	0x0000_0000
EPWM_SYNC	EPWMx_BA+0x08	R/W	EPWM Synchronization Register	0x0000_0000
EPWM_SWSYNC	EPWMx_BA+0x0C	R/W	EPWM Software Control Synchronization Register	0x0000_0000
EPWM_CLKSRC	EPWMx_BA+0x10	R/W	EPWM Clock Source Register	0x0000_0000
EPWM_CLKPSC0_1	EPWMx_BA+0x14	R/W	EPWM Clock Prescale Register 0/1	0x0000_0000
EPWM_CLKPSC2_3	EPWMx_BA+0x18	R/W	EPWM Clock Prescale Register 2/3	0x0000_0000
EPWM_CLKPSC4_5	EPWMx_BA+0x1C	R/W	EPWM Clock Prescale Register 4/5	0x0000_0000
EPWM_CNTEN	EPWMx_BA+0x20	R/W	EPWM Counter Enable Register	0x0000_0000
EPWM_CNTCLR	EPWMx_BA+0x24	R/W	EPWM Clear Counter Register	0x0000_0000
EPWM_LOAD	EPWMx_BA+0x28	R/W	EPWM Load Register	0x0000_0000
EPWM_PERIOD0	EPWMx_BA+0x30	R/W	EPWM Period Register 0	0x0000_0000
EPWM_PERIOD1	EPWMx_BA+0x34	R/W	EPWM Period Register 1	0x0000_0000
EPWM_PERIOD2	EPWMx_BA+0x38	R/W	EPWM Period Register 2	0x0000_0000
EPWM_PERIOD3	EPWMx_BA+0x3C	R/W	EPWM Period Register 3	0x0000_0000
EPWM_PERIOD4	EPWMx_BA+0x40	R/W	EPWM Period Register 4	0x0000_0000
EPWM_PERIOD5	EPWMx_BA+0x44	R/W	EPWM Period Register 5	0x0000_0000
EPWM_CMPDAT0	EPWMx_BA+0x50	R/W	EPWM Comparator Register 0	0x0000_0000
EPWM_CMPDAT1	EPWMx_BA+0x54	R/W	EPWM Comparator Register 1	0x0000_0000
EPWM_CMPDAT2	EPWMx_BA+0x58	R/W	EPWM Comparator Register 2	0x0000_0000
EPWM_CMPDAT3	EPWMx_BA+0x5C	R/W	EPWM Comparator Register 3	0x0000_0000
EPWM_CMPDAT4	EPWMx_BA+0x60	R/W	EPWM Comparator Register 4	0x0000_0000
EPWM_CMPDAT5	EPWMx_BA+0x64	R/W	EPWM Comparator Register 5	0x0000_0000
EPWM_DTCTL0_1	EPWMx_BA+0x70	R/W	EPWM Dead-time Control Register 0/1	0x0000_0000
EPWM_DTCTL2_3	EPWMx_BA+0x74	R/W	EPWM Dead-time Control Register 2/3	0x0000_0000
EPWM_DTCTL4_5	EPWMx_BA+0x78	R/W	EPWM Dead-time Control Register 4/5	0x0000_0000

EPWM_PHS0_1	EPWMx_BA+0x80	R/W	EPWM Counter Phase Register 0/1	0x0000_0000
EPWM_PHS2_3	EPWMx_BA+0x84	R/W	EPWM Counter Phase Register 2/3	0x0000_0000
EPWM_PHS4_5	EPWMx_BA+0x88	R/W	EPWM Counter Phase Register 4/5	0x0000_0000
EPWM_CNT0	EPWMx_BA+0x90	R	EPWM Counter Register 0	0x0000_0000
EPWM_CNT1	EPWMx_BA+0x94	R	EPWM Counter Register 1	0x0000_0000
EPWM_CNT2	EPWMx_BA+0x98	R	EPWM Counter Register 2	0x0000_0000
EPWM_CNT3	EPWMx_BA+0x9C	R	EPWM Counter Register 3	0x0000_0000
EPWM_CNT4	EPWMx_BA+0xA0	R	EPWM Counter Register 4	0x0000_0000
EPWM_CNT5	EPWMx_BA+0xA4	R	EPWM Counter Register 5	0x0000_0000
EPWM_WGCTL0	EPWMx_BA+0xB0	R/W	EPWM Generation Register 0	0x0000_0000
EPWM_WGCTL1	EPWMx_BA+0xB4	R/W	EPWM Generation Register 1	0x0000_0000
EPWM_MSKEN	EPWMx_BA+0xB8	R/W	EPWM Mask Enable Register	0x0000_0000
EPWM_MSK	EPWMx_BA+0xBC	R/W	EPWM Mask Data Register	0x0000_0000
EPWM_BNF	EPWMx_BA+0xC0	R/W	EPWM Brake Noise Filter Register	0x0000_0000
EPWM_FAILBRK	EPWMx_BA+0xC4	R/W	EPWM System Fail Brake Control Register	0x0000_0000
EPWM_BRKCTL0_1	EPWMx_BA+0xC8	R/W	EPWM Brake Edge Detect Control Register 0/1	0x0000_0000
EPWM_BRKCTL2_3	EPWMx_BA+0xCC	R/W	EPWM Brake Edge Detect Control Register 2/3	0x0000_0000
EPWM_BRKCTL4_5	EPWMx_BA+0xD0	R/W	EPWM Brake Edge Detect Control Register 4/5	0x0000_0000
EPWM_POLCTL	EPWMx_BA+0xD4	R/W	EPWM Pin Polar Inverse Register	0x0000_0000
EPWM_POEN	EPWMx_BA+0xD8	R/W	EPWM Output Enable Register	0x0000_0000
EPWM_SWBRK	EPWMx_BA+0xDC	W	EPWM Software Brake Control Register	0x0000_0000
EPWM_INTEN0	EPWMx_BA+0xE0	R/W	EPWM Interrupt Enable Register 0	0x0000_0000
EPWM_INTEN1	EPWMx_BA+0xE4	R/W	EPWM Interrupt Enable Register 1	0x0000_0000
EPWM_INTSTS0	EPWMx_BA+0xE8	R/W	EPWM Interrupt Flag Register 0	0x0000_0000
EPWM_INTSTS1	EPWMx_BA+0xEC	R/W	EPWM Interrupt Flag Register 1	0x0000_0000
EPWM_EADCTS0	EPWMx_BA+0xF8	R/W	EPWM Trigger EADC Source Select Register 0	0x0000_0000
EPWM_EADCTS1	EPWMx_BA+0xFC	R/W	EPWM Trigger EADC Source Select Register 1	0x0000_0000
EPWM_FTCMPDAT0_1	EPWMx_BA+0x100	R/W	EPWM Free Trigger Compare Register 0/1	0x0000_0000
EPWM_FTCMPDAT2_3	EPWMx_BA+0x104	R/W	EPWM Free Trigger Compare Register 2/3	0x0000_0000
EPWM_FTCMPDAT4_5	EPWMx_BA+0x108	R/W	EPWM Free Trigger Compare Register 4/5	0x0000_0000
EPWM_SSCTL	EPWMx_BA+0x110	R/W	EPWM Synchronous Start Control Register	0x0000_0000

EPWM_SSTRG	EPWMx_BA+0x114	W	EPWM Synchronous Start Trigger Register	0x0000_0000
EPWM_STATUS	EPWMx_BA+0x120	R/W	EPWM Status Register	0x0000_0000
EPWM_IFA0	EPWMx_BA+0x130	R/W	EPWM Interrupt Flag Accumulator Register 0	0x0000_0000
EPWM_IFA1	EPWMx_BA+0x134	R/W	EPWM Interrupt Flag Accumulator Register 1	0x0000_0000
EPWM_IFA2	EPWMx_BA+0x138	R/W	EPWM Interrupt Flag Accumulator Register 2	0x0000_0000
EPWM_IFA3	EPWMx_BA+0x13C	R/W	EPWM Interrupt Flag Accumulator Register 3	0x0000_0000
EPWM_IFA4	EPWMx_BA+0x140	R/W	EPWM Interrupt Flag Accumulator Register 4	0x0000_0000
EPWM_IFA5	EPWMx_BA+0x144	R/W	EPWM Interrupt Flag Accumulator Register 5	0x0000_0000
EPWM_AINTSTS	EPWMx_BA+0x150	R/W	EPWM Accumulator Interrupt Flag Register	0x0000_0000
EPWM_AINTEN	EPWMx_BA+0x154	R/W	EPWM Accumulator Interrupt Enable Register	0x0000_0000
EPWM_APDMACTL	EPWMx_BA+0x158	R/W	EPWM Accumulator PDMA Control Register	0x0000_0000
EPWM_FDEN	EPWMx_BA+0x160	R/W	EPWM Fault Detect Enable Register	0x0000_0000
EPWM_FDCTL0	EPWMx_BA+0x164	R/W	EPWM Fault Detect Control Register 0	0x0000_0000
EPWM_FDCTL1	EPWMx_BA+0x168	R/W	EPWM Fault Detect Control Register 1	0x0000_0000
EPWM_FDCTL2	EPWMx_BA+0x16C	R/W	EPWM Fault Detect Control Register 2	0x0000_0000
EPWM_FDCTL3	EPWMx_BA+0x170	R/W	EPWM Fault Detect Control Register 3	0x0000_0000
EPWM_FDCTL4	EPWMx_BA+0x174	R/W	EPWM Fault Detect Control Register 4	0x0000_0000
EPWM_FDCTL5	EPWMx_BA+0x178	R/W	EPWM Fault Detect Control Register 5	0x0000_0000
EPWM_FDIEN	EPWMx_BA+0x17C	R/W	EPWM Fault Detect Interrupt Enable Register	0x0000_0000
EPWM_FDSTS	EPWMx_BA+0x180	R/W	EPWM Fault Detect Interrupt Flag Register	0x0000_0000
EPWM_EADCPSCCTL	EPWMx_BA+0x184	R/W	EPWM Trigger EADC Prescale Control Register	0x0000_0000
EPWM_EADCPSC0	EPWMx_BA+0x188	R/W	EPWM Trigger EADC Prescale Register 0	0x0000_0000
EPWM_EADCPSC1	EPWMx_BA+0x18C	R/W	EPWM Trigger EADC Prescale Register 1	0x0000_0000
EPWM_EADCPSCNT0	EPWMx_BA+0x190	R/W	EPWM Trigger EADC Prescale Counter Register 0	0x0000_0000
EPWM_EADCPSCNT1	EPWMx_BA+0x194	R/W	EPWM Trigger EADC Prescale Counter Register 1	0x0000_0000
EPWM_CAPINEN	EPWMx_BA+0x200	R/W	EPWM Capture Input Enable Register	0x0000_0000
EPWM_CAPCTL	EPWMx_BA+0x204	R/W	EPWM Capture Control Register	0x0000_0000
EPWM_CAPSTS	EPWMx_BA+0x208	R	EPWM Capture Status Register	0x0000_0000
EPWM_RCAPDAT0	EPWMx_BA+0x20C	R	EPWM Rising Capture Data Register 0	0x0000_0000
EPWM_FCAPDAT0	EPWMx_BA+0x210	R	EPWM Falling Capture Data Register 0	0x0000_0000
EPWM_RCAPDAT1	EPWMx_BA+0x214	R	EPWM Rising Capture Data Register 1	0x0000_0000

EPWM_FCAPDAT1	EPWMx_BA+0x218	R	EPWM Falling Capture Data Register 1	0x0000_0000
EPWM_RCAPDAT2	EPWMx_BA+0x21C	R	EPWM Rising Capture Data Register 2	0x0000_0000
EPWM_FCAPDAT2	EPWMx_BA+0x220	R	EPWM Falling Capture Data Register 2	0x0000_0000
EPWM_RCAPDAT3	EPWMx_BA+0x224	R	EPWM Rising Capture Data Register 3	0x0000_0000
EPWM_FCAPDAT3	EPWMx_BA+0x228	R	EPWM Falling Capture Data Register 3	0x0000_0000
EPWM_RCAPDAT4	EPWMx_BA+0x22C	R	EPWM Rising Capture Data Register 4	0x0000_0000
EPWM_FCAPDAT4	EPWMx_BA+0x230	R	EPWM Falling Capture Data Register 4	0x0000_0000
EPWM_RCAPDAT5	EPWMx_BA+0x234	R	EPWM Rising Capture Data Register 5	0x0000_0000
EPWM_FCAPDAT5	EPWMx_BA+0x238	R	EPWM Falling Capture Data Register 5	0x0000_0000
EPWM_PDMACTL	EPWMx_BA+0x23C	R/W	EPWM PDMA Control Register	0x0000_0000
EPWM_PDMACAP0_1	EPWMx_BA+0x240	R	EPWM Capture Channel 01 PDMA Register	0x0000_0000
EPWM_PDMACAP2_3	EPWMx_BA+0x244	R	EPWM Capture Channel 23 PDMA Register	0x0000_0000
EPWM_PDMACAP4_5	EPWMx_BA+0x248	R	EPWM Capture Channel 45 PDMA Register	0x0000_0000
EPWM_CAPIEN	EPWMx_BA+0x250	R/W	EPWM Capture Interrupt Enable Register	0x0000_0000
EPWM_CAPIF	EPWMx_BA+0x254	R/W	EPWM Capture Interrupt Flag Register	0x0000_0000
EPWM_PBUF0	EPWMx_BA+0x304	R	EPWM PERIOD0 Buffer	0x0000_0000
EPWM_PBUF1	EPWMx_BA+0x308	R	EPWM PERIOD1 Buffer	0x0000_0000
EPWM_PBUF2	EPWMx_BA+0x30C	R	EPWM PERIOD2 Buffer	0x0000_0000
EPWM_PBUF3	EPWMx_BA+0x310	R	EPWM PERIOD3 Buffer	0x0000_0000
EPWM_PBUF4	EPWMx_BA+0x314	R	EPWM PERIOD4 Buffer	0x0000_0000
EPWM_PBUF5	EPWMx_BA+0x318	R	EPWM PERIOD5 Buffer	0x0000_0000
EPWM_CMPBUF0	EPWMx_BA+0x31C	R	EPWM CMPDAT0 Buffer	0x0000_0000
EPWM_CMPBUF1	EPWMx_BA+0x320	R	EPWM CMPDAT1 Buffer	0x0000_0000
EPWM_CMPBUF2	EPWMx_BA+0x324	R	EPWM CMPDAT2 Buffer	0x0000_0000
EPWM_CMPBUF3	EPWMx_BA+0x328	R	EPWM CMPDAT3 Buffer	0x0000_0000
EPWM_CMPBUF4	EPWMx_BA+0x32C	R	EPWM CMPDAT4 Buffer	0x0000_0000
EPWM_CMPBUF5	EPWMx_BA+0x330	R	EPWM CMPDAT5 Buffer	0x0000_0000
EPWM_CPSCBUF0_1	EPWMx_BA+0x334	R	EPWM CLKPSC0_1 Buffer	0x0000_0000
EPWM_CPSCBUF2_3	EPWMx_BA+0x338	R	EPWM CLKPSC2_3 Buffer	0x0000_0000
EPWM_CPSCBUF4_5	EPWMx_BA+0x33C	R	EPWM CLKPSC4_5 Buffer	0x0000_0000
EPWM_FTCBUF0_1	EPWMx_BA+0x340	R	EPWM FTCMPDAT0_1 Buffer	0x0000_0000

EPWM_FTCBUF2_3	EPWMx_BA+0x344	R	EPWM FTCMPDAT2_3 Buffer	0x0000_0000
EPWM_FTCBUF4_5	EPWMx_BA+0x348	R	EPWM FTCMPDAT4_5 Buffer	0x0000_0000
EPWM_FTCI	EPWMx_BA+0x34C	R/W	EPWM FTCMPDAT Indicator Register	0x0000_0000

6.18.7 Register Description

EPWM Control Register 0 (EPWM_CTL0)

Register	Offset	R/W	Description	Reset Value
EPWM_CTL0	EPWMx_BA+0x00	R/W	EPWM Control Register 0	0x0000_0000

31	30	29	28	27	26	25	24
DBGTRIOFF	DBGHALT	Reserved					GROUPEN
23	22	21	20	19	18	17	16
Reserved		IMMLDEN5	IMMLDEN4	IMMLDEN3	IMMLDEN2	IMMLDEN1	IMMLDEN0
15	14	13	12	11	10	9	8
Reserved		WINLDEN5	WINLDEN4	WINLDEN3	WINLDEN2	WINLDEN1	WINLDEN0
7	6	5	4	3	2	1	0
Reserved		CTRLD5	CTRLD4	CTRLD3	CTRLD2	CTRLD1	CTRLD0

Bits	Description	
[31]	**DBGTRIOFF**	**ICE Debug Mode Acknowledge Disable Bit (Write Protect)** 0 = ICE debug mode acknowledgement effects EPWM output. EPWM pin will be forced as tri-state while ICE debug mode acknowledged. 1 = ICE debug mode acknowledgement disabled. EPWM pin will keep output no matter ICE debug mode acknowledged or not. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register
[30]	**DBGHALT**	**ICE Debug Mode Counter Halt (Write Protect)** If counter halt is enabled, all EPWM counters will keep current value until exit ICE debug mode. 0 = ICE debug mode counter halt Disabled. 1 = ICE debug mode counter halt Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register
[29:26]	**Reserved**	Reserved.
[24]	**GROUPEN**	**Group Function Enable Bit** 0 = The output waveform of each EPWM channel are independent. 1 = Unify the EPWM_CH2 and EPWM_CH4 to output the same waveform as EPWM_CH0 and unify the EPWM_CH3 and EPWM_CH5 to output the same waveform as EPWM_CH1.
[23:22]	**Reserved**	Reserved.
[16+n] n=0,1..5	**IMMLDENn**	**Immediately Load Enable Bits** 0 = PERIOD will load to PBUF at the end point of each period. CMPDAT will load to CMPBUF at the end point or center point of each period by setting CTRLD bit. 1 = PERIOD/CMPDAT will load to PBUF and CMPBUF immediately when software update PERIOD/CMPDAT. **Note:** If IMMLDENn is enabled, WINLDENn and CTRLDn will be invalid.
[15:14]	**Reserved**	Reserved.

[8+n] n=0,1..5	**WINLDENn**	**Window Load Enable Bits** 0 = PERIOD will load to PBUF at the end point of each period. CMPDAT will load to CMPBUF at the end point or center point of each period by setting CTRLD bit. 1 = PERIOD will load to PBUF at the end point of each period. CMPDAT will load to CMPBUF at the end point of each period when valid reload window is set. The valid reload window is set by software write 1 to EPWM_LOAD register and cleared by hardware after load success.
[7:6]	**Reserved**	Reserved.
[n] n=0,1..5	**CTRLDn**	**Center Re-load** In up-down counter type, PERIOD will load to PBUF at the end point of each period. CMPDAT will load to CMPBUF at the center point of a period.

EPWM Control Register 1 (EPWM_CTL1)

Register	Offset	R/W	Description	Reset Value
EPWM_CTL1	EPWMx_BA+0x04	R/W	EPWM Control Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					OUTMODE4	OUTMODE2	OUTMODE0
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		CNTMODE5	CNTMODE4	CNTMODE3	CNTMODE2	CNTMODE1	CNTMODE0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				CNTTYPE5		CNTTYPE4	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNTTYPE3		CNTTYPE2		CNTTYPE1		CNTTYPE0	

Bits	Description	
[31:27]	**Reserved**	Reserved.
[24+n/2] n=0,2,4	**OUTMODEn**	**EPWM Output Mode** Each bit n controls the output mode of corresponding EPWM channel n. 0 = EPWM independent mode. 1 = EPWM complementary mode. **Note:** When operating in group function, these bits must all be set to the same mode.
[23:22]	**Reserved**	Reserved.
[16+n] n=0,1..5	**CNTMODEn**	**EPWM Counter Mode** 0 = Auto-reload mode. 1 = One-shot mode.
[15:12]	**Reserved**	Reserved.
[2n+1:2n] n=0,1..5	**CNTTYPEn**	**EPWM Counter Behavior Type** 00 = Up counter type (supported in capture mode). 01 = Down count type (supported in capture mode). 10 = Up-down counter type. 11 = Reserved.

EPWM Synchronization Register (EPWM_SYNC)

Register	Offset	R/W	Description	Reset Value
EPWM_SYNC	EPWMx_BA+0x08	R/W	EPWM Synchronization Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					PHSDIR4	PHSDIR2	PHSDIR0
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SINPINV	SFLTCNT			SFLTCSEL			SNFLTEN
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		SINSRC4		SINSRC2		SINSRC0	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					PHSEN4	PHSEN2	PHSEN0

Bits	Description	
[31:27]	**Reserved**	Reserved.
[24+n/2] n=0,2,4	**PHSDIRn**	**EPWM Phase Direction Control** 0 = Control EPWM counter count decrement after synchronizing. 1 = Control EPWM counter count increment after synchronizing.
[23]	**SINPINV**	**SYNC Input Pin Inverse** 0 = The state of pin SYNC is passed to the negative edge detector. 1 = The inversed state of pin SYNC is passed to the negative edge detector.
[22:20]	**SFLTCNT**	**SYNC Edge Detector Filter Count** The register bits control the counter number of edge detector.
[19:17]	**SFLTCSEL**	**SYNC Edge Detector Filter Clock Selection** 000 = Filter clock = HCLK. 001 = Filter clock = HCLK/2. 010 = Filter clock = HCLK/4. 011 = Filter clock = HCLK/8. 100 = Filter clock = HCLK/16. 101 = Filter clock = HCLK/32. 110 = Filter clock = HCLK/64. 111 = Filter clock = HCLK/128.
[16]	**SNFLTEN**	**EPWM0_SYNC_IN Noise Filter Enable Bits** 0 = Noise filter of input pin EPWM0_SYNC_IN Disabled. 1 = Noise filter of input pin EPWM0_SYNC_IN Enabled.
[15:14]	**Reserved**	Reserved.
[9+n:8+n] n=0,2,4	**SINSRCn**	**EPWM0_SYNC_IN Source Selection** 00 = Synchronize source from SYNC_IN or SWSYNC. 01 = Counter equal to 0. 10 = Counter equal to EPWM_CMPDATm, m denotes 1, 3, 5.

		11 = SYNC_OUT will not be generated.
[7:3]	**Reserved**	Reserved.
[n/2] n=0,2,4	**PHSENn**	**SYNC Phase Enable Bits** 0 = EPWM counter disabled to load PHS value. 1 = EPWM counter enabled to load PHS value.

EPWM Software Control Synchronization Register (EPWM_SWSYNC)

Register	Offset	R/W	Description	Reset Value
EPWM_SWSYNC	EPWMx_BA+0x0C	R/W	EPWM Software Control Synchronization Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					SWSYNC4	SWSYNC2	SWSYNC0

Bits	Description	
[31:3]	**Reserved**	Reserved.
[n/2] n=0,2,4	**SWSYNCn**	**Software SYNC Function** When SINSRCn (EPWM_SYNC[13:8]) is selected to 0, SYNC_OUT source comes from SYNC_IN or this bit.

EPWM Clock Source Register (EPWM_CLKSRC)

Register	Offset	R/W	Description	Reset Value
EPWM_CLKSRC	EPWMx_BA+0x10	R/W	EPWM Clock Source Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				ECLKSRC4			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				ECLKSRC2			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				ECLKSRC0			

Bits	Description	
[31:20]	**Reserved**	Reserved.
[19:16]	**ECLKSRC4**	**EPWM_CH45 External Clock Source Select** 0000 = EPWMx_CLK, x denotes 0, 1 or 2. 0001 = TIMER0 overflow. 0010 = TIMER1 overflow. 0011 = TIMER2 overflow. 0100 = TIMER3 overflow. 0101 = TIMER4 overflow. 0110 = TIMER5 overflow. 0111 = TIMER6 overflow. 1000 = TIMER7 overflow. 1001 = TIMER8 overflow. 1010 = TIMER9 overflow. 1011 = TIMER10 overflow. 1100 = TIMER11 overflow. Others = Reserved.
[15:12]	**Reserved**	Reserved.
[11:8]	**ECLKSRC2**	**EPWM_CH23 External Clock Source Select** 0000 = EPWMx_CLK, x denotes 0, 1 or 2. 0001 = TIMER0 overflow. 0010 = TIMER1 overflow. 0011 = TIMER2 overflow. 0100 = TIMER3 overflow. 0101 = TIMER4 overflow. 0110 = TIMER5 overflow. 0111 = TIMER6 overflow. 1000 = TIMER7 overflow. 1001 = TIMER8 overflow.

		1010 = TIMER9 overflow. 1011 = TIMER10 overflow. 1100 = TIMER11 overflow. Others = Reserved.
[7:4]	**Reserved**	Reserved.
[3:0]	**ECLKSRC0**	**EPWM_CH01 External Clock Source Select** 0000 = EPWMx_CLK, x denotes 0, 1 or 2. 0001 = TIMER0 overflow. 0010 = TIMER1 overflow. 0011 = TIMER2 overflow. 0100 = TIMER3 overflow. 0101 = TIMER4 overflow. 0110 = TIMER5 overflow. 0111 = TIMER6 overflow. 1000 = TIMER7 overflow. 1001 = TIMER8 overflow. 1010 = TIMER9 overflow. 1011 = TIMER10 overflow. 1100 = TIMER11 overflow. Others = Reserved.

EPWM Clock Prescale Register 0_1, 2_3, 4_5 (EPWM_CLKPSC0_1, 2_3, 4_5)

Register	Offset	R/W	Description	Reset Value
EPWM_CLKPSC0_1	EPWMx_BA+0x14	R/W	EPWM Clock Prescale Register 0/1	0x0000_0000
EPWM_CLKPSC2_3	EPWMx_BA+0x18	R/W	EPWM Clock Prescale Register 2/3	0x0000_0000
EPWM_CLKPSC4_5	EPWMx_BA+0x1C	R/W	EPWM Clock Prescale Register 4/5	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				CLKPSC			
7	6	5	4	3	2	1	0
CLKPSC							

Bits	Description	
[31:12]	**Reserved**	Reserved.
[11:0]	**CLKPSC**	**EPWM Counter Clock Prescale** The clock of EPWM counter is decided by clock prescaler. Each EPWM pair share one EPWM counter clock prescaler. The clock of EPWM counter is divided by (CLKPSC+ 1).

EPWM Counter Enable Register (EPWM_CNTEN)

Register	Offset	R/W	Description	Reset Value
EPWM_CNTEN	EPWMx_BA+0x20	R/W	EPWM Counter Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		CNTEN5	CNTEN4	CNTEN3	CNTEN2	CNTEN1	CNTEN0

Bits	Description	
[31:6]	**Reserved**	Reserved.
[n] n=0,1..5	**CNTENn**	**EPWM Counter Enable Bits** 0 = EPWM Counter and clock prescaler stop running. 1 = EPWM Counter and clock prescaler start running.

EPWM Clear Counter Register (EPWM_CNTCLR)

Register	Offset	R/W	Description	Reset Value
EPWM_CNTCLR	EPWMx_BA+0x24	R/W	EPWM Clear Counter Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		CNTCLR5	CNTCLR4	CNTCLR3	CNTCLR2	CNTCLR1	CNTCLR0

Bits	Description	
[31:6]	**Reserved**	Reserved.
[n] n=0,1..5	**CNTCLRn**	**Clear EPWM Counter Control Bit** It is automatically cleared by hardware. Each bit n controls the corresponding EPWM channel n. 0 = No effect. 1 = Clear 16-bit EPWM counter to 0000H.

EPWM Load Register (EPWM_LOAD)

Register	Offset	R/W	Description	Reset Value
EPWM_LOAD	EPWMx_BA+0x28	R/W	EPWM Load Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**LOAD5**	**LOAD4**	**LOAD3**	**LOAD2**	**LOAD1**	**LOAD0**

Bits	Description	
[31:6]	**Reserved**	Reserved.
[n] n=0,1..5	**LOADn**	**Re-load EPWM Comparator Register Control Bit** This bit is software write, hardware clear when current EPWM period ended. Write Operation: 0 = No effect. 1 = Set load window of window loading mode. Read Operation: 0 = No load window is set. 1 = Load window is set. **Note:** This bit is only used in window loading mode, WINLDENn(EPWM_CTL0[13:8]) = 1.

EPWM Period Register 0~5 (EPWM_PERIOD0~5)

Register	Offset	R/W	Description	Reset Value
EPWM_PERIOD0	EPWMx_BA+0x30	R/W	EPWM Period Register 0	0x0000_0000
EPWM_PERIOD1	EPWMx_BA+0x34	R/W	EPWM Period Register 1	0x0000_0000
EPWM_PERIOD2	EPWMx_BA+0x38	R/W	EPWM Period Register 2	0x0000_0000
EPWM_PERIOD3	EPWMx_BA+0x3C	R/W	EPWM Period Register 3	0x0000_0000
EPWM_PERIOD4	EPWMx_BA+0x40	R/W	EPWM Period Register 4	0x0000_0000
EPWM_PERIOD5	EPWMx_BA+0x44	R/W	EPWM Period Register 5	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PERIOD							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PERIOD							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**PERIOD**	**EPWM Period Register** Up-Count mode: In this mode, EPWM counter counts from 0 to PERIOD, and restarts from 0. EPWM period time = (PERIOD+1) * (CLKPSC+1) * EPWM_CLK . Down-Count mode: In this mode, EPWM counter counts from PERIOD to 0, and restarts from PERIOD. EPWM period time = (PERIOD+1) * (CLKPSC+1) * EPWM_CLK . Up-Down-Count mode: In this mode, EPWM counter counts from 0 to PERIOD, then decrements to 0 and repeats again. EPWM period time = 2 * PERIOD * (CLKPSC+1) * EPWM_CLK.

EPWM Comparator Register 0~5 (EPWM_CMPDAT0~5)

Register	Offset	R/W	Description	Reset Value
EPWM_CMPDAT0	EPWMx_BA+0x50	R/W	EPWM Comparator Register 0	0x0000_0000
EPWM_CMPDAT1	EPWMx_BA+0x54	R/W	EPWM Comparator Register 1	0x0000_0000
EPWM_CMPDAT2	EPWMx_BA+0x58	R/W	EPWM Comparator Register 2	0x0000_0000
EPWM_CMPDAT3	EPWMx_BA+0x5C	R/W	EPWM Comparator Register 3	0x0000_0000
EPWM_CMPDAT4	EPWMx_BA+0x60	R/W	EPWM Comparator Register 4	0x0000_0000
EPWM_CMPDAT5	EPWMx_BA+0x64	R/W	EPWM Comparator Register 5	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CMP							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CMP							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**CMP**	**EPWM Comparator Register** CMP is used to compare with CNT (EPWM_CNTn[15:0]) bits to generate EPWM waveform, interrupt and trigger EADC. In independent mode, CMPDAT0~5 is denoted as 6 independent EPWM_CH0~5 compared point. In complementary mode, CMPDAT0, 2, 4 is denoted as the first compared point, and CMPDAT1, 3, 5 is denoted as the second compared point for the corresponding 3 complementary pairs EPWM_CH0 and EPWM_CH1, EPWM_CH2 and EPWM_CH3, EPWM_CH4 and EPWM_CH5.

EPWM Dead-time Control Register 0_1, 2_3, 4_5 (EPWM_DTCTL0_1, 2_3, 4_5)

Register	Offset	R/W	Description	Reset Value
EPWM_DTCTL0_1	EPWMx_BA+0x70	R/W	EPWM Dead-time Control Register 0/1	0x0000_0000
EPWM_DTCTL2_3	EPWMx_BA+0x74	R/W	EPWM Dead-time Control Register 2/3	0x0000_0000
EPWM_DTCTL4_5	EPWMx_BA+0x78	R/W	EPWM Dead-time Control Register 4/5	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							DTCKSEL
23	22	21	20	19	18	17	16
Reserved							DTEN
15	14	13	12	11	10	9	8
Reserved				DTCNT			
7	6	5	4	3	2	1	0
DTCNT							

Bits	Description	
[31:25]	**Reserved**	Reserved.
[24]	**DTCKSEL**	**Dead-time Clock Select (Write Protect)** 0 = Dead-time clock source from EPWM_CLK. 1 = Dead-time clock source from prescaler output. **Note:** This bit is write protected. Refer to REGWRPROT register.
[23:17]	**Reserved**	Reserved.
[16]	**DTEN**	**Enable Dead-time Insertion for EPWM Pair (Write Protect)** Dead-time insertion is only active when this pair of complementary EPWM is enabled. If dead- time insertion is inactive, the outputs of pin pair are complementary without any delay. 0 = Dead-time insertion disabled on the pin pair. 1 = Dead-time insertion enabled on the pin pair. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register
[15:12]	**Reserved**	Reserved.
[11:0]	**DTCNT**	**Dead-time Counter (Write Protect)** The dead-time can be calculated from the following formula: DTCKSEL=0: Dead-time = (DTCNT[11:0]+1) * EPWM_CLK period. DTCKSEL=1: Dead-time = (DTCNT[11:0]+1) * EPWM_CLK period * (CLKPSC+1). **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register

EPWM Counter Phase Register 0_1, 2_3, 4_5 (EPWM_PHS0_1, 2_3, 4_5)

Register	Offset	R/W	Description	Reset Value
EPWM_PHS0_1	EPWMx_BA+0x80	R/W	EPWM Counter Phase Register 0/1	0x0000_0000
EPWM_PHS2_3	EPWMx_BA+0x84	R/W	EPWM Counter Phase Register 2/3	0x0000_0000
EPWM_PHS4_5	EPWMx_BA+0x88	R/W	EPWM Counter Phase Register 4/5	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PHS							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PHS							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**PHS**	**EPWM Synchronous Start Phase Bits** PHS determines the EPWM synchronous start phase value. These bits are only used for synchronous function.

EPWM Counter Register 0~5 (EPWM_CNT0~5)

Register	Offset	R/W	Description	Reset Value
EPWM_CNT0	EPWMx_BA+0x90	R	EPWM Counter Register 0	0x0000_0000
EPWM_CNT1	EPWMx_BA+0x94	R	EPWM Counter Register 1	0x0000_0000
EPWM_CNT2	EPWMx_BA+0x98	R	EPWM Counter Register 2	0x0000_0000
EPWM_CNT3	EPWMx_BA+0x9C	R	EPWM Counter Register 3	0x0000_0000
EPWM_CNT4	EPWMx_BA+0xA0	R	EPWM Counter Register 4	0x0000_0000
EPWM_CNT5	EPWMx_BA+0xA4	R	EPWM Counter Register 5	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							DIRF
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CNT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNT							

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**DIRF**	**EPWM Direction Indicator Flag (Read Only)** 0 = Counter is counting down. 1 = Counter is counting up.
[15:0]	**CNT**	**EPWM Data Register (Read Only)** User can monitor CNT to know the current value in 16-bit period counter.

EPWM Generation Register 0 (EPWM_WGCTL0)

Register	Offset	R/W	Description	Reset Value
EPWM_WGCTL0	EPWMx_BA+0xB0	R/W	EPWM Generation Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				PRDPCTL5		PRDPCTL4	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PRDPCTL3		PRDPCTL2		PRDPCTL1		PRDPCTL0	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				ZPCTL5		ZPCTL4	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ZPCTL3		ZPCTL2		ZPCTL1		ZPCTL0	

Bits	Description	
[31:28]	**Reserved**	Reserved.
[17+2n:16+2n] n=0,1..5	**PRDPCTLn**	**EPWM Period Point Control** EPWM can control output level when EPWM counter counts to (PERIODn+1). 00 = Do nothing. 01 = EPWM period (center) point output Low. 10 = EPWM period (center) point output High. 11 = EPWM period (center) point output Toggle. **Note:** This bit is center point control when EPWM counter operating in up-down counter type.
[15:12]	**Reserved**	Reserved.
[1+2n:2n] n=0,1..5	**ZPCTLn**	**EPWM Zero Point Control** EPWM can control output level when EPWM counter counts to 0. 00 = Do nothing. 01 = EPWM zero point output Low. 10 = EPWM zero point output High. 11 = EPWM zero point output Toggle.

EPWM Generation Register 1 (EPWM_WGCTL1)

Register	Offset	R/W	Description	Reset Value
EPWM_WGCTL1	EPWMx_BA+0xB4	R/W	EPWM Generation Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				CMPDCTL5		CMPDCTL4	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CMPDCTL3		CMPDCTL2		CMPDCTL1		CMPDCTL0	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				CMPUCTL5		CMPUCTL4	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CMPUCTL3		CMPUCTL2		CMPUCTL1		CMPUCTL0	

Bits	Description	
[31:28]	**Reserved**	Reserved.
[17+2n:16+2n] n=0,1..5	**CMPDCTLn**	**EPWM Compare Down Point Control** EPWM can control output level when EPWM counter counts down to CMPDAT. 00 = Do nothing. 01 = EPWM compare down point output Low. 10 = EPWM compare down point output High. 11 = EPWM compare down point output Toggle. **Note:** In complementary mode, CMPDCTL1, 3, 5 is used as another CMPDCTL for channel 0, 2, 4.
[15:12]	**Reserved**	Reserved.
[1+2n:2n] n=0,1..5	**CMPUCTLn**	**EPWM Compare Up Point Control** EPWM can control output level when EPWM counter counts up to CMPDAT. 00 = Do nothing. 01 = EPWM compare up point output Low. 10 = EPWM compare up point output High. 11 = EPWM compare up point output Toggle. **Note:** In complementary mode, CMPUCTL1, 3, 5 is used as another CMPUCTL for channel 0, 2, 4.

EPWM Mask Enable Register (EPWM_MSKEN)

Register	Offset	R/W	Description	Reset Value
EPWM_MSKEN	EPWMx_BA+0xB8	R/W	EPWM Mask Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		MSKEN5	MSKEN4	MSKEN3	MSKEN2	MSKEN1	MSKEN0

Bits	Description	
[31:6]	**Reserved**	Reserved.
[n] n=0,1..5	**MSKENn**	**EPWM Mask Enable Bits** The EPWM output signal will be masked when this bit is enabled. The corresponding EPWM channel n will output MSKDATn (EPWM_MSK[5:0]) data. 0 = EPWM output signal is non-masked. 1 = EPWM output signal is masked and output MSKDATn data.

EPWM Mask DATA Register (EPWM_MSK)

Register	Offset	R/W	Description	Reset Value
EPWM_MSK	EPWMx_BA+0xBC	R/W	EPWM Mask Data Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		MSKDAT5	MSKDAT4	MSKDAT3	MSKDAT2	MSKDAT1	MSKDAT0

Bits	Description	
[31:6]	**Reserved**	Reserved.
[n] n=0,1..5	**MSKDATn**	**EPWM Mask Data Bit** This data bit control the state of EPWMn output pin, if corresponding mask function is enabled. 0 = Output logic low to EPWM channel n. 1 = Output logic high to EPWM channel n.

EPWM Brake Noise Filter Register (EPWM_BNF)

Register	Offset	R/W	Description	Reset Value
EPWM_BNF	EPWMx_BA+0xC0	R/W	EPWM Brake Noise Filter Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						BK1SRC	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						BK0SRC	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
BRK1PINV	BRK1FCNT			BRK1NFSEL			BRK1NFEN
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BRK0PINV	BRK0FCNT			BRK0NFSEL			BRK0NFEN

Bits	Description	
[31:26]	**Reserved**	Reserved.
[25:24]	**BK1SRC**	**Brake 1 Pin Source Select** For EPWM0 setting: 0 = Brake 1 pin source come from EPWM0_BRAKE1. 1 = Brake 1 pin source come from EPWM1_BRAKE1. 2 = Brake 1 pin source come from EPWM2_BRAKE1. For EPWM1 setting: 0 = Brake 1 pin source come from EPWM1_BRAKE1. 1 = Brake 1 pin source come from EPWM0_BRAKE1. 2 = Brake 1 pin source come from EPWM2_BRAKE1. For EPWM2 setting: 0 = Brake 1 pin source come from EPWM2_BRAKE1. 1 = Brake 1 pin source come from EPWM0_BRAKE1. 2 = Brake 1 pin source come from EPWM1_BRAKE1.
[23:18]	**Reserved**	Reserved.
[17:16]	**BK0SRC**	**Brake 0 Pin Source Select** For EPWM0 setting: 0 = Brake 0 pin source come from EPWM0_BRAKE0. 1 = Brake 0 pin source come from EPWM1_BRAKE0. 2 = Brake 0 pin source come from EPWM2_BRAKE0. For EPWM1 setting: 0 = Brake 0 pin source come from EPWM1_BRAKE0. 1 = Brake 0 pin source come from EPWM0_BRAKE0. 2 = Brake 1 pin source come from EPWM2_BRAKE0. For EPWM2 setting: 0 = Brake 0 pin source come from EPWM2_BRAKE0. 1 = Brake 0 pin source come from EPWM0_BRAKE0. 2 = Brake 1 pin source come from EPWM1_BRAKE0.

[15]	**BRK1PINV**	**Brake 1 Pin Inverse** 0 **=** brake pin event will be detected if EPWMx BRAKE1 pin status transfer from low to high in edge-detect, or pin status is high in level-detect. 1 = brake pin event will be detected if EPWMx BRAKE1 pin status transfer from high to low in edge-detect, or pin status is low in level-detect.
[14:12]	**BRK1FCNT**	**Brake 1 Edge Detector Filter Count** The register bits control the Brake1 filter counter to count from 0 to BRK1FCNT.
[11:9]	**BRK1NFSEL**	**Brake 1 Edge Detector Filter Clock Selection** 000 = Filter clock = HCLK. 001 = Filter clock = HCLK/2. 010 = Filter clock = HCLK/4. 011 = Filter clock = HCLK/8. 100 = Filter clock = HCLK/16. 101 = Filter clock = HCLK/32. 110 = Filter clock = HCLK/64. 111 = Filter clock = HCLK/128.
[8]	**BRK1NFEN**	**EPWM Brake 1 Noise Filter Enable Bit** 0 = Noise filter of EPWM Brake 1 Disabled. 1 = Noise filter of EPWM Brake 1 Enabled.
[7]	**BRK0PINV**	**Brake 0 Pin Inverse** 0 **=** brake pin event will be detected if EPWMx BRAKE0 pin status transfer from low to high in edge-detect, or pin status is high in level-detect. 1 = brake pin event will be detected if EPWMx BRAKE0 pin status transfer from high to low in edge-detect, or pin status is low in level-detect.
[6:4]	**BRK0FCNT**	**Brake 0 Edge Detector Filter Count** The register bits control the Brake0 filter counter to count from 0 to BRK0FCNT.
[3:1]	**BRK0NFSEL**	**Brake 0 Edge Detector Filter Clock Selection** 000 = Filter clock = HCLK. 001 = Filter clock = HCLK/2. 010 = Filter clock = HCLK/4. 011 = Filter clock = HCLK/8. 100 = Filter clock = HCLK/16. 101 = Filter clock = HCLK/32. 110 = Filter clock = HCLK/64. 111 = Filter clock = HCLK/128.
[0]	**BRK0NFEN**	**EPWM Brake 0 Noise Filter Enable Bit** 0 = Noise filter of EPWM Brake 0 Disabled. 1 = Noise filter of EPWM Brake 0 Enabled.

EPWM System Fail Brake Control Register (EPWM_FAILBRK)

Register	Offset	R/W	Description	Reset Value
EPWM_FAILBRK	EPWMx_BA+0xC4	R/W	EPWM System Fail Brake Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				CORBRKEN	Reserved	LVDBRKEN	CSSBRKEN

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3]	**CORBRKEN**	**Core Lockup Detection Trigger EPWM Brake Function 0 Enable Bit** 0 = Brake Function triggered by Core lockup detection Disabled. 1 = Brake Function triggered by Core lockup detection Enabled.
[2]	**Reserved**	Reserved.
[1]	**LVDBRKEN**	**Low Voltage Interrupt Flag Detection Trigger EPWM Brake Function 0 Enable Bit** 0 = Brake Function triggered by LVD Disabled. 1 = Brake Function triggered by LVD Enabled.
[0]	**CSSBRKEN**	**Clock Security System Detection Trigger EPWM Brake Function 0 Enable Bit** 0 = Brake Function triggered by CSS detection Disabled. 1 = Brake Function triggered by CSS detection Enabled.

EPWM Brake Edge Detect Control Register 0_1, 2_3, 4_5 (EPWM_BRKCTL0_1, 2_3, 4_5)

Register	Offset	R/W	Description	Reset Value
EPWM_BRKCTL0_1	EPWMx_BA+0xC8	R/W	EPWM Brake Edge Detect Control Register 0/1	0x0000_0000
EPWM_BRKCTL2_3	EPWMx_BA+0xCC	R/W	EPWM Brake Edge Detect Control Register 2/3	0x0000_0000
EPWM_BRKCTL4_5	EPWMx_BA+0xD0	R/W	EPWM Brake Edge Detect Control Register 4/5	0x0000_0000

31	30	29	28	27	26	25	24
Reserved			EADCLBEN	Reserved			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			EADCEBEN	BRKAODD		BRKAEVEN	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SYSLBEN	Reserved	BRKP1LEN	BRKP0LEN	Reserved			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SYSEBEN	Reserved	BRKP1EEN	BRKP0EEN	Reserved			

Bits	Description	
[31:29]	**Reserved**	Reserved.
[28]	**EADCLBEN**	**Enable EADC Result Monitor As Level-detect Brake Source (Write Protect)** 0 = EADCRM as level-detect brake source Disabled. 1 = EADCRM as level-detect brake source Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register
[27:21]	**Reserved**	Reserved.
[20]	**EADCEBEN**	**Enable EADC Result Monitor As Edge-detect Brake Source (Write Protect)** 0 = EADCRM as edge-detect brake source Disabled. 1 = EADCRM as edge-detect brake source Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register
[19:18]	**BRKAODD**	**EPWM Brake Action Select for Odd Channel (Write Protect)** 00 = EPWMx brake event will not affect odd channels output. 01 = EPWM odd channel output tri-state when EPWMx brake event happened. 10 = EPWM odd channel output low level when EPWMx brake event happened. 11 = EPWM odd channel output high level when EPWMx brake event happened. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register
[17:16]	**BRKAEVEN**	**EPWM Brake Action Select for Even Channel (Write Protect)** 00 = EPWMx brake event will not affect even channels output. 01 = EPWM even channel output tri-state when EPWMx brake event happened. 10 = EPWM even channel output low level when EPWMx brake event happened. 11 = EPWM even channel output high level when EPWMx brake event happened. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or

		SYS_RLKSUBM register.
[15]	**SYSLBEN**	**Enable System Fail As Level-detect Brake Source (Write Protect)** 0 = System Fail condition as level-detect brake source Disabled. 1 = System Fail condition as level-detect brake source Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register.
[14]	**Reserved**	Reserved.
[13]	**BRKP1LEN**	**Enable BKP1 Pin As Level-detect Brake Source (Write Protect)** 0 = EPWMx_BRAKE1 pin as level-detect brake source Disabled. 1 = EPWMx_BRAKE1 pin as level-detect brake source Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register.
[12]	**BRKP0LEN**	**Enable BKP0 Pin As Level-detect Brake Source (Write Protect)** 0 = EPWMx_BRAKE0 pin as level-detect brake source Disabled. 1 = EPWMx_BRAKE0 pin as level-detect brake source Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register.
[11:8]	**Reserved**	Reserved.
[7]	**SYSEBEN**	**Enable System Fail As Edge-detect Brake Source (Write Protect)** 0 = System Fail condition as edge-detect brake source Disabled. 1 = System Fail condition as edge-detect brake source Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register.
[6]	**Reserved**	Reserved.
[5]	**BRKP1EEN**	**Enable EPWMx_BRAKE1 Pin As Edge-detect Brake Source (Write Protect)** 0 = EPWMx_BRAKE1 pin as edge-detect brake source Disabled. 1 = EPWMx_BRAKE1 pin as edge-detect brake source Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register.
[4]	**BRKP0EEN**	**Enable EPWMx_BRAKE0 Pin As Edge-detect Brake Source (Write Protect)** 0 = EPWMx_BRAKE0 pin as edge-detect brake source Disabled. 1 = EPWMx_BRAKE0 pin as edge-detect brake source Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register.
[3:0]	**Reserved**	Reserved.

EPWM Pin Polar Inverse Control (EPWM_POLCTL)

Register	Offset	R/W	Description	Reset Value
EPWM_POLCTL	EPWMx_BA+0xD4	R/W	EPWM Pin Polar Inverse Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		PINV5	PINV4	PINV3	PINV2	PINV1	PINV0

Bits	Description	
[31:6]	**Reserved**	Reserved.
[n] n=0,1..5	**PINVn**	**EPWM PIN Polar Inverse Control** The register controls polarity state of EPWMx_CHn output pin. 0 = EPWMx_CHn output pin polar inverse Disabled. 1 = EPWMx_CHn output pin polar inverse Enabled.

EPWM Output Enable Register (EPWM_POEN)

Register	Offset	R/W	Description	Reset Value
EPWM_POEN	EPWMx_BA+0xD8	R/W	EPWM Output Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		POEN5	POEN4	POEN3	POEN2	POEN1	POEN0

Bits	Description	
[31:6]	**Reserved**	Reserved.
[n] n=0,1..5	**POENn**	**EPWM Pin Output Enable Bits** 0 = EPWMx_CHn pin at tri-state. 1 = EPWMx_CHn pin in output mode.

EPWM Software Brake Control Register (EPWM_SWBRK)

Register	Offset	R/W	Description	Reset Value
EPWM_SWBRK	EPWMx_BA+0xDC	W	EPWM Software Brake Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					BRKLTRG4	BRKLTRG2	BRKLTRG0
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					BRKETRG4	BRKETRG2	BRKETRG0

Bits	Description	
[31:11]	**Reserved**	Reserved.
[8+n/2] n=0,2,4	**BRKLTRGn**	**EPWM Level Brake Software Trigger (Write Only) (Write Protect)** Write 1 to this bit will trigger level brake, and set BRKLIFn to 1 in EPWM_INTSTS1 register. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register.
[7:3]	**Reserved**	Reserved.
[n/2] n=0,2,4	**BRKETRGn**	**EPWM Edge Brake Software Trigger (Write Only) (Write Protect)** Write 1 to this bit will trigger edge brake, and set BRKEIFn to 1 in EPWM_INTSTS1 register. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register.

EPWM Interrupt Enable Register 0 (EPWM_INTEN0)

Register	Offset	R/W	Description	Reset Value
EPWM_INTEN0	EPWMx_BA+0xE0	R/W	EPWM Interrupt Enable Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		CMPDIEN5	CMPDIEN4	CMPDIEN3	CMPDIEN2	CMPDIEN1	CMPDIEN0
23	22	21	20	19	18	17	16
Reserved		CMPUIEN5	CMPUIEN4	CMPUIEN3	CMPUIEN2	CMPUIEN1	CMPUIEN0
15	14	13	12	11	10	9	8
Reserved		PIEN5	PIEN4	PIEN3	PIEN2	PIEN1	PIEN0
7	6	5	4	3	2	1	0
Reserved		ZIEN5	ZIEN4	ZIEN3	ZIEN2	ZIEN1	ZIEN0

Bits	Description	
[31:30]	**Reserved**	Reserved.
[24+n] n=0,1..5	**CMPDIENn**	**EPWM Compare Down Count Interrupt Enable Bits** 0 = Compare down count interrupt Disabled. 1 = Compare down count interrupt Enabled. **Note:** In complementary mode, CMPDIEN1, 3, 5 is used as another CMPDIEN for channel 0, 2, 4.
[23:22]	**Reserved**	Reserved.
[16+n] n=0,1..5	**CMPUIENn**	**EPWM Compare Up Count Interrupt Enable Bits** 0 = Compare up count interrupt Disabled. 1 = Compare up count interrupt Enabled. **Note:** In complementary mode, CMPUIEN1, 3, 5 is used as another CMPUIEN for channel 0, 2, 4.
[15:14]	**Reserved**	Reserved.
[8+n] n=0,1..5	**PIENn**	**EPWM Period Point Interrupt Enable Bits** 0 = Period point interrupt Disabled. 1 = Period point interrupt Enabled. **Note 1:** When up-down counter type period point means center point. **Note 2:** Odd channels will read always 0 at complementary mode.
[7:6]	**Reserved**	Reserved.
[n] n=0,1..5	**ZIENn**	**EPWM Zero Point Interrupt Enable Bits** 0 = Zero point interrupt Disabled. 1 = Zero point interrupt Enabled. **Note:** Odd channels will read always 0 at complementary mode.

EPWM Interrupt Enable Register 1 (EPWM_INTEN1)

Register	Offset	R/W	Description	Reset Value
EPWM_INTEN1	EPWMx_BA+0xE4	R/W	EPWM Interrupt Enable Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					BRKLIEN4_5	BRKLIEN2_3	BRKLIEN0_1
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					BRKEIEN4_5	BRKEIEN2_3	BRKEIEN0_1

Bits	Description	
[31:11]	**Reserved**	Reserved.
[10]	**BRKLIEN4_5**	**EPWM Level-detect Brake Interrupt Enable for Channel4/5 (Write Protect)** 0 = Level-detect Brake interrupt for channel4/5 Disabled. 1 = Level-detect Brake interrupt for channel4/5 Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register.
[9]	**BRKLIEN2_3**	**EPWM Level-detect Brake Interrupt Enable for Channel2/3 (Write Protect)** 0 = Level-detect Brake interrupt for channel2/3 Disabled. 1 = Level-detect Brake interrupt for channel2/3 Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register.
[8]	**BRKLIEN0_1**	**EPWM Level-detect Brake Interrupt Enable for Channel0/1 (Write Protect)** 0 = Level-detect Brake interrupt for channel0/1 Disabled. 1 = Level-detect Brake interrupt for channel0/1 Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register.
[7:3]	**Reserved**	Reserved.
[2]	**BRKEIEN4_5**	**EPWM Edge-detect Brake Interrupt Enable for Channel4/5 (Write Protect)** 0 = Edge-detect Brake interrupt for channel4/5 Disabled. 1 = Edge-detect Brake interrupt for channel4/5 Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register.
[1]	**BRKEIEN2_3**	**EPWM Edge-detect Brake Interrupt Enable for Channel2/3 (Write Protect)** 0 = Edge-detect Brake interrupt for channel2/3 Disabled. 1 = Edge-detect Brake interrupt for channel2/3 Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register.
[0]	**BRKEIEN0_1**	**EPWM Edge-detect Brake Interrupt Enable for Channel0/1 (Write Protect)**

		0 = Edge-detect Brake interrupt for channel0/1 Disabled. 1 = Edge-detect Brake interrupt for channel0/1 Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register.

EPWM Interrupt Flag Register 0 (EPWM_INTSTS0)

Register	Offset	R/W	Description	Reset Value
EPWM_INTSTS0	EPWMx_BA+0xE8	R/W	EPWM Interrupt Flag Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		CMPDIF5	CMPDIF4	CMPDIF3	CMPDIF2	CMPDIF1	CMPDIF0
23	22	21	20	19	18	17	16
Reserved		CMPUIF5	CMPUIF4	CMPUIF3	CMPUIF2	CMPUIF1	CMPUIF0
15	14	13	12	11	10	9	8
Reserved		PIF5	PIF4	PIF3	PIF2	PIF1	PIF0
7	6	5	4	3	2	1	0
Reserved		ZIF5	ZIF4	ZIF3	ZIF2	ZIF1	ZIF0

Bits	Description	
[31:30]	**Reserved**	Reserved.
[24+n] n=0,1..5	**CMPDIFn**	**EPWM Compare Down Count Interrupt Flag** Flag is set by hardware when EPWM counter down count and reaches EPWM_CMPDATn; software can clear this bit by writing 1 to it. **Note:** In complementary mode, CMPDIF1, 3, 5 is used as another CMPDIF for channel 0, 2, 4.
[23:22]	**Reserved**	Reserved.
[16+n] n=0,1..5	**CMPUIFn**	**EPWM Compare Up Count Interrupt Flag** Flag is set by hardware when EPWM counter up count and reaches EPWM_CMPDATn; software can clear this bit by writing 1 to it. **Note:** In complementary mode, CMPUIF1, 3, 5 is used as another CMPUIF for channel 0, 2, 4.
[15:14]	**Reserved**	Reserved.
[8+n] n=0,1..5	**PIFn**	**EPWM Period Point Interrupt Flag** This bit is set by hardware when EPWM counter reaches EPWM_PERIODn. **Note:** This bit can be cleared to 0 by software writing 1.
[7:6]	**Reserved**	Reserved.
[n] n=0,1..5	**ZIFn**	**EPWM Zero Point Interrupt Flag** This bit is set by hardware when EPWM counter reaches 0. **Note:** This bit can be cleared to 0 by software writing 1

EPWM Interrupt Flag Register 1 (EPWM_INTSTS1)

Register	Offset	R/W	Description	Reset Value
EPWM_INTSTS1	EPWMx_BA+0xEC	R/W	EPWM Interrupt Flag Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		BRKLSTS5	BRKLSTS4	BRKLSTS3	BRKLSTS2	BRKLSTS1	BRKLSTS0
23	22	21	20	19	18	17	16
Reserved		BRKESTS5	BRKESTS4	BRKESTS3	BRKESTS2	BRKESTS1	BRKESTS0
15	14	13	12	11	10	9	8
Reserved		BRKLIF5	BRKLIF4	BRKLIF3	BRKLIF2	BRKLIF1	BRKLIF0
7	6	5	4	3	2	1	0
Reserved		BRKEIF5	BRKEIF4	BRKEIF3	BRKEIF2	BRKEIF1	BRKEIF0

Bits	Description	
[31:30]	**Reserved**	Reserved.
[24+n] n=0,1..5	**BRKLSTSn**	**EPWM Channel n Level-detect Brake Status (Read Only)** 0 = EPWM channel n level-detect brake state is released. 1 = When EPWM channel n level-detect brake detects a falling edge of any enabled brake source; this flag will be set to indicate the EPWM channel n at brake state. **Note:** This bit is read only and auto cleared by hardware. When enabled brake source return to high level, EPWM will release brake state until current EPWM period finished. The EPWM waveform will start output from next full EPWM period.
[23:22]	**Reserved**	Reserved.
[16+n] n=0,1..5	**BRKESTSn**	**EPWM Channel n Edge-detect Brake Status (Read Only)** 0 = EPWM channel n edge-detect brake state is released. 1 = When EPWM channel n edge-detect brake detects a falling edge of any enabled brake source; this flag will be set to indicate the EPWM channel n at brake state. **Note:** This bit is read only and auto cleared by hardware. When edge-detect brake interrupt flag is cleared, EPWM will release brake state until current EPWM period finished. The EPWM waveform will start output from next full EPWM period.
[15:14]	**Reserved**	Reserved.
[8+n] n=0,1..5	**BRKLIFn**	**EPWM Channel n Level-detect Brake Interrupt Flag (Write Protect)** 0 = EPWM channel n level-detect brake event did not happen. 1 = When EPWM channel n level-detect brake event happened, this bit is set to 1; writing 1 to clear. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register.
[7:6]	**Reserved**	Reserved.
[n] n=0,1..5	**BRKEIFn**	**EPWM Channel n Edge-detect Brake Interrupt Flag (Write Protect)** 0 = EPWM channel n edge-detect brake event did not happen. 1 = When EPWM channel n edge-detect brake event happened, this bit is set to 1; writing 1 to clear. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register.

EPWM Trigger EADC Source Select Register 0 (EPWM_EADCTS0)

Register	Offset	R/W	Description	Reset Value
EPWM_EADCTS0	EPWMx_BA+0xF8	R/W	EPWM Trigger EADC Source Select Register 0	0x0000_0000

31	30	29	28	27	26	25	24
TRGEN3	**Reserved**			**TRGSEL3**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TRGEN2	**Reserved**			**TRGSEL2**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TRGEN1	**Reserved**			**TRGSEL1**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TRGEN0	**Reserved**			**TRGSEL0**			

Bits	Description	
[31]	**TRGEN3**	**EPWM_CH3 Trigger EADC Enable Bit** 0 = EPWM_CH3 Trigger EADC function Disabled. 1 = EPWM_CH3 Trigger EADC function Enabled.
[30:28]	**Reserved**	Reserved.
[27:24]	**TRGSEL3**	**EPWM_CH3 Trigger EADC Source Select** 0000 = EPWM_CH2 zero point. 0001 = EPWM_CH2 period point. 0010 = EPWM_CH2 zero or period point. 0011 = EPWM_CH2 up-count compared point. 0100 = EPWM_CH2 down-count compared point. 0101 = EPWM_CH3 zero point. 0110 = EPWM_CH3 period point. 0111 = EPWM_CH3 zero or period point. 1000 = EPWM_CH3 up-count compared point. 1001 = EPWM_CH3 down-count compared point. 1010 = EPWM_CH0 up-count free trigger compared point. 1011 = EPWM_CH0 down-count free trigger compared point. 1100 = EPWM_CH2 up-count free trigger compared point. 1101 = EPWM_CH2 down-count free trigger compared point. 1110 = EPWM_CH4 up-count free trigger compared point. 1111 = EPWM_CH4 down-count free trigger compared point.
[23]	**TRGEN2**	**EPWM_CH2 Trigger EADC Enable Bit** 0 = EPWM_CH2 Trigger EADC function Disabled. 1 = EPWM_CH2 Trigger EADC function Enabled.
[22:20]	**Reserved**	Reserved.
[19:16]	**TRGSEL2**	**EPWM_CH2 Trigger EADC Source Select** 0000 = EPWM_CH2 zero point.

		0001 = EPWM_CH2 period point. 0010 = EPWM_CH2 zero or period point. 0011 = EPWM_CH2 up-count compared point. 0100 = EPWM_CH2 down-count compared point. 0101 = EPWM_CH3 zero point. 0110 = EPWM_CH3 period point. 0111 = EPWM_CH3 zero or period point. 1000 = EPWM_CH3 up-count compared point. 1001 = EPWM_CH3 down-count compared point. 1010 = EPWM_CH0 up-count free trigger compared point. 1011 = EPWM_CH0 down-count free trigger compared point. 1100 = EPWM_CH2 up-count free trigger compared point. 1101 = EPWM_CH2 down-count free trigger compared point. 1110 = EPWM_CH4 up-count free trigger compared point. 1111 = EPWM_CH4 down-count free trigger compared point.
[15]	**TRGEN1**	**EPWM_CH1 Trigger EADC Enable Bit** 0 = EPWM_CH1 Trigger EADC function Disabled. 1 = EPWM_CH1 Trigger EADC function Enabled.
[14:12]	**Reserved**	Reserved.
[11:8]	**TRGSEL1**	**EPWM_CH1 Trigger EADC Source Select** 0000 = EPWM_CH0 zero point. 0001 = EPWM_CH0 period point. 0010 = EPWM_CH0 zero or period point. 0011 = EPWM_CH0 up-count compared point. 0100 = EPWM_CH0 down-count compared point. 0101 = EPWM_CH1 zero point. 0110 = EPWM_CH1 period point. 0111 = EPWM_CH1 zero or period point. 1000 = EPWM_CH1 up-count compared point. 1001 = EPWM_CH1 down-count compared point. 1010 = EPWM_CH0 up-count free trigger compared point. 1011 = EPWM_CH0 down-count free trigger compared point. 1100 = EPWM_CH2 up-count free trigger compared point. 1101 = EPWM_CH2 down-count free trigger compared point. 1110 = EPWM_CH4 up-count free trigger compared point. 1111 = EPWM_CH4 down-count free trigger compared point.
[7]	**TRGEN0**	**EPWM_CH0 Trigger EADC Enable Bit** 0 = EPWM_CH0 Trigger EADC function Disabled. 1 = EPWM_CH0 Trigger EADC function Enabled.
[6:4]	**Reserved**	Reserved.
[3:0]	**TRGSEL0**	**EPWM_CH0 Trigger EADC Source Select** 0000 = EPWM_CH0 zero point. 0001 = EPWM_CH0 period point. 0010 = EPWM_CH0 zero or period point. 0011 = EPWM_CH0 up-count compared point. 0100 = EPWM_CH0 down-count compared point. 0101 = EPWM_CH1 zero point.

		0110 = EPWM_CH1 period point. 0111 = EPWM_CH1 zero or period point. 1000 = EPWM_CH1 up-count compared point. 1001 = EPWM_CH1 down-count compared point. 1010 = EPWM_CH0 up-count free trigger compared point. 1011 = EPWM_CH0 down-count free trigger compared point. 1100 = EPWM_CH2 up-count free trigger compared point. 1101 = EPWM_CH2 down-count free trigger compared point. 1110 = EPWM_CH4 up-count free trigger compared point. 1111 = EPWM_CH4 down-count free trigger compared point.

EPWM Trigger EADC Source Select Register 1 (EPWM_EADCTS1)

Register	Offset	R/W	Description	Reset Value
EPWM_EADCTS1	EPWMx_BA+0xFC	R/W	EPWM Trigger EADC Source Select Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TRGEN5	Reserved			TRGSEL5			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TRGEN4	Reserved			TRGSEL4			

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15]	**TRGEN5**	**EPWM_CH5 Trigger EADC Enable Bit** 0 = EPWM_CH5 Trigger EADC function Disabled. 1 = EPWM_CH5 Trigger EADC function Enabled.
[14:12]	**Reserved**	Reserved.
[11:8]	**TRGSEL5**	**EPWM_CH5 Trigger EADC Source Select** 0000 = EPWM_CH4 zero point. 0001 = EPWM_CH4 period point. 0010 = EPWM_CH4 zero or period point. 0011 = EPWM_CH4 up-count compared point. 0100 = EPWM_CH4 down-count compared point. 0101 = EPWM_CH5 zero point. 0110 = EPWM_CH5 period point. 0111 = EPWM_CH5 zero or period point. 1000 = EPWM_CH5 up-count compared point. 1001 = EPWM_CH5 down-count compared point. 1010 = EPWM_CH0 up-count free trigger compared point. 1011 = EPWM_CH0 down-count free trigger compared point. 1100 = EPWM_CH2 up-count free trigger compared point. 1101 = EPWM_CH2 down-count free trigger compared point. 1110 = EPWM_CH4 up-count free trigger compared point. 1111 = EPWM_CH4 down-count free trigger compared point.
[7]	**TRGEN4**	**EPWM_CH4 Trigger EADC Enable Bit** 0 = EPWM_CH4 Trigger EADC function Disabled. 1 = EPWM_CH4 Trigger EADC function Enabled.
[6:4]	**Reserved**	Reserved.

[3:0]	**TRGSEL4**	**EPWM_CH4 Trigger EADC Source Select** 0000 = EPWM_CH4 zero point. 0001 = EPWM_CH4 period point. 0010 = EPWM_CH4 zero or period point. 0011 = EPWM_CH4 up-count compared point. 0100 = EPWM_CH4 down-count compared point. 0101 = EPWM_CH5 zero point. 0110 = EPWM_CH5 period point. 0111 = EPWM_CH5 zero or period point. 1000 = EPWM_CH5 up-count compared point. 1001 = EPWM_CH5 down-count compared point. 1010 = EPWM_CH0 up-count free trigger compared point. 1011 = EPWM_CH0 down-count free trigger compared point. 1100 = EPWM_CH2 up-count free trigger compared point. 1101 = EPWM_CH2 down-count free trigger compared point. 1110 = EPWM_CH4 up-count free trigger compared point. 1111 = EPWM_CH4 down-count free trigger compared point.

EPWM Free Trigger Compare Register 0_1, 2_3, 4_5 (EPWM_FTCMPDAT0_1, 2_3, 4_5)

Register	Offset	R/W	Description	Reset Value
EPWM_FTCMPDAT0_1	EPWMx_BA+0x100	R/W	EPWM Free Trigger Compare Register 0/1	0x0000_0000
EPWM_FTCMPDAT2_3	EPWMx_BA+0x104	R/W	EPWM Free Trigger Compare Register 2/3	0x0000_0000
EPWM_FTCMPDAT4_5	EPWMx_BA+0x108	R/W	EPWM Free Trigger Compare Register 4/5	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
FTCMP							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FTCMP							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**FTCMP**	**EPWM Free Trigger Compare Register** FTCMP uses to compare with even CNT (EPWM_CNTm[15:0], m=0,2,4) to trigger EADC. FTCMPDAT0, 2, 4 corresponding complementary pairs EPWM_CH0and EPWM_CH1, EPWM_CH2 and EPWM_CH3, EPWM_CH4 and EPWM_CH5.

EPWM Synchronous Start Control Register (EPWM_SSCTL)

Register	Offset	R/W	Description	Reset Value
EPWM_SSCTL	EPWMx_BA+0x110	R/W	EPWM Synchronous Start Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						SSRC	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		SSEN5	SSEN4	SSEN3	SSEN2	SSEN1	SSEN0

Bits	Description	
[31:10]	**Reserved**	Reserved.
[9:8]	**SSRC**	**EPWM Synchronous Start Source Select Bits** 00 = Synchronous start source come from EPWM0. 01 = Synchronous start source come from EPWM1. 10 = Synchronous start source come from EPWM2. 11 = Reserved.
[7:6]	**Reserved**	Reserved.
[n] n=0,1..5	**SSENn**	**EPWM Synchronous Start Function Enable Bits** When synchronous start function is enabled, the EPWM counter enable register (EPWM_CNTEN) can be enabled by writing EPWM synchronous start trigger bit (CNTSEN). 0 = EPWM synchronous start function Disabled. 1 = EPWM synchronous start function Enabled.

EPWM Synchronous Start Trigger Register (EPWM_SSTRG)

Register	Offset	R/W	Description	Reset Value
EPWM_SSTRG	EPWMx_BA+0x114	W	EPWM Synchronous Start Trigger Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							CNTSEN

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**CNTSEN**	**EPWM Counter Synchronous Start Enable (Write Only)** PMW counter synchronous enable function is used to make selected EPWM channels (include EPWM0_CHx and EPWM1_CHx) start counting at the same time. Writing this bit to 1 will also set the counter enable bit (CNTENn, n denotes channel 0 to 5) if correlated EPWM channel counter synchronous start function is enabled.

EPWM Status Register (EPWM_STATUS)

Register	Offset	R/W	Description	Reset Value
EPWM_STATUS	EPWMx_BA+0x120	R/W	EPWM Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		EADCTRGF5	EADCTRGF4	EADCTRGF3	EADCTRGF2	EADCTRGF1	EADCTRGF0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					SYNCINF4	SYNCINF2	SYNCINF0
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		CNTMAXF5	CNTMAXF4	CNTMAXF3	CNTMAXF2	CNTMAXF1	CNTMAXF0

Bits	Description	
[31:22]	**Reserved**	Reserved.
[16+n] n=0,1..5	**EADCTRGFn**	**EADC Start of Conversion Flag** 0 = No EADC start of conversion trigger event occurred. 1 = An EADC start of conversion trigger event occurred. **Note:** This bit can be cleared by software writing 1.
[15:11]	**Reserved**	Reserved.
[8+n/2] n=0,2,4	**SYNCINFn**	**Input Synchronization Latched Flag** 0 = No SYNC_IN event occurred. 1 = A SYNC_IN event occurred. **Note:** This bit can be cleared by software writing 1.
[7:6]	**Reserved**	Reserved.
[n] n=0,1..5	**CNTMAXFn**	**Time-base Counter Equal to 0xFFFF Latched Flag** 0 = The time-base counter never reached its maximum value 0xFFFF. 1 = The time-base counter reached its maximum value. **Note:** This bit can be cleared by software writing 1.

EPWM Interrupt Flag Accumulator Register (EPWM_IFAn)

Register	Offset	R/W	Description	Reset Value
EPWM_IFA0	EPWMx_BA+0x130	R/W	EPWM Interrupt Flag Accumulator Register 0	0x0000_0000
EPWM_IFA1	EPWMx_BA+0x134	R/W	EPWM Interrupt Flag Accumulator Register 1	0x0000_0000
EPWM_IFA2	EPWMx_BA+0x138	R/W	EPWM Interrupt Flag Accumulator Register 2	0x0000_0000
EPWM_IFA3	EPWMx_BA+0x13C	R/W	EPWM Interrupt Flag Accumulator Register 3	0x0000_0000
EPWM_IFA4	EPWMx_BA+0x140	R/W	EPWM Interrupt Flag Accumulator Register 4	0x0000_0000
EPWM_IFA5	EPWMx_BA+0x144	R/W	EPWM Interrupt Flag Accumulator Register 5	0x0000_0000

31	30	29	28	27	26	25	24
IFAEN	Reserved	IFASEL		Reserved			STPMOD
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
IFACNT							
7	6	5	4	3	2	1	0
IFACNT							

Bits	Description	
[31]	**IFAEN**	**EPWM_CHn Interrupt Flag Accumulator Enable Bits** 0 = EPWM_CHn interrupt flag accumulator Disabled. 1 = EPWM_CHn interrupt flag accumulator Enabled.
[30]	**Reserved**	Reserved.
[29:28]	**IFASEL**	**EPWM_CHn Interrupt Flag Accumulator Source Select** 00 = EPWM_CHn zero point. 01 = EPWM_CHn period in channel n. 10 = EPWM_CHn up-count compared point. 11 = EPWM_CHn down-count compared point.
[27:25]	**Reserved**	Reserved.
[24]	**STPMOD**	**EPWM_CHn Accumulator Stop Mode Enable Bits** 0 = EPWM_CHn Stop Mode Disable. 1 = EPWM_CHn Stop Mode Enable.
[23:16]	**Reserved**	Reserved.
[15:0]	**IFACNT**	**EPWM_CHn Interrupt Flag Counter** The register sets the count number which defines how many times of EPWM_CHn period occurs to set bit IFAIFn to request the EPWM period interrupt. EPWM flag will be set in every IFACNT[15:0] times of EPWM period.

EPWM Accumulator Interrupt Flag Register (EPWM_AINTSTS)

Register	Offset	R/W	Description	Reset Value
EPWM_AINTSTS	EPWMx_BA+0x150	R/W	EPWM Accumulator Interrupt Flag Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**IFAIF5**	**IFAIF4**	**IFAIF3**	**IFAIF2**	**IFAIF1**	**IFAIF0**

Bits	Description	
[31:6]	**Reserved**	Reserved.
[n] n=0,1..5	**IFAIFn**	**EPWM_CHn Interrupt Flag Accumulator Interrupt Flag** Flag is set by hardware when condition matches IFASEL in EPWM_IFAn register; software can clear this bit by writing 1 to it.

EPWM Accumulator Interrupt Enable Register (EPWM_AINTEN)

Register	Offset	R/W	Description	Reset Value
EPWM_AINTEN	EPWMx_BA+0x154	R/W	EPWM Accumulator Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		IFAIEN5	IFAIEN4	IFAIEN3	IFAIEN2	IFAIEN1	IFAIEN0

Bits	Description	
[31:6]	**Reserved**	Reserved.
[n] n=0,1..5	**IFAIENn**	**EPWM_CHn Interrupt Flag Accumulator Interrupt Enable Bits** 0 = Interrupt Flag accumulator interrupt Disabled. 1 = Interrupt Flag accumulator interrupt Enabled.

EPWM Accumulator PDMA Control Register (EPWM_APDMACTL)

Register	Offset	R/W	Description	Reset Value
EPWM_APDMACTL	EPWMx_BA+0x158	R/W	EPWM Accumulator PDMA Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		APDMAEN5	APDMAEN4	APDMAEN3	APDMAEN2	APDMAEN1	APDMAEN0

Bits	Description	
[31:6]	**Reserved**	Reserved.
[n] n=0,1..5	**APDMAENn**	**Channel n Accumulator PDMA Enable Bits** 0 = Channel n PDMA function Disabled. 1 = Channel n PDMA function Enabled for the channel n to trigger PDMA to transfer memory data to register.

EPWM Fault Detect Enable Register (EPWM_FDEN)

Register	Offset	R/W	Description	Reset Value
EPWM_FDEN	EPWMx_BA+0x160	R/W	EPWM Fault Detect Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved		FDCKS5	FDCKS4	FDCKS3	FDCKS2	FDCKS1	FDCKS0
15	14	13	12	11	10	9	8
Reserved		FDODIS5	FDODIS4	FDODIS3	FDODIS2	FDODIS1	FDODIS0
7	6	5	4	3	2	1	0
Reserved		FDEN5	FDEN4	FDEN3	FDEN2	FDEN1	FDEN0

Bits	Description	
[31:22]	**Reserved**	Reserved.
[16+n] n=0,1..5	**FDCKSn**	**EPWM Channel n Fault Detect Clock Source Select Bit** 0 = EPWMx_CLK, x denotes 0, 1or 2. 1 = EPWMx_CLK divide by prescaler, x denotes 0, 1 or 2.
[8+n] n=0,1..5	**FDODISn**	**EPWM Channel n Output Fault Detect Disable Bit** 0 = EPWM detect fault and output Enable. 1 = EPWM detect fault and output Disable.
[n] n=0,1..5	**FDENn**	**EPWM Fault Detect Function Enable Bit** 0 = Fault detect function Disable. 1 = Fault detect function Enable.

EPWM Fault Detect Control Register 0~5 (EPWM_FDCTL0~5)

Register	Offset	R/W	Description	Reset Value
EPWM_FDCTL0	EPWMx_BA+0x164	R/W	EPWM Fault Detect Control Register 0	0x0000_0000
EPWM_FDCTL1	EPWMx_BA+0x168	R/W	EPWM Fault Detect Control Register 1	0x0000_0000
EPWM_FDCTL2	EPWMx_BA+0x16C	R/W	EPWM Fault Detect Control Register 2	0x0000_0000
EPWM_FDCTL3	EPWMx_BA+0x170	R/W	EPWM Fault Detect Control Register 3	0x0000_0000
EPWM_FDCTL4	EPWMx_BA+0x174	R/W	EPWM Fault Detect Control Register 4	0x0000_0000
EPWM_FDCTL5	EPWMx_BA+0x178	R/W	EPWM Fault Detect Control Register 5	0x0000_0000

31	30	29	28	27	26	25	24
FDDGEN	Reserved	FDCKSEL		Reserved			
23	22	21	20	19	18	17	16
Reserved					DGSMPCYC		
15	14	13	12	11	10	9	8
FDMSKEN	Reserved						
7	6	5	4	3	2	1	0
Reserved	TRMSKCNT						

Bits	Description	
[31]	**FDDGEN**	**Fault Detect Deglitch Enable Bit** 0 = Fault detect deglitch function Disable. 1 = Fault detect deglitch function Enable.
[30]	**Reserved**	Reserved.
[29:28]	**FDCKSEL**	**EPWM Channel Fault Detect Clock Select** 00 = FLT_CLK/1. 01 = FLT_CLK/2. 10 = FLT_CLK/4. 11 = FLT_CLK/8. **Note:** FLT_CLK is FDCKSn (EPWM_FDENn[16+n], n=0,1..5) selected clock.
[27:19]	**Reserved**	Reserved.
[18:16]	**DGSMPCYC**	**Deglitch Sampling Cycle** 1. FDCKS is set to 0: Sampling detect signal each EPWMx_CLK * (2^FDCKSEL) period and detect DGSMPCYC+1 times 2. FDCKS is set to 1: Sampling detect signal each EPWMx_CLK * CLKPSC * (2^FDCKSEL) period and detect DGSMPCYC+1 times **Note:** CLKPSC (EPWM_CLKPSC) is 0:

		TRMSKCNT >= DGSMPCYC + 2. FDCKS is 1 and CLKPSC (EPWM_CLKPSC) is 1: TRMSKCNT >= DGSMPCYC + 1. FDCKS is 1 and CLKPSC (EPWM_CLKPSC) is 2: TRMSKCNT >= DGSMPCYC.
[15]	**FDMSKEN**	**Fault Detect Mask Enable Bit** 0 = Fault detect mask function Disabled. 1 = Fault detect mask function Enabled.
[14:7]	**Reserved**	Reserved.
[6:0]	**TRMSKCNT**	**Transition Mask Counter** The fault detect result will be masked before counter count from 0 to TRMSKCNT. ● FDCKS is set to 0: Mask time is EPWMx_CLK * (2^FDCKSEL) * (TRMSKCNT +2) ● FDCKS is set to 1: Mask time EPWMx_CLK * CLKPSC * (2^FDCKSEL) * (TRMSKCNT +2) **Note:** CLKPSC (EPWM_CLKPSC) is 0: TRMSKCNT >= DGSMPCYC + 2. FDCKS is 1 and CLKPSC (EPWM_CLKPSC) is 1: TRMSKCNT >= DGSMPCYC + 1. FDCKS is 1 and CLKPSC (EPWM_CLKPSC) is 2: TRMSKCNT >= DGSMPCYC.

EPWM Fault Detect Interrupt Enable Register (EPWM_FDIEN)

Register	Offset	R/W	Description	Reset Value
EPWM_FDIEN	EPWMx_BA+0x17C	R/W	EPWM Fault Detect Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		FDIEN5	FDIEN4	FDIEN3	FDIEN2	FDIEN1	FDIEN0

Bits	Description	
[31:6]	**Reserved**	Reserved.
[n]	**FDIENn**	**EPWM Channel n Fault Detect Interrupt Enable Bit** 0 = EPWM Channel n Fault Detect Interrupt Disabled. 1 = EPWM Channel n Fault Detect Interrupt Enabled.

EPWM Fault Detect Interrupt Flag Register (EPWM_FDSTS)

Register	Offset	R/W	Description	Reset Value
EPWM_FDSTS	EPWMx_BA+0x180	R/W	EPWM Fault Detect Interrupt Flag Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		FDIF5	FDIF4	FDIF3	FDIF2	FDIF1	FDIF0

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5:0]	**FDIFn**	**EPWM Channel n Fault Detect Interrupt Flag Bit** Fault Detect Interrupt Flag will be set when EPWM output short. Software can clear this bit by writing 1 to it.

EPWM Trigger EADC Prescale Control Register (EPWM_EADCPSCCTL)

Register	Offset	R/W	Description	Reset Value
EPWM_EADCPSCCTL	EPWMx_BA+0x184	R/W	EPWM Trigger EADC Prescale Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		PSCEN5	PSCEN4	PSCEN3	PSCEN2	PSCEN1	PSCEN0

Bits	Description	
[31:6]	**Reserved**	Reserved.
[n] n=0,1..5	**PSCENn**	**EPWM Trigger EADC Pre-scale Function Enable Bits** 0 = EPWM Trigger EADC Pre-scale Function Disabled. 1 = EPWM Trigger EADC Pre-scale Function Enabled.

EPWM Trigger EADC Prescale Register 0 (EPWM_EADCPSC0)

Register	Offset	R/W	Description	Reset Value
EPWM_EADCPSC0	EPWMx_BA+0x188	R/W	EPWM Trigger EADC Prescale Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				EADCPSC3			
23	22	21	20	19	18	17	16
Reserved				EADCPSC2			
15	14	13	12	11	10	9	8
Reserved				EADCPSC1			
7	6	5	4	3	2	1	0
Reserved				EADCPSC0			

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:24]	**EADCPSC3**	**EPWM Channel 3 Trigger EADC Prescale** The register sets the count number which defines (EADCPSC3+1) times of EPWM_CH3 trigger EADC event occurs to trigger EADC and set trigger EADC flag bit EADCTRGF3.
[23:20]	**Reserved**	Reserved.
[19:16]	**EADCPSC2**	**EPWM Channel 2 Trigger EADC Prescale** The register sets the count number which defines (EADCPSC2+1) times of EPWM_CH2 trigger EADC event occurs to trigger EADC and set trigger EADC flag bit EADCTRGF3.
[15:12]	**Reserved**	Reserved.
[11:8]	**EADCPSC1**	**EPWM Channel 1 Trigger EADC Prescale** The register sets the count number which defines (EADCPSC1+1) times of EPWM_CH1 trigger EADC event occurs to trigger EADC and set trigger EADC flag bit EADCTRGF1.
[7:4]	**Reserved**	Reserved.
[3:0]	**EADCPSC0**	**EPWM Channel 0 Trigger EADC Prescale** The register sets the count number which defines (EADCPSC0+1) times of EPWM_CH0 trigger EADC event occurs to trigger EADC and set trigger EADC flag bit EADCTRGF0.

EPWM Trigger EADC Prescale Register 1 (EPWM_EADCPSC1)

Register	Offset	R/W	Description	Reset Value
EPWM_EADCPSC1	EPWMx_BA+0x18C	R/W	EPWM Trigger EADC Prescale Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				EADCPSC5			
7	6	5	4	3	2	1	0
Reserved				EADCPSC4			

Bits	Description	
[31:12]	**Reserved**	Reserved.
[11:8]	**EADCPSC5**	**EPWM Channel 5 Trigger EADC Prescale** The register sets the count number which defines (EADCPSC5+1) times of EPWM_CH3 trigger EADC event occurs to trigger EADC and set trigger EADC flag bit EADCTRGF5.
[7:4]	**Reserved**	Reserved.
[3:0]	**EADCPSC4**	**EPWM Channel 4 Trigger EADC Prescale** The register sets the count number which defines (EADCPSC4+1) times of EPWM_CH3 trigger EADC event occurs to trigger EADC and set trigger EADC flag bit EADCTRGF4.

EPWM Trigger EADC Prescale Counter Register 0 (EPWM_EADCPSCNT0)

Register	Offset	R/W	Description	Reset Value
EPWM_EADCPSCNT0	EPWMx_BA+0x190	R/W	EPWM Trigger EADC Prescale Counter Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				PSCNT3			
23	22	21	20	19	18	17	16
Reserved				PSCNT2			
15	14	13	12	11	10	9	8
Reserved				PSCNT1			
7	6	5	4	3	2	1	0
Reserved				PSCNT0			

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:24]	**PSCNT3**	**EPWM Trigger EADC Prescale Counter 3** User can monitor PSCNT3 to know the current value in 4-bit trigger EADC prescale counter. **Note 1:** user can write only when PSCEN3 is 0. **Note 2:** Write data limitation: PSCNT3 < EADCPSC3.
[23:20]	**Reserved**	Reserved.
[19:16]	**PSCNT2**	**EPWM Trigger EADC Prescale Counter 2** User can monitor PSCNT2 to know the current value in 4-bit trigger EADC prescale counter. **Note 1:** user can write only when PSCEN2 is 0. **Note 2:** Write data limitation: PSCNT2 < EADCPSC2.
[15:12]	**Reserved**	Reserved.
[11:8]	**PSCNT1**	**EPWM Trigger EADC Prescale Counter 1** User can monitor PSCNT1 to know the current value in 4-bit trigger EADC prescale counter. **Note 1:** user can write only when PSCEN1 is 0. **Note 2:** Write data limitation: PSCNT1 < EADCPSC1.
[7:4]	**Reserved**	Reserved.
[3:0]	**PSCNT0**	**EPWM Trigger EADC Prescale Counter 0** User can monitor PSCNT0 to know the current value in 4-bit trigger EADC prescale counter. **Note 1:** user can write only when PSCEN0 is 0. **Note 2:** Write data limitation: PSCNT0 < EADCPSC0.

EPWM Trigger EADC Prescale Counter Register 1 (EPWM_EADCPSCNT1)

Register	Offset	R/W	Description	Reset Value
EPWM_EADCPSCNT1	EPWMx_BA+0x194	R/W	EPWM Trigger EADC Prescale Counter Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				PSCNT5			
7	6	5	4	3	2	1	0
Reserved				PSCNT4			

Bits	Description	
[31:12]	**Reserved**	Reserved.
[11:8]	**PSCNT5**	**EPWM Trigger EADC Prescale Counter 5** User can monitor PSCNT5 to know the current value in 4-bit trigger EADC prescale counter. **Note 1:** user can write only when PSCEN5 is 0. **Note 2:** Write data limitation: PSCNT5 < EADCPSC5.
[7:4]	**Reserved**	Reserved.
[3:0]	**PSCNT4**	**EPWM Trigger EADC Prescale Counter 4** User can monitor PSCNT4 to know the current value in 4-bit trigger EADC prescale counter. **Note 1:** user can write only when PSCEN4 is 0. **Note 2:** Write data limitation: PSCNT4 < EADCPSC4.

EPWM Capture Input Enable Register (EPWM_CAPINEN)

Register	Offset	R/W	Description	Reset Value
EPWM_CAPINEN	EPWMx_BA+0x200	R/W	EPWM Capture Input Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	CAPINEN5		CAPINEN4	CAPINEN3	CAPINEN2	CAPINEN1	CAPINEN0

Bits	Description	
[31:6]	**Reserved**	Reserved.
[n] n=0,1..5	**CAPINENn**	**Capture Input Enable Bits** 0 = EPWM Channel capture input path Disabled. The input of EPWM channel capture function is always regarded as 0. 1 = EPWM Channel capture input path Enabled. The input of EPWM channel capture function comes from correlative multifunction pin.

EPWM Capture Control Register (EPWM_CAPCTL)

Register	Offset	R/W	Description	Reset Value
EPWM_CAPCTL	EPWMx_BA+0x204	R/W	EPWM Capture Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		FCRLDEN5	FCRLDEN4	FCRLDEN3	FCRLDEN2	FCRLDEN1	FCRLDEN0
23	22	21	20	19	18	17	16
Reserved		RCRLDEN5	RCRLDEN4	RCRLDEN3	RCRLDEN2	RCRLDEN1	RCRLDEN0
15	14	13	12	11	10	9	8
Reserved		CAPINV5	CAPINV4	CAPINV3	CAPINV2	CAPINV1	CAPINV0
7	6	5	4	3	2	1	0
Reserved	CAPEN5		CAPEN4	CAPEN3	CAPEN2	CAPEN1	CAPEN0

Bits	Description	
[31:30]	**Reserved**	Reserved.
[24+n] n=0,1..5	**FCRLDENn**	**Falling Capture Reload Enable Bits** 0 = Falling capture reload counter Disabled. 1 = Falling capture reload counter Enabled.
[23:22]	**Reserved**	Reserved.
[16+n] n=0,1..5	**RCRLDENn**	**Rising Capture Reload Enable Bits** 0 = Rising capture reload counter Disabled. 1 = Rising capture reload counter Enabled.
[15:14]	**Reserved**	Reserved.
[8+n] n=0,1..5	**CAPINVn**	**Capture Inverter Enable Bits** 0 = Capture source inverter Disabled. 1 = Capture source inverter Enabled. Reverse the input signal from GPIO.
[7:6]	**Reserved**	Reserved.
[n] n=0,1..5	**CAPENn**	**Capture Function Enable Bits** 0 = Capture function Disabled. RCAPDAT/FCAPDAT register will not be updated. 1 = Capture function Enabled. Capture latched the EPWM counter value when detected rising or falling edge of input signal and saved to RCAPDAT (Rising latch) and FCAPDAT (Falling latch).

EPWM Capture Status Register (EPWM_CAPSTS)

Register	Offset	R/W	Description	Reset Value
EPWM_CAPSTS	EPWMx_BA+0x208	R	EPWM Capture Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		CFLIFOV5	CFLIFOV4	CFLIFOV3	CFLIFOV2	CFLIFOV1	CFLIFOV0
7	6	5	4	3	2	1	0
Reserved		CRLIFOV5	CRLIFOV4	CRLIFOV3	CRLIFOV2	CRLIFOV1	CRLIFOV0

Bits	Description	
[31:14]	**Reserved**	Reserved.
[8+n] n=0,1..5	**CFLIFOVn**	**Capture Falling Latch Interrupt Flag Overrun Status (Read Only)** This flag indicates if falling latch happened when the corresponding CFLIF is 1. **Note:** This bit will be cleared automatically when user clear corresponding CFLIF.
[7:6]	**Reserved**	Reserved.
[n] n=0,1..5	**CRLIFOVn**	**Capture Rising Latch Interrupt Flag Overrun Status (Read Only)** This flag indicates if rising latch happened when the corresponding CRLIF is 1. **Note:** This bit will be cleared automatically when user clear corresponding CRLIF.

EPWM Rising Capture Data Register 0~5 (EPWM_RCAPDAT 0~5)

Register	Offset	R/W	Description	Reset Value
EPWM_RCAPDAT0	EPWMx_BA+0x20C	R	EPWM Rising Capture Data Register 0	0x0000_0000
EPWM_RCAPDAT1	EPWMx_BA+0x214	R	EPWM Rising Capture Data Register 1	0x0000_0000
EPWM_RCAPDAT2	EPWMx_BA+0x21C	R	EPWM Rising Capture Data Register 2	0x0000_0000
EPWM_RCAPDAT3	EPWMx_BA+0x224	R	EPWM Rising Capture Data Register 3	0x0000_0000
EPWM_RCAPDAT4	EPWMx_BA+0x22C	R	EPWM Rising Capture Data Register 4	0x0000_0000
EPWM_RCAPDAT5	EPWMx_BA+0x234	R	EPWM Rising Capture Data Register 5	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RCAPDAT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RCAPDAT							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**RCAPDAT**	**EPWM Rising Capture Data Register (Read Only)** When rising capture condition happened, the EPWM counter value will be saved in this register.

EPWM Falling Capture Data Register 0~5 (EPWM_FCAPDAT 0~5)

Register	Offset	R/W	Description	Reset Value
EPWM_FCAPDAT0	EPWMx_BA+0x210	R	EPWM Falling Capture Data Register 0	0x0000_0000
EPWM_FCAPDAT1	EPWMx_BA+0x218	R	EPWM Falling Capture Data Register 1	0x0000_0000
EPWM_FCAPDAT2	EPWMx_BA+0x220	R	EPWM Falling Capture Data Register 2	0x0000_0000
EPWM_FCAPDAT3	EPWMx_BA+0x228	R	EPWM Falling Capture Data Register 3	0x0000_0000
EPWM_FCAPDAT4	EPWMx_BA+0x230	R	EPWM Falling Capture Data Register 4	0x0000_0000
EPWM_FCAPDAT5	EPWMx_BA+0x238	R	EPWM Falling Capture Data Register 5	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
FCAPDAT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FCAPDAT							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**FCAPDAT**	**EPWM Falling Capture Data Register (Read Only)** When falling capture condition happened, the EPWM counter value will be saved in this register.

EPWM PDMA Control Register (EPWM_PDMACTL)

Register	Offset	R/W	Description	Reset Value
EPWM_PDMACTL	EPWMx_BA+0x23C	R/W	EPWM PDMA Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			CHSEL4_5	CAPORD4_5	CAPMOD4_5		CHEN4_5
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			CHSEL2_3	CAPORD2_3	CAPMOD2_3		CHEN2_3
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			CHSEL0_1	CAPORD0_1	CAPMOD0_1		CHEN0_1

Bits	Description	
[31:21]	**Reserved**	Reserved.
[20]	**CHSEL4_5**	**Select Channel 4/5 to Do PDMA Transfer** 0 = Channel4. 1 = Channel5.
[19]	**CAPORD4_5**	**Capture Channel 4/5 Rising/Falling Order** Set this bit to determine whether the EPWM_RCAPDAT4/5 or EPWM_FCAPDAT4/5 is the first captured data transferred to memory through PDMA when CAPMOD4_5 =11. 0 = EPWM_FCAPDAT4/5 is the first captured data to memory. 1 = EPWM_RCAPDAT4/5 is the first captured data to memory.
[18:17]	**CAPMOD4_5**	**Select EPWM_RCAPDAT4/5 or EPWM_FCAPDAT4/5 to Do PDMA Transfer** 00 = Reserved. 01 = EPWM_RCAPDAT4/5. 10 = EPWM_FCAPDAT4/5. 11 = Both EPWM_RCAPDAT4/5 and EPWM_FCAPDAT4/5.
[16]	**CHEN4_5**	**Channel 4/5 PDMA Enable Bit** 0 = Channel 4/5 PDMA function Disabled. 1 = Channel 4/5 PDMA function Enabled for the channel 4/5 captured data and transfer to memory.
[15:13]	**Reserved**	Reserved.
[12]	**CHSEL2_3**	**Select Channel 2/3 to Do PDMA Transfer** 0 = Channel2. 1 = Channel3.
[11]	**CAPORD2_3**	**Capture Channel 2/3 Rising/Falling Order** Set this bit to determine whether the EPWM_RCAPDAT2/3 or EPWM_FCAPDAT2/3 is the first captured data transferred to memory through PDMA when CAPMOD2_3 =11. 0 = EPWM_FCAPDAT2/3 is the first captured data to memory. 1 = EPWM_RCAPDAT2/3 is the first captured data to memory.

[10:9]	**CAPMOD2_3**	**Select EPWM_RCAPDAT2/3 or EPWM_FCAODAT2/3 to Do PDMA Transfer** 00 = Reserved. 01 = EPWM_RCAPDAT2/3. 10 = EPWM_FCAPDAT2/3. 11 = Both EPWM_RCAPDAT2/3 and EPWM_FCAPDAT2/3.
[8]	**CHEN2_3**	**Channel 2/3 PDMA Enable Bit** 0 = Channel 2/3 PDMA function Disabled. 1 = Channel 2/3 PDMA function Enabled for the channel 2/3 captured data and transfer to memory.
[7:5]	**Reserved**	Reserved.
[4]	**CHSEL0_1**	**Select Channel 0/1 to Do PDMA Transfer** 0 = Channel0. 1 = Channel1.
[3]	**CAPORD0_1**	**Capture Channel 0/1 Rising/Falling Order** Set this bit to determine whether the EPWM_RCAPDAT0/1 or EPWM_FCAPDAT0/1 is the first captured data transferred to memory through PDMA when CAPMOD0_1 =11. 0 = EPWM_FCAPDAT0/1 is the first captured data to memory. 1 = EPWM_RCAPDAT0/1 is the first captured data to memory.
[2:1]	**CAPMOD0_1**	**Select EPWM_RCAPDAT0/1 or EPWM_FCAPDAT0/1 to Do PDMA Transfer** 00 = Reserved. 01 = EPWM_RCAPDAT0/1. 10 = EPWM_FCAPDAT0/1. 11 = Both EPWM_RCAPDAT0/1 and EPWM_FCAPDAT0/1.
[0]	**CHEN0_1**	**Channel 0/1 PDMA Enable Bit** 0 = Channel 0/1 PDMA function Disabled. 1 = Channel 0/1 PDMA function Enabled for the channel 0/1 captured data and transfer to memory.

EPWM Capture Channel 0_1, 2_3, 4_5 PDMA Register (EPWM_PDMACAP 0_1, 2_3, 4_5)

Register	Offset	R/W	Description	Reset Value
EPWM_PDMACAP0_1	EPWMx_BA+0x240	R	EPWM Capture Channel 01 PDMA Register	0x0000_0000
EPWM_PDMACAP2_3	EPWMx_BA+0x244	R	EPWM Capture Channel 23 PDMA Register	0x0000_0000
EPWM_PDMACAP4_5	EPWMx_BA+0x248	R	EPWM Capture Channel 45 PDMA Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CAPBUF							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CAPBUF							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**CAPBUF**	**EPWM Capture PDMA Register (Read Only)** This register is used as a buffer to transfer EPWM capture rising or falling data to memory by PDMA.

EPWM Capture Interrupt Enable Register (EPWM_CAPIEN)

Register	Offset	R/W	Description	Reset Value
EPWM_CAPIEN	EPWMx_BA+0x250	R/W	EPWM Capture Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		CAPFIEN5	CAPFIEN4	CAPFIEN3	CAPFIEN2	CAPFIEN1	CAPFIEN0
7	6	5	4	3	2	1	0
Reserved		CAPRIEN5	CAPRIEN4	CAPRIEN3	CAPRIEN2	CAPRIEN1	CAPRIEN0

Bits	Description	
[31:14]	**Reserved**	Reserved.
[8+n] n=0,1..5	**CAPFIENn**	**EPWM Capture Falling Latch Interrupt Enable Bits** 0 = Capture falling edge latch interrupt Disabled. 1 = Capture falling edge latch interrupt Enabled.
[7:6]	**Reserved**	Reserved.
[n] n=0,1..5	**CAPRIENn**	**EPWM Capture Rising Latch Interrupt Enable Bits** 0 = Capture rising edge latch interrupt Disabled. 1 = Capture rising edge latch interrupt Enabled.

EPWM Capture Interrupt Flag Register (EPWM_CAPIF)

Register	Offset	R/W	Description	Reset Value
EPWM_CAPIF	EPWMx_BA+0x254	R/W	EPWM Capture Interrupt Flag Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		CFLIF5	CFLIF4	CFLIF3	CFLIF2	CFLIF1	CFLIF0
7	6	5	4	3	2	1	0
Reserved		CRLIF5	CRLIF4	CRLIF3	CRLIF2	CRLIF1	CRLIF0

Bits	Description	
[31:14]	**Reserved**	Reserved.
[8+n] n=0,1..5	**CFLIFn**	**EPWM Capture Falling Latch Interrupt Flag** 0 = No capture falling latch condition happened. 1 = Capture falling latch condition happened, this flag will be set to high. **Note 1:** When Capture with PDMA operating, CAPIF corresponding channel CFLIF will be cleared by hardware after PDMA transfer data. **Note 2:** This bit is cleared by writing 1 to it.
[7:6]	**Reserved**	Reserved.
[n] n=0,1..5	**CRLIFn**	**EPWM Capture Rising Latch Interrupt Flag** 0 = No capture rising latch condition happened. 1 = Capture rising latch condition happened, this flag will be set to high. **Note 1:** When Capture with PDMA operating, CAPIF corresponding channel CRLIF will be cleared by hardware after PDMA transfer data. **Note 2:** This bit is cleared by writing 1 to it.

EPWM Period Register Buffer 0~5 (EPWM_PBUF0~5)

Register	Offset	R/W	Description	Reset Value
EPWM_PBUF0	EPWMx_BA+0x304	R	EPWM PERIOD0 Buffer	0x0000_0000
EPWM_PBUF1	EPWMx_BA+0x308	R	EPWM PERIOD1 Buffer	0x0000_0000
EPWM_PBUF2	EPWMx_BA+0x30C	R	EPWM PERIOD2 Buffer	0x0000_0000
EPWM_PBUF3	EPWMx_BA+0x310	R	EPWM PERIOD3 Buffer	0x0000_0000
EPWM_PBUF4	EPWMx_BA+0x314	R	EPWM PERIOD4 Buffer	0x0000_0000
EPWM_PBUF5	EPWMx_BA+0x318	R	EPWM PERIOD5 Buffer	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PBUF							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PBUF							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**PBUF**	**EPWM Period Register Buffer (Read Only)** Used as PERIOD active register.

EPWM Comparator Register Buffer 0~5 (EPWM_CMPBUF0~5)

Register	Offset	R/W	Description	Reset Value
EPWM_CMPBUF0	EPWMx_BA+0x31C	R	EPWM CMPDAT0 Buffer	0x0000_0000
EPWM_CMPBUF1	EPWMx_BA+0x320	R	EPWM CMPDAT1 Buffer	0x0000_0000
EPWM_CMPBUF2	EPWMx_BA+0x324	R	EPWM CMPDAT2 Buffer	0x0000_0000
EPWM_CMPBUF3	EPWMx_BA+0x328	R	EPWM CMPDAT3 Buffer	0x0000_0000
EPWM_CMPBUF4	EPWMx_BA+0x32C	R	EPWM CMPDAT4 Buffer	0x0000_0000
EPWM_CMPBUF5	EPWMx_BA+0x330	R	EPWM CMPDAT5 Buffer	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CMPBUF							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CMPBUF							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**CMPBUF**	**EPWM Comparator Register Buffer (Read Only)** Used as CMP active register.

EPWM CLKPSC Buffer 0_1, 2_3, 4_5 (EPWM_CPSCBUF0_1, 2_3, 4_5)

Register	Offset	R/W	Description	Reset Value
EPWM_CPSCBUF0_1	EPWMx_BA+0x334	R	EPWM CLKPSC0_1 Buffer	0x0000_0000
EPWM_CPSCBUF2_3	EPWMx_BA+0x338	R	EPWM CLKPSC2_3 Buffer	0x0000_0000
EPWM_CPSCBUF4_5	EPWMx_BA+0x33C	R	EPWM CLKPSC4_5 Buffer	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				CPSCBUF			
7	6	5	4	3	2	1	0
CPSCBUF							

Bits	Description	
[31:12]	**Reserved**	Reserved.
[11:0]	**CPSCBUF**	**EPWM Counter Clock Prescale Buffer** Used as EPWM counter clock pre-scare active register.

EPWM_FTCMPDAT Buffer (EPWM_FTCBUF0_1,2_3,4_5)

Register	Offset	R/W	Description	Reset Value
EPWM_FTCBUF0_1	EPWMx_BA+0x340	R	EPWM FTCMPDAT0_1 Buffer	0x0000_0000
EPWM_FTCBUF2_3	EPWMx_BA+0x344	R	EPWM FTCMPDAT2_3 Buffer	0x0000_0000
EPWM_FTCBUF4_5	EPWMx_BA+0x348	R	EPWM FTCMPDAT4_5 Buffer	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
FTCMPBUF							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FTCMPBUF							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**FTCMPBUF**	**EPWM FTCMPDAT Buffer (Read Only)** Used as FTCMPDAT active register.

EPWM FTCMPDAT Indicator Register (EPWM_FTCI)

Register	Offset	R/W	Description	Reset Value
EPWM_FTCI	EPWMx_BA+0x34C	R/W	EPWM FTCMPDAT Indicator Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					FTCMD4	FTCMD2	FTCMD0
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					FTCMU4	FTCMU2	FTCMU0

Bits	Description	
[31:11]	**Reserved**	Reserved.
[8+n/2]n=0,2,4	**FTCMDn**	**EPWM FTCMPDAT Down Indicator** Indicator is set by hardware when EPWM counter down count and reaches EPWM_FTCMPDATn; software can clear this bit by writing 1 to it.
[7:3]	**Reserved**	Reserved.
[n/2] n=0,2,4	**FTCMUn**	**EPWM FTCMPDAT Up Indicator** Indicator is set by hardware when EPWM counter up count and reaches EPWM_FTCMPDATn; software can clear this bit by writing 1 to it.

6.19 Enhanced Input Capture Timer (ECAP)

6.19.1 Overview

This device provides up to three units of Input Capture Timer/Counter whose capture function can detect the digital edge-changed signal at channel inputs. Each unit has three input capture channels. The timer/counter is equipped with up counting, reload and compare-match capabilities.

6.19.2 Features

- Up to three Input Capture Timer/Counter units, CAP0, CAP1 and CAP2.
- Each unit has 3 input channels.
- Each unit has its own interrupt vector.
- Each input channel has its own capture counter hold register.
- 24-bit Input Capture up-counting timer/counter.
- With noise filter in front end of input ports.
- Edge detector with three options:
  - Rising edge detection
  - Falling edge detection
  - Both edge detection
- Captured events reset and/or reload capture counter.
- Supports compare-match function.

6.19.3 Block Diagram

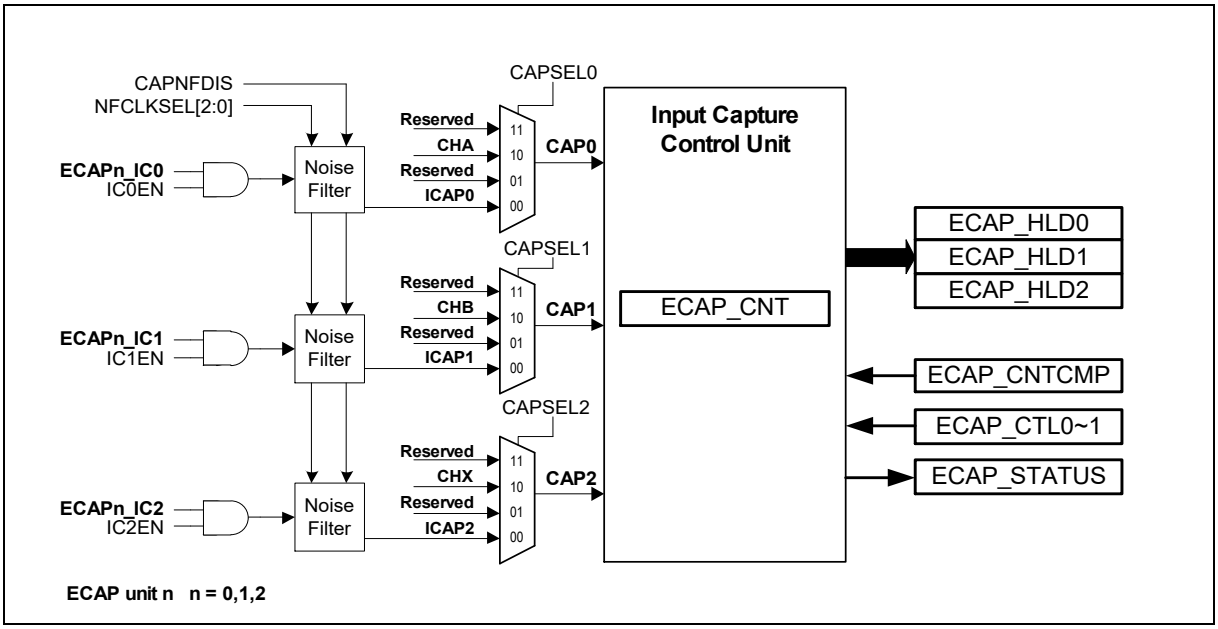


Figure 6.19-1 Input Capture Timer/Counter Architecture

6.19.4 Basic Configuration

6.19.4.1 ECAP0 Basic Configuration

- Clock Source Configuration
  - Enable ECAP0 peripheral clock in ECAP0CKEN(CLK_APBCLK2[8]).
- Reset Configuration
  - Reset ECAP0 peripheral in ECAP0RST (SYS_IPRST2[26]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
ECAP0	ECAP0_IC0	PD.13	MFP9
		PG.5	MFP6
		PI.7, PL.3	MFP12
	ECAP0_IC1	PL.4, PK.0	MFP12
		PG.6	MFP6
		PD.14	MFP9
	ECAP0_IC2	PD.15	MFP9
		PG.7	MFP6
		PK.1, PL.5	MFP12

6.19.4.2 ECAP1 Basic Configuration

- Clock Source Configuration
  - Enable ECAP1 peripheral clock in ECAP1CKEN (CLK_APBCLK2[9]).
- Reset Configuration
  - Reset ECAP1 peripheral in ECAP1RST (SYS_IPRST2[27]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
ECAP1	ECAP1_IC0	PL.12	MFP3
		PL.6	MFP12
	ECAP1_IC1	PL.13	MFP3
		PM.14	MFP9
	ECAP1_IC2	PG.4	MFP14
		PM.15	MFP9
		PI.2	MFP12

6.19.4.3 ECAP2 Basic Configuration

- Clock Source Configuration
  - Enable ECAP2 peripheral clock in ECAP2CKEN (CLK_APBCLK2[10]).

- Reset Configuration
  - Reset ECAP2 peripheral in ECAP2RST (SYS_IPRST3[21]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
ECAP2	ECAP2_IC0	PB.12	MFP14
	ECAP2_IC1	PB.13	MFP14
	ECAP2_IC2	PB.14	MFP14

6.19.5 Functional Description

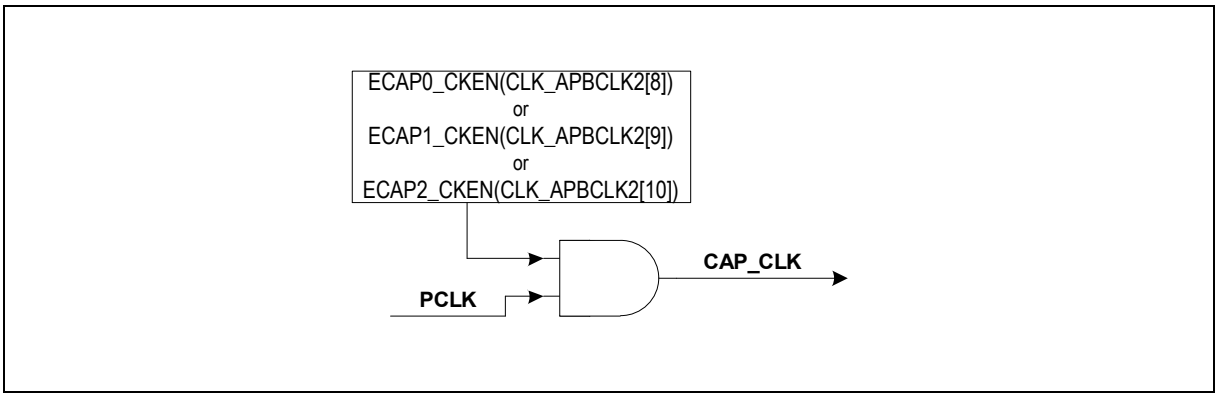


Figure 6.19-2 Input Capture Timer/Counter Clock Source Control

Figure 6.19-1 illustrates the architecture of the Input Capture. Each input capture timer/counter unit supports 3 input channels with programmable input signal sources. The port pins ECAP_IC0 to ECAP_IC2 can be fed to the inputs of capture unit through noise filter or bypass it (CAPNFDIS (ECAP_CTL0[3] = 1), and the QEI controller input signals (CHA, CHB and CHX) also can be internally routed to the capture inputs by setting the register ECAP_CTL0 (CAPSEL0~ CAPSEL 2).

6.19.5.1 Input Noise Filter

The architecture of input noise filter is similar to that one used for QEI. Refer to the figure - Noise Filter - in QEI section . With 6 sampling-rate options, it supports a wide range of filtering noise whose duration is from 55 ns (NFCLKSEL (ECAP_CTL0[2:0]) = 000) to 3.5 us (NFCLKSEL (ECAP_CTL0[2:0]) = 101) while PCLK is running at the frequency of 55 MHz. Table 6.19-1 lists the relation between the setting of NFCLKSEL (ECAP_CTL0[2:0]), the duration of filtered noise and the duration of the signal that is guaranteed to be sampled.

The maximum duration of the noise that is filtered out should not be more than 3 clock cycle, while the minimum duration of the signal that is guaranteed to be sampled should be more than 4 clock cycle.

NFCLKSEL(ECAP_CTL0[2:0])	Maximum Duration Of Noise	Minimum Duration Of Signal
000	55 ns	73 ns
001	109 ns	145 ns
010	218 ns	291 ns
011	873 ns	1164 ns
100	1745 ns	2327 ns

101	3491 ns	4655 ns
		PCLK= 55 MHz

Table 6.19-1 Typical Case of Noise Filter Settings

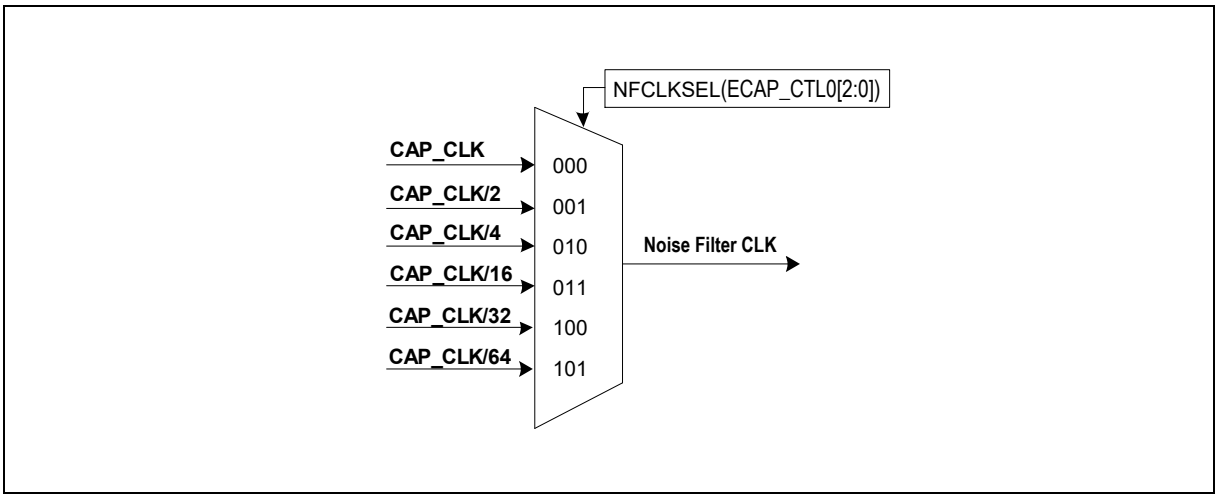


Figure 6.19-3 Noise Filter Sampling Clock Selection

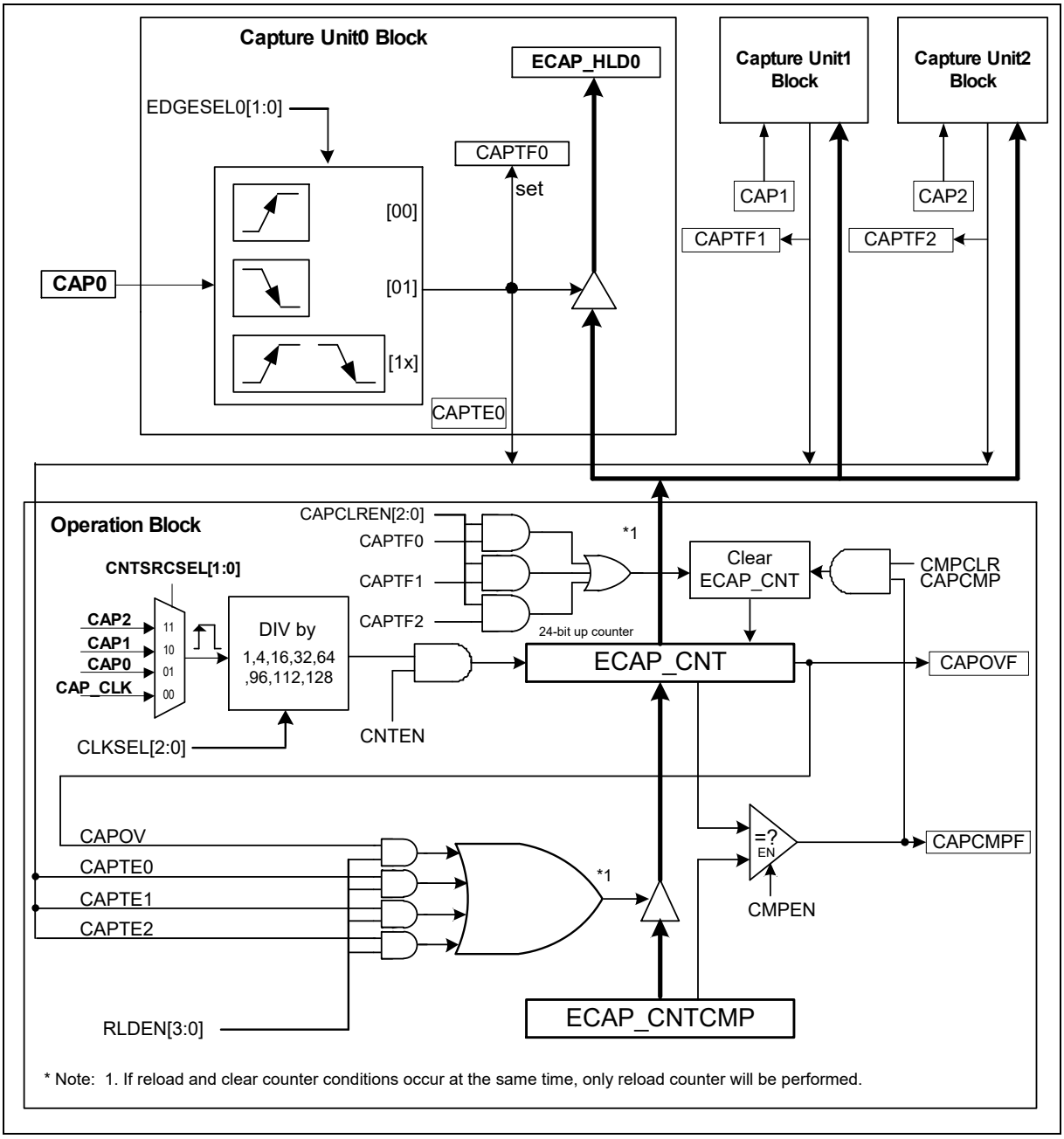


Figure 6.19-4 Input Capture Timer/Counter Function Block

6.19.5.2 Input Capture Timer/Counter Operation

An Input Capture Timer/Counter unit consists of 2 main functional blocks, Capture block and Operation block. There are 3 Input Capture units in Capture block for 3 input channels.

The capture unit function as detecting and measuring the pulse width and the period of a square wave. The input channel 0 to 2 have their own edge detectors, which are in Input Capture block but share with one capture timer/counter, ECAP_CNT, which is in Operation block. The edge trigger option is programmable through EDGESEL (ECAP_CTL1[5,4], ECAP_CTL1 [3,2], ECAP_CTL1 [1,0]) register supporting positive edge, negative edge and both edge triggers. Each capture unit consists of an enable control bit, IC0EN ~ IC2EN (ECAP_CTL0[6:4]) to enable/disable each input channel and a status bit

CAP0 ~ CAP2 (ECAP_STATUS[10:8]) to let software monitor the current status of each channel.

The Input Capture supports reload mode and compare mode. For both modes, the capture counter (ECAP_CNT) serves as a 24-bit up-counting counter whose clock comes from the output of the clock divider and is gated with CNTEN (ECAP_CTL0[24]), and the clock source of the clock divider, which can be set by CLKSEL (ECAP_CTL1 [14:12]) to divide clock by 1, 4, 16, 32, 64, 96, 112 and 128, is programmable (by setting CNTSRCSEL (ECAP_CTL1[17:16])) to be from system clock source, CAP_CLK or input channel CAP0 ~ CAP2(ECAP_STATUS[10:8]). In reload mode, ECAP_CNTCMP serves as a reload register while in compare mode ECAP_CNTCMP serves as a compare register. The Input Capture Timer/Counter Enable bit (CAPEN (ECAP_CTL0[29])) must be set to enable Input Capture Timer/Counter functions. More details of operation are described in the following.

Capture Function

Each time when the capture input detect a valid edge change, it triggers a valid capture event (CAPTE0~2) so that the content of the free running 24-bit capture counter ECAP_CNT will be captured/transferred into the capture hold registers, ECAP_HLD0~2 depending on which channel is triggered. This event also causes the corresponding flag CAPTFx (ECAP_STATUS[2:0]) to be set, which will generate an interrupt if the corresponding interrupt enable bit CAPIENx (ECAP_CTL0[18:16]) is set. Triggered Flags are set by hardware and should be cleared by software. Software can read the register ECAP_STATUS to get the status of flags and has to write 1 to the corresponding bit(s) of ECAP_STATUS to clear flag(s).

In addition, setting the CAPxCLREN(ECAP_CTL1[22:20]) will allow hardware to reset capture counter (ECAP_CNT) automatically whenever the event happens.

Compare Mode

The compare function is enabled by setting the CMPEN (ECAP_CTL0[28]) bit to 1, and ECAP_CNTCMP will serve as a compare register. As ECAP_CNT counting up, upon matching ECAP_CNTCMP value, the flag, CAPCMPF (ECAP_STATUS[4]), will be set, which will generate an interrupt, CMP_INT, if compare interrupt enable bit, CMPIEN (ECAP_CTL0[21]), is set.

Besides, setting the CMPCLREN (ECAP_CTL0[25]) will allow hardware to make capture counter cleared to zero automatically after a compare-match event occurs.

Reload Mode

The Input Capture Timer/Counter can also be configured for reload mode. The reload function is enabled by setting OVRLDEN (ECAP_CTL1[11]), CAPxRLDEN (ECAP_CTL1[10:8]) to 1, and each bit enables a reload source.

In this mode, ECAP_CNTCMP serves as a reload register. If OVRLDEN (ECAP_CTL1[11]) is set, a reload event is generated and causes the content of the ECAP_CNTCMP register to be loaded into the ECAP_CNT register when ECAP_CNT overflows. Furthermore, CAPxRLDEN (ECAP_CTL1[10], ECAP_CTL1[9], ECAP_CTL1[8]), x=0~2, are enable bits of making CAPTE2 ~ CAPTE0 as reload sources.

One thing should be noted is that if CAPxCLREN (ECAP_CTL1[22], ECAP_CTL1[21], ECAP_CTL1[20]) as well as CAPxRLDEN (ECAP_CTL1[10], ECAP_CTL1[9], ECAP_CTL1[8]) are set, when a valid trigger event (CAPTEx) occurs, only RELAOD function will be executed.

6.19.5.3 Input Capture Timer/Counter Interrupt Architecture

Figure 6.19-5 demonstrates the architecture of Input Capture Timer/Counter interrupt module. There are 5 interrupt sources (OVF_INT, CMP_INT, CAPTF0_INT~CAPTF2_INT), which are logical 'OR' together, in a input capture unit, and each one has an interrupt flag (CAPOVF (ECAP_STATUS[5]), CAPCMPF(ECAP_STATUS[4]), CAPTF0 ~ CAPTF2 (ECAP_STATUS[2:0])), which can trigger Interrupt (ECAP_INT), as well as an the enable control bit (OVIEN (ECAP_CTL0[20]), CMPIEN (ECAP_CTL0[21]), CAPIEN0 ~ CAPIEN2 (ECAP_CTL0[18:16])) to enable/disable the flag.

Note that all the interrupt flags are set by hardware and must be cleared by software by writing 1 to the

bit (ECAP_STATUS[5:4], ECAP_STATUS [2:0]) corresponding to the flag.

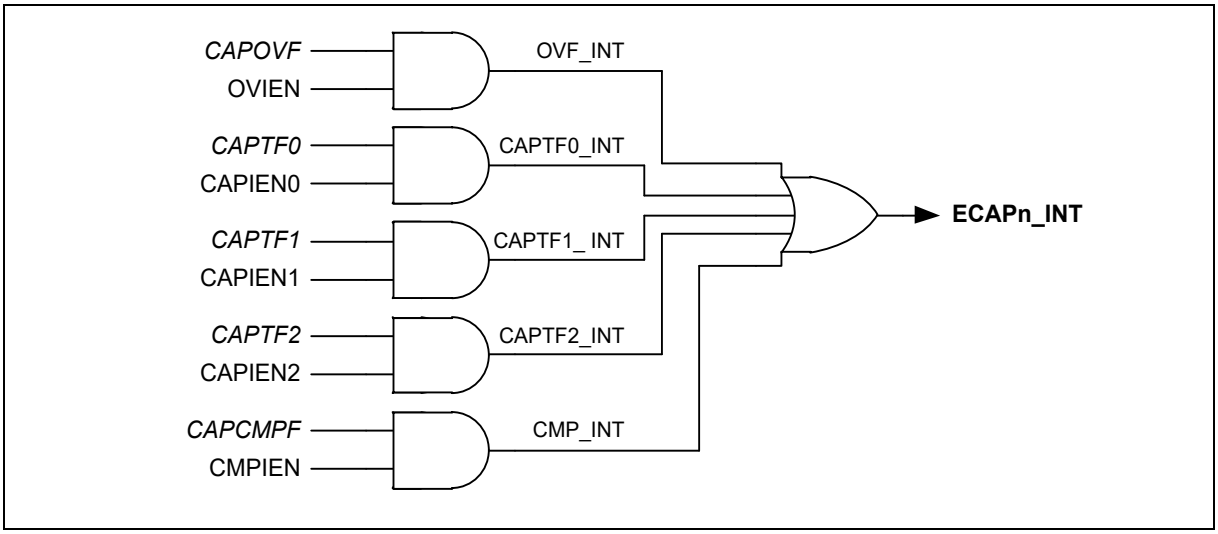


Figure 6.19-5 Input Capture Timer/Counter Interrupt Architecture Diagram

6.19.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
ECAP Base Address: **ECAPn_BA = 0x40B4_0000 + (0x0001_0000*n)** **n=0, 1, 2**				
ECAP_CNT	ECAPn_BA+0x00	R/W	Input Capture Counter (24-bit Up Counter)	0x0000_0000
ECAP_HLD0	ECAPn_BA+0x04	R/W	Input Capture Hold Register 0	0x0000_0000
ECAP_HLD1	ECAPn_BA+0x08	R/W	Input Capture Hold Register 1	0x0000_0000
ECAP_HLD2	ECAPn_BA+0x0C	R/W	Input Capture Hold Register 2	0x0000_0000
ECAP_CNTCMP	ECAPn_BA+0x10	R/W	Input Capture Compare Register	0x0000_0000
ECAP_CTL0	ECAPn_BA+0x14	R/W	Input Capture Control Register 0	0x0000_0000
ECAP_CTL1	ECAPn_BA+0x18	R/W	Input Capture Control Register 1	0x0000_0000
ECAP_STATUS	ECAPn_BA+0x1C	R/W	Input Capture Status Register	0x0000_0000

6.19.7 Register Description

Input Capture Counter (ECAP_CNT)

Register	Offset	R/W	Description	Reset Value
ECAP_CNT	ECAPn_BA+0x00	R/W	Input Capture Counter (24-bit Up Counter)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CNT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CNT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNT							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**CNT**	**Input Capture Timer/Counter** The input Capture Timer/Counter is a 24-bit up-counting counter. The clock source for the counter is from the clock divider.

Input Capture Counter Hold Register (ECAP_HLD0~2)

Register	Offset	R/W	Description	Reset Value
ECAP_HLD0	ECAPn_BA+0x04	R/W	Input Capture Hold Register 0	0x0000_0000
ECAP_HLD1	ECAPn_BA+0x08	R/W	Input Capture Hold Register 1	0x0000_0000
ECAP_HLD2	ECAPn_BA+0x0C	R/W	Input Capture Hold Register 2	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
HOLD							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
HOLD							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
HOLD							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**HOLD**	**Input Capture Counter Hold Register** When an active input capture channel detects a valid edge signal change, the ECAPCNT value is latched into the corresponding holding register. Each input channel has its own holding register named by ECAP_HLDx where x is from 0 to 2 to indicate inputs from IC0 to IC2, respectively.

Input Capture Counter Compare Register (ECAP_CNTCMP)

Register	Offset	R/W	Description	Reset Value
ECAP_CNTCMP	ECAPn_BA+0x10	R/W	Input Capture Compare Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CNTCMP							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CNTCMP							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNTCMP							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**CNTCMP**	**Input Capture Counter Compare Register** If the compare function is enabled (CMPEN (ECAP_CTL0[28]) = 1), this register (ECAP_CNTCMP) is used to compare with the capture counter (ECAP_CNT). If the reload control is enabled (RLDEN[n] = 1, n=0~3), an overflow event or capture events will trigger the hardware to load the value of this register (ECAP_CNTCMP) into ECAP_CNT.

Input Capture Timer/Counter Control Register (ECAP_CTL0)

Register	Offset	R/W	Description	Reset Value
ECAP_CTL0	ECAPn_BA+0x14	R/W	Input Capture Control Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		CAPEN	CMPEN	Reserved		CMPCLREN	CNTEN
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		CMPIEN	OVIEN	Reserved	CAPIEN2	CAPIEN1	CAPIEN0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		CAPSEL2		CAPSEL1		CAPSEL0	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	IC2EN	IC1EN	IC0EN	CAPNFDIS	NFCLKSEL		

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29]	**CAPEN**	**Input Capture Timer/Counter Enable Bit** 0 = Input Capture function Disabled. 1 = Input Capture function Enabled.
[28]	**CMPEN**	**Compare Function Enable Bit** The compare function in input capture timer/counter is to compare the dynamic counting ECAP_CNT with the compare register ECAP_CNTCMP, if ECAP_CNT value reaches ECAP_CNTCMP, the flag CAPCMPF (ECAP_STATUS[4]) will be set. 0 = The compare function Disabled. 1 = The compare function Enabled.
[27:26]	**Reserved**	Reserved.
[25]	**CMPCLREN**	**Input Capture Counter Cleared by Compare-match Control** If this bit is set to 1, the capture counter (ECAP_CNT) will be cleared to 0 when the compare-match event (CAPCMPF (ECAP_STATUS[4]) = 1) occurs. 0 = Compare-match event (CAPCMPF (ECAP_STATUS[4])) can clear capture counter (ECAP_CNT) Disabled. 1 = Compare-match event (CAPCMPF (ECAP_STATUS[4])) can clear capture counter (ECAP_CNT) Enabled.
[24]	**CNTEN**	**Input Capture Counter Start Counting Control** Setting this bit to 1, the capture counter (ECAP_CNT) starts up-counting synchronously with the clock. 0 = ECAP_CNT stops counting. 1 = ECAP_CNT starts up-counting.
[23:22]	**Reserved**	Reserved.
[21]	**CMPIEN**	**CAPCMPF Trigger Input Capture Interrupt Enable Bit** 0 = The flag CAPCMPF (ECAP_STATUS[4]) can trigger Input Capture interrupt Disabled. 1 = The flag CAPCMPF (ECAP_STATUS[4]) can trigger Input Capture interrupt Enabled.

Bits	Description	
[20]	**OVIEN**	**CAPOVF Trigger Input Capture Interrupt Enable Bit** 0 = The flag CAPOVF (ECAP_STATUS[5]) can trigger Input Capture interrupt Disabled. 1 = The flag CAPOVF (ECAP_STATUS[5]) can trigger Input Capture interrupt Enabled.
[19]	**Reserved**	Reserved.
[18]	**CAPIEN2**	**Input Capture Channel 2 Interrupt Enable Bit** 0 = The flag CAPTF2 (ECAP_STATUS[2]) can trigger Input Capture interrupt Disabled. 1 = The flag CAPTF2 (ECAP_STATUS[2]) can trigger Input Capture interrupt Enabled.
[17]	**CAPIEN1**	**Input Capture Channel 1 Interrupt Enable Bit** 0 = The flag CAPTF1 (ECAP_STATUS[1]) can trigger Input Capture interrupt Disabled. 1 = The flag CAPTF1 (ECAP_STATUS[1]) can trigger Input Capture interrupt Enabled.
[16]	**CAPIEN0**	**Input Capture Channel 0 Interrupt Enable Bit** 0 = The flag CAPTF0 (ECAP_STATUS[0]) can trigger Input Capture interrupt Disabled. 1 = The flag CAPTF0 (ECAP_STATUS[0]) can trigger Input Capture interrupt Enabled.
[15:14]	**Reserved**	Reserved.
[13:12]	**CAPSEL2**	**CAP2 Input Source Selection** 00 = CAP2 input is from port pin ICAP2. 01 = Reserved. 10 = CAP2 input is from signal CHX of QEI controller unit n. 11 = Reserved. **Note:** Input capture unit n matches QEIn, where n = 0~1.
[11:10]	**CAPSEL1**	**CAP1 Input Source Selection** 00 = CAP1 input is from port pin ICAP1. 01 = Reserved. 10 = CAP1 input is from signal CHB of QEI controller unit n. 11 = Reserved. **Note:** Input capture unit n matches QEIn, where n = 0~1.
[9:8]	**CAPSEL0**	**CAP0 Input Source Selection** 00 = CAP0 input is from port pin ICAP0. 01 = Reserved. 10 = CAP0 input is from signal CHA of QEI controller unit n. 11 = Reserved. **Note:** Input capture unit n matches QEIn, where n = 0~1.
[7]	**Reserved**	Reserved.
[6]	**IC2EN**	**Port Pin IC2 Input to Input Capture Unit Enable Bit** 0 = IC2 input to Input Capture Unit Disabled. 1 = IC2 input to Input Capture Unit Enabled.
[5]	**IC1EN**	**Port Pin IC1 Input to Input Capture Unit Enable Bit** 0 = IC1 input to Input Capture Unit Disabled. 1 = IC1 input to Input Capture Unit Enabled.
[4]	**IC0EN**	**Port Pin IC0 Input to Input Capture Unit Enable Bit** 0 = IC0 input to Input Capture Unit Disabled. 1 = IC0 input to Input Capture Unit Enabled.

Bits	Description	
[3]	**CAPNFDIS**	**Input Capture Noise Filter Disable Control** 0 = Noise filter of Input Capture Enabled. 1 = Noise filter of Input Capture Disabled (Bypass).
[2:0]	**NFCLKSEL**	**Noise Filter Clock Pre-divide Selection** To determine the sampling frequency of the Noise Filter clock 000 = CAP_CLK. 001 = CAP_CLK/2. 010 = CAP_CLK/4. 011 = CAP_CLK/16. 100 = CAP_CLK/32. 101 = CAP_CLK/64.

Input Capture Timer/Counter Control Register (ECAP_CTL1)

Register	Offset	R/W	Description	Reset Value
ECAP_CTL1	ECAPn_BA+0x18	R/W	Input Capture Control Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved	CAP2CLREN	CAP1CLREN	CAP0CLREN	Reserved		CNTSRCSEL	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	CLKSEL			OVRLDEN	CAP2RLDEN	CAP1RLDEN	CAP0RLDEN
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		EDGESEL2		EDGESEL1		EDGESEL0	

Bits	Description	
[31:23]	**Reserved**	Reserved.
[22]	**CAP2CLREN**	**Capture Counter Cleared by Capture Event2 Control** 0 = Event CAPTE2 can clear capture counter (ECAP_CNT[23:0]) Disabled. 1 = Event CAPTE2 can clear capture counter (ECAP_CNT[23:0]) Enabled.
[21]	**CAP1CLREN**	**Capture Counter Cleared by Capture Event1 Control** 0 = Event CAPTE1 can clear capture counter (ECAP_CNT[23:0]) Disabled. 1 = Event CAPTE1 can clear capture counter (ECAP_CNT[23:0]) Enabled.
[20]	**CAP0CLREN**	**Capture Counter Cleared by Capture Event0 Control** 0 = Event CAPTE0 can clear capture counter (ECAP_CNT[23:0]) Disabled. 1 = Event CAPTE0 can clear capture counter (ECAP_CNT[23:0]) Enabled.
[19:18]	**Reserved**	Reserved.
[17:16]	**CNTSRCSEL**	**Capture Timer/Counter Clock Source Selection** Select the capture timer/counter clock source. 00 = CAP_CLK (default). 01 = CAP0. 10 = CAP1. 11 = CAP2.
[15]	**Reserved**	Reserved.

Bits	Description	
[14:12]	**CLKSEL**	**Capture Timer Clock Divide Selection** The capture timer clock has a pre-divider with eight divided options controlled by CLKSEL. 000 = CAP_CLK/1. 001 = CAP_CLK/4. 010 = CAP_CLK/16. 011 = CAP_CLK/32. 100 = CAP_CLK/64. 101 = CAP_CLK/96. 110 = CAP_CLK/112. 111 = CAP_CLK/128.
[11]	**OVRLDEN**	**Capture Counter's Reload Function Triggered by Overflow Enable Bit** 0 = The reload triggered by OVIEN (ECAP_CTL0[20]) Disabled. 1 = The reload triggered by OVIEN (ECAP_CTL0[20]) Enabled.
[10]	**CAP2RLDEN**	**Capture Counter's Reload Function Triggered by Event CAPTE2 Enable Bit** 0 = The reload triggered by Event CAPTE2 Disabled. 1 = The reload triggered by Event CAPTE2 Enabled.
[9]	**CAP1RLDEN**	**Capture Counter's Reload Function Triggered by Event CAPTE1 Enable Bit** 0 = The reload triggered by Event CAPTE1 Disabled. 1 = The reload triggered by Event CAPTE1 Enabled.
[8]	**CAP0RLDEN**	**Capture Counter's Reload Function Triggered by Event CAPTE0 Enable Bit** 0 = The reload triggered by Event CAPTE0 Disabled. 1 = The reload triggered by Event CAPTE0 Enabled.
[7:6]	**Reserved**	Reserved.
[5:4]	**EDGESEL2**	**Channel 2 Captured Edge Selection** Input capture2 can detect falling edge change only, rising edge change only or both edge changes 00 = Detect rising edge only. 01 = Detect falling edge only. 1x = Detect both rising and falling edge.
[3:2]	**EDGESEL1**	**Channel 1 Captured Edge Selection** Input capture1 can detect falling edge change only, rising edge change only or both edge change 00 = Detect rising edge only. 01 = Detect falling edge only. 1x = Detect both rising and falling edge.
[1:0]	**EDGESEL0**	**Channel 0 Captured Edge Selection** Input capture0 can detect falling edge change only, rising edge change only or both edge change 00 = Detect rising edge only. 01 = Detect falling edge only. 1x = Detect both rising and falling edge.

Input Capture Timer/Counter Status Register (ECAP_STATUS)

Register	Offset	R/W	Description	Reset Value
ECAP_STATUS	ECAPn_BA+0x1C	R/W	Input Capture Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved					CAP2	CAP1	CAP0
7	6	5	4	3	2	1	0
Reserved		CAPOVF	CAPCMPF	Reserved	CAPTF2	CAPTF1	CAPTF0

Bits	Description	
[31:11]	**Reserved**	Reserved.
[10]	**CAP2**	**Value of Input Channel 2, CAP2 (Read Only)** Reflecting the value of input channel 2, CAP2. **Note:** The bit is read only and write is ignored.
[9]	**CAP1**	**Value of Input Channel 1, CAP1 (Read Only)** Reflecting the value of input channel 1, CAP1 **Note:** The bit is read only and write is ignored.
[8]	**CAP0**	**Value of Input Channel 0, CAP0 (Read Only)** Reflecting the value of input channel 0, CAP0 **Note:** The bit is read only and write is ignored.
[7:6]	**Reserved**	Reserved.
[5]	**CAPOVF**	**Input Capture Counter Overflow Flag** Flag is set by hardware when counter (ECAP_CNT) overflows from 0x00FF_FFFF to zero. 0 = No overflow event has occurred since last clear. 1 = Overflow event(s) has/have occurred since last clear. **Note:** This bit is only cleared by writing 1 to it.
[4]	**CAPCMPF**	**Input Capture Compare-match Flag** If the input capture compare function is enabled, the flag is set by hardware when capture counter (ECAP_CNT) up counts and reaches the ECAP_CNTCMP value. 0 = ECAP_CNT has not matched ECAP_CNTCMP value since last clear. 1 = ECAP_CNT has matched ECAP_CNTCMP value at least once since last clear. **Note:** This bit is only cleared by writing 1 to it.
[3]	**Reserved**	Reserved.

Bits	Description	
[2]	**CAPTF2**	**Input Capture Channel 2 Triggered Flag** When the input capture channel 2 detects a valid edge change at CAP2 input, it will set flag CAPTF2 to high. 0 = No valid edge change has been detected at CAP2 input since last clear. 1 = At least a valid edge change has been detected at CAP2 input since last clear. **Note:** This bit is only cleared by writing 1 to it.
[1]	**CAPTF1**	**Input Capture Channel 1 Triggered Flag** When the input capture channel 1 detects a valid edge change at CAP1 input, it will set flag CAPTF1 to high. 0 = No valid edge change has been detected at CAP1 input since last clear. 1 = At least a valid edge change has been detected at CAP1 input since last clear. **Note:** This bit is only cleared by writing 1 to it.
[0]	**CAPTF0**	**Input Capture Channel 0 Triggered Flag** When the input capture channel 0 detects a valid edge change at CAP0 input, it will set flag CAPTF0 to high. 0 = No valid edge change has been detected at CAP0 input since last clear. 1 = At least a valid edge change has been detected at CAP0 input since last clear. **Note:** This bit is only cleared by writing 1 to it.

6.20 Quadrature Encoder Interface (QEI)

6.20.1 Overview

There are three Quadrature Encoder Interfaces (QEI) QEI controllers in this device. The QEI decodes speed of rotation and motion sensor information and can be used in any application that uses a quadrature encoder for feedback.

6.20.2 Features

6.20.2.1 Quadrature Encoder Interface (QEI) Features

- Up to three QEI controllers, QEI0, QEI1 and QEI2.
- Two QEI phase inputs, QEA and QEB; One Index input.
- A 32-bit up/down Quadrature Encoder Pulse Counter (QEI_CNT)
- A 32-bit software-latch Quadrature Encoder Pulse Counter Hold Register (QEI_CNTHOLD)
- A 32-bit Quadrature Encoder Pulse Counter Index Latch Register (QEI_CNTLATCH)
- A 32-bit Quadrature Encoder Pulse Counter Compare Register (QEI_CNTCMP) with a Pre-set Maximum Count Register (QEI_CNTMAX)
- One QEI control register (QEI_CTL) and one QEI Status Register (QEI_STATUS)
- Four Quadrature encoder pulse counter operation modes
  - Supports x4 free-counting mode
  - Supports x2 free-counting mode
  - Supports x4 compare-counting mode
  - Supports x2 compare-counting mode
- Encoder Pulse Width measurement mode
- Input frequency of QEA/QEB/IDX without noise filter must lower than PCLK/4
- Input frequency of QEA/QEB/IDX with noise filter must lower than Noise Filter Clk/8

6.20.3 Block Diagram

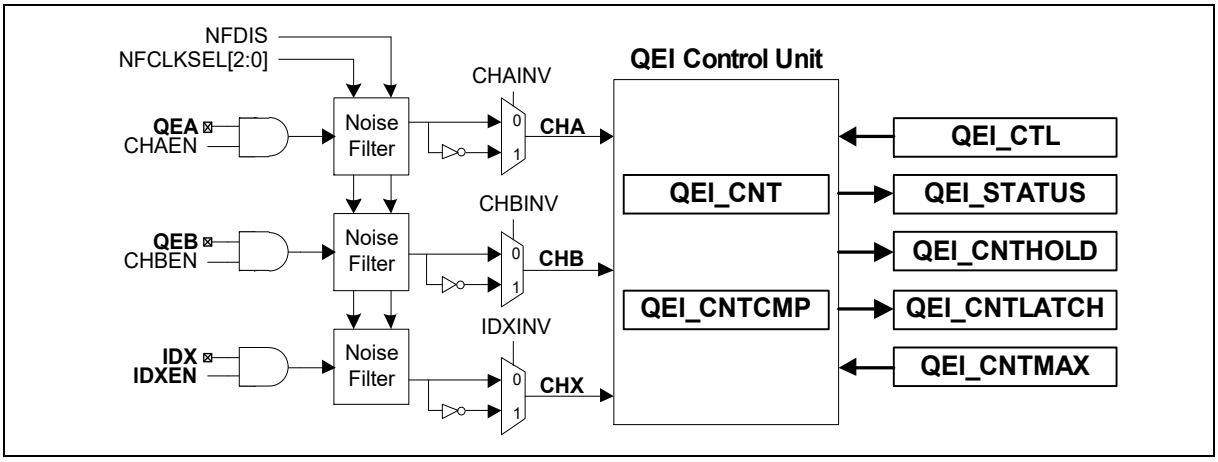


Figure 6.20-1 QEI Block Diagram

The QEI controller inputs, QEA and QEB, accept the outputs from a quadrature encoded source, such as incremental optical shaft encoder. Two channels, A and B, nominally 90 degrees out of phase, are

required. A quadrature encoder usually provides an index signal (to pin IDX) which can be used to indicate an absolute position. There is a noise filter and polarity control for each signal before QEI control unit.

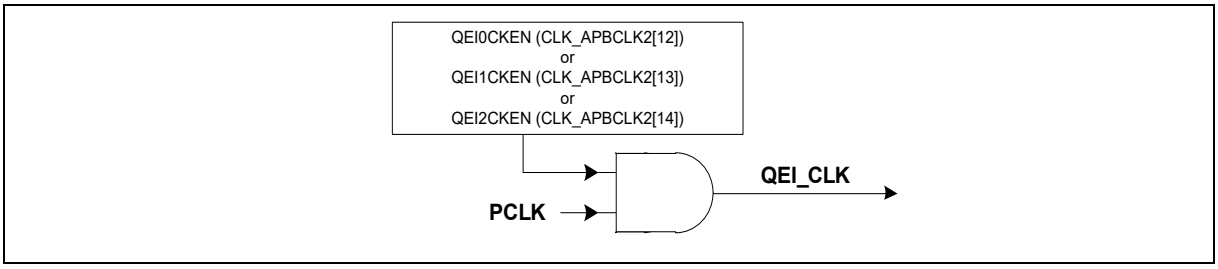


Figure 6.20-2 QEI Clock Source Control

6.20.4 Basic Configuration

6.20.4.1 QEI0 Basic Configuration

- Clock Source Configuration
  - Enable QEI0 peripheral clock in QEI0CKEN (CLK_APBCLK2[12]).
- Reset Configuration
  - Reset QEI0 controller in QEI0RST (SYS_IPRST2[22]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
QEI0	QEI0_A	PG.2	MFP10
		PA.15	MFP9
		PL.9, PK.2	MFP12
	QEI0_B	PG.1	MFP9
		PG.3	MFP10
		PK.3, PL.10	MFP12
	QEI0_INDEX	PD.12	MFP9
		PG.0	MFP10
		PJ.12, PL.11	MFP12

6.20.4.2 QEI1 Basic Configuration

- Clock Source Configuration
  - Enable QEI1 peripheral clock in QEI1CKEN (CLK_APBCLK2[13]).
- Reset Configuration
  - Reset QEI1 controller in QEI1RST (SYS_IPRST2[23]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
QEI1	QEI1_A	PL.0	MFP12
	QEI1_B	PL.1	MFP12

	QEI1_INDEX	PG.4	MFP6
		PI.6, PL.2, PK.8	MFP12

6.20.4.3 QEI2 Basic Configuration

- Clock Source Configuration
  - Enable QEI2 peripheral clock in QEI2CKEN (CLK_APBCLK2[14]).
- Reset Configuration
  - Reset QEI2 controller in QEI2RST (SYS_IPRST3[22]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
QEI2	QEI2_A	PB.9	MFP14
	QEI2_B	PB.10	MFP14
	QEI2_INDEX	PF.14, PB.8, PB.11	MFP14

6.20.5 Functional Description

The QEI control logic detects the relation of phase lead/lag between the filtered signals CHA and CHB and CHX to produce direction indication bit DIRF(QEI_STATUS[8]) to control the pulse counter. The comparator/reload logic compares the pulse counter and maximum count and control the function of reloading pulse counter in compare-counting mode. In Free-counting mode the pulse counter CNT(QEI_CNT[31:0]) will count until the 0xFFFF_FFFF value; while in Compare-counting mode the pulse counter will count until the CNTMAX (QEI_CNTMAX[31:0]) and the pulse counter will be reset to 0 to restart the next cyclic counting.

6.20.5.1 Input Noise Filter

Each pin of QEI inputs is equipped with a noise filter which can filter the unwanted noise from. The QEA, QEB and IDX noise filters can be disabled through bits NFDIS (QEI_CTL[3]). If enabled, the capture logic is required to sample 4 consecutive same capture input values in order to recognize an edge as a capture event. In Figure 6.20-5, it is a possible implementation of digital noise filter; the interval between pulses requirement for input capture is 4 QEI_CLK clocks width. Any pulse width less than or equal to 3 QEI_CLK clocks will not have any trigger. The timing requirement through Noise Filter is also shown in Figure 6.20-5. CHA, CHB and CHX are the outputs of QEA, QEB and IDX respectively after going through noise filter and polarity control. Refer to Figure 6.20-3. If the noise filter is disabled the input signals QEA, QEB and IDX are passed to the internal signals CHA, CHB and CHX respectively without any delay.

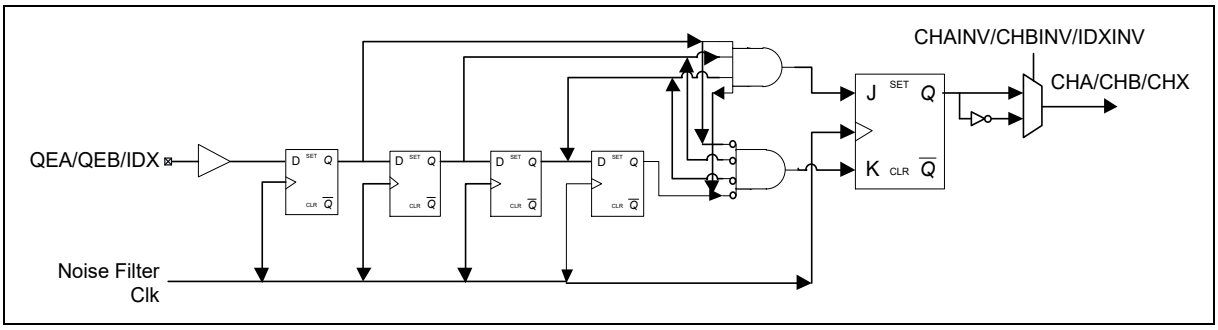


Figure 6.20-3 Noise Filter

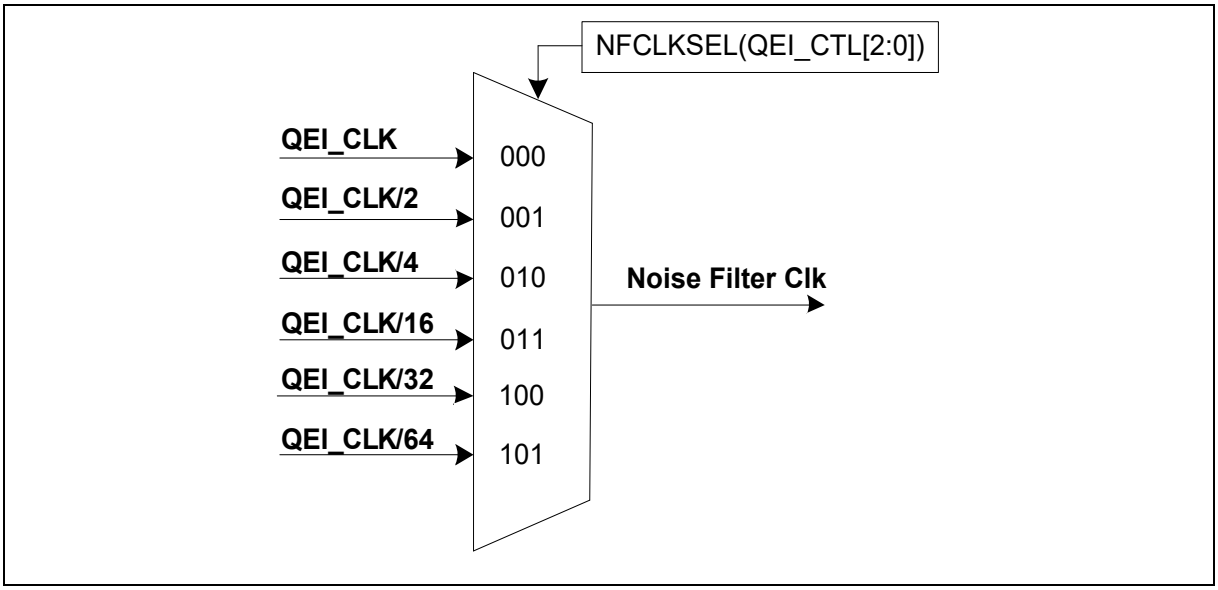


Figure 6.20-4 Noise Filter Sampling Clock Selection

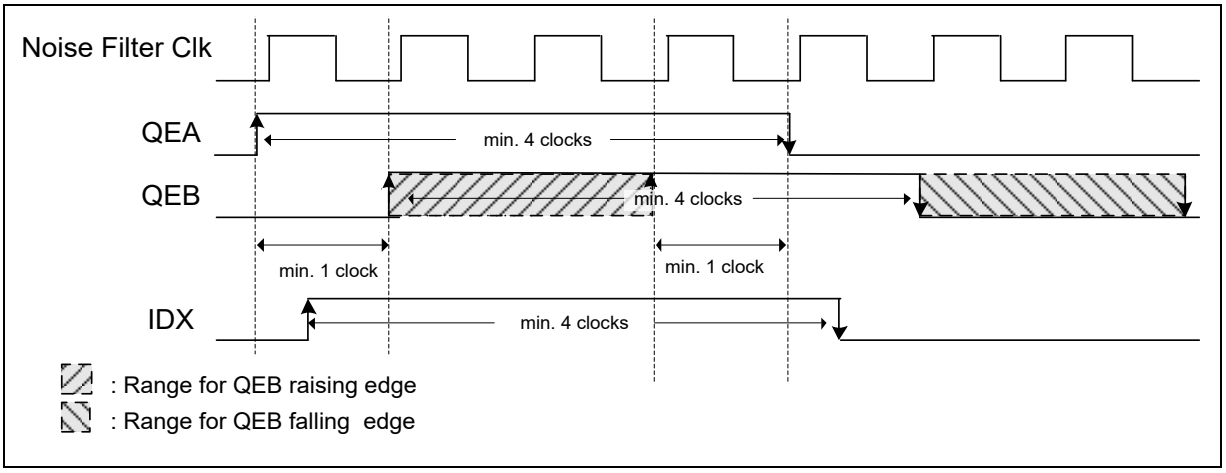


Figure 6.20-5 QEA/QEB/IDX Timing Requirement through Noise Filter

6.20.5.2 Operation of Quadrature Encoder Interface

There are four Quadrature encoder pulse counter operation modes:

- Mode0: x4 free-counting mode
- Mode1: x2 free-counting mode
- Mode2: x4 compare-counting mode
- Mode3: x2 compare-counting mode

6.20.5.3 Free-counting Mode

The quadrature encoder pulse counter CNT(QEI_CNT[31:0]) up or down counts according to the direction indication bit DIRF (QEI_STATUS[8]). When overflow or underflow occurs, it sets flag OVUNF (QEI_STATUS[2]). Refer to Figure 6.20-6 and Figure 6.20-7 for detailed timing.

6.20.5.4 Compare-counting Mode

The pulse counter up or down counts according to the direction indication bit DIRF (QEI_STATUS[8]).

On up counting, flag OVUNF (QEI_STATUS[2]) will be asserted when CNT(QEI_CNT[31:0]) overflows from CNTMAX (QEI_CNTMAX[31:0]) to 0 on the next CHA edge for x2 counting mode, and on CHA/CHB edge for x4 counting mode. On down counting, flag OVUNF (QEI_STATUS[2]) will be asserted when CNT(QEI_CNT[31:0]) underflows from 0 to CNTMAX (QEI_CNTMAX[31:0]) on the next CHA edge for x2 counting mode, and on CHA/CHB edge for x4 counting mode. This mode provides the position of a rotor to user. If a quadrature encoder outputs 1024 pulses to CHA per round, user can write QEI_MAXCNT and CNTCMP(QEI_CNTCMP[31:0]) with 4095 in x4 mode or 2047 in x2 mode and reset CNT(QEI_CNT[31:0]) at initial before compare-counting mode is active. When the CNT(QEI_CNT[31:0]) overflows from CNTCMP(QEI_CNTCMP[31:0]), here CNTCMP(QEI_CNTCMP[31:0]) should be preset the same value as CNTMAX (QEI_CNTMAX[31:0]), it means that rotor runs one round on next CHA/CHB edge. Refer to Figure 6.20-6 and Figure 6.20-7 for detailed timing.

6.20.5.5 X4/X2 Counting Modes

In **X4 Counting mode**, the pulse counter increases or decreases one on every CHA and CHB edge based on the phase relationship of CHA and CHB signals.

QEI x4 Counting mode provides a finer resolution of the rotor position, since the counter increments or decrements more frequently for each QEA/QEB input pulse pair than in QEI x2 mode. This mode is selected by setting the QEI Counting Mode Selection bits MODE(QEI_CTL[9:8]) to 00b or 01b. In this mode, the QEI logic detects every edge on every QEA and QEB input edges.

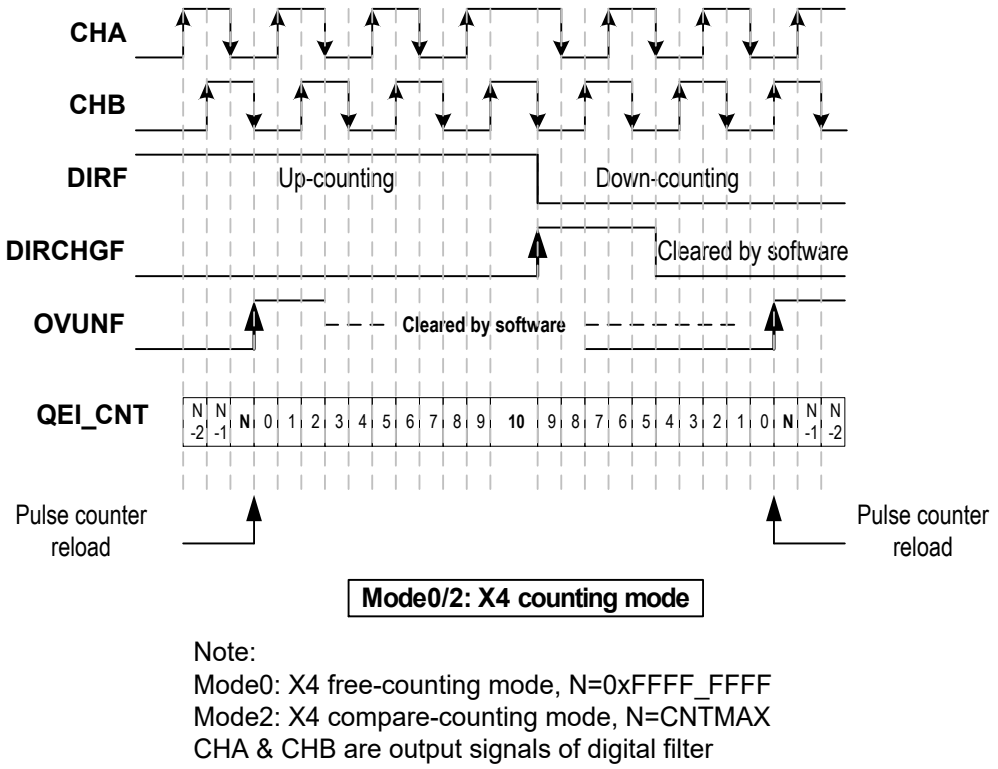
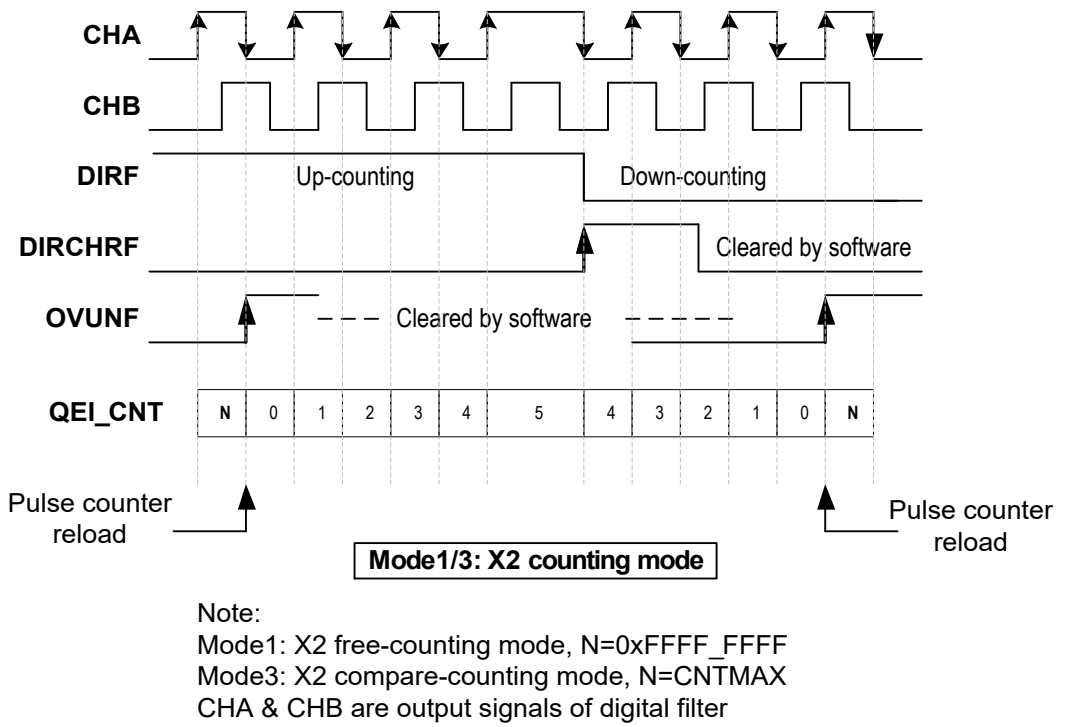


Figure 6.20-6 X4 Counting Mode

In **X2 Counting mode**, the pulse counter increases or decreases one on every CHA edge based on the phase relationship of CHA and CHB signals.

QEI x2 Counting mode is selected by setting the QEI Counting Mode Selection bits (QEI_CTL[9:8]) to 01b or 11b. In this mode, the QEI logic detects every edge on the QEA input only. Every rising and falling edge on the QEA signal clocks the pulse counter.



Note:
Mode1: X2 free-counting mode, N=0xFFFF_FFFF
Mode3: X2 compare-counting mode, N=CNTMAX
CHA & CHB are output signals of digital filter

Figure 6.20-7 X2 Counting Mode

6.20.5.6 Direction of Counting

Refer to Table 6.20-1. If CHA leads CHB, the pulse counter is increased by 1. If CHA lags CHB, the pulse counter is decreased by 1. The QEI control logic generates a signal that sets the DIRF (QEI_STATUS[8]); this in turn determines the direction of the count. When CHA leads CHB, DIRF(QEI_STATUS[8]) is set as 1, and the position counter increments on every active edge. When CHA lags CHB, DIRF(QEI_STATUS[8]) is cleared, and the position counter decrements on every active edge.

Current Signal Detected	**Previous Signal Detected**				**DIR (Counting Direction)**
	Rising		Falling		
	CHA	CHB	CHA	CHB	
CHA rising				√	1 (Increment)
		√			0 (Decrement)
			√		Toggle (direction change)
CHA falling				√	0 (Decrement)
		√			1 (Increment)
	√				Toggle (direction change)
CHB rising	√				1 (Increment)
			√		0 (Decrement)
				√	Toggle (direction change)

CHB falling			√		1 (Increment)
	√				0 (Decrement)
		√			Toggle (direction change)

Table 6.20-1 Direction of Counting

6.20.5.7 Up-Counting

Under the forward direction the DIRF(QEI_STATUS[8]) bit is 1 when up-counting. Software needs to clear the OVUNF (QEI_STATUS[2]) flag. For the free-counting mode the CNT(QEI_CNT[31:0]) counter will count until it matches 0xFFFF_FFFF and next edges on the forward direction will set the bit OVUNF (QEI_STATUS[2]) high and reset CNT(QEI_CNT[31:0]) to 0. For compare-counting mode the CNT(QEI_CNT[31:0]) counter counts until the CNTMAX (QEI_CNTMAX[31:0]) value and next edges on the forward direction will set the bit OVUNF (QEI_STATUS[2]) high and reset CNT(QEI_CNT[31:0]) to 0. Changes of direction trigger a down-count and CNT(QEI_CNT[31:0]) decreasing in counter value. For X2 mode, only CHA edge will set OVUNF (QEI_STATUS[2]) while for X4 mode both CHA and CHB edges will set OVUNF (QEI_STATUS[2]).

6.20.5.8 Down-Counting

A change of direction will cause the counter to down count for X2/X4 counting mode. It is indicated with the DIRF(QEI_STATUS[8]) bit as 0 and DIRCHGF (QEI_STATUS[3]) flag is set to 1. At this stage the CNT(QEI_CNT[31:0]) will start to down-count. In free-counting mode the pulse counter will reload with 0xFFFF_FFFF when it down counts to 0 and sets OVUNF (QEI_STATUS[2]) to high in the next edge. The pulse counter will reload with CNTMAX (QEI_CNTMAX[31:0]) when it down counts to 0 in compare-counting mode and sets OVUNF (QEI_STATUS[2]) to high in the next edge. For X2 mode, only CHA edge will set OVUNF (QEI_STATUS[2]) while for X4 mode both CHA and CHB edges will set OVUNF (QEI_STATUS[2]).

6.20.5.9 Compare Function

The compare function in QEI controller is used to compare the dynamic counting CNT(QEI_CNT[31:0]) with the compare register CNTCMP(QEI_CNTCMP[31:0]). When CNT(QEI_CNT[31:0]) up or down counts and reaches CNTCMP(QEI_CNTCMP[31:0]), the flag CMPF(QEI_STATUS[1]) will be set. Set bit CMP_EN (QEI_CTL[28]) to one to enable the compare function otherwise disable it.

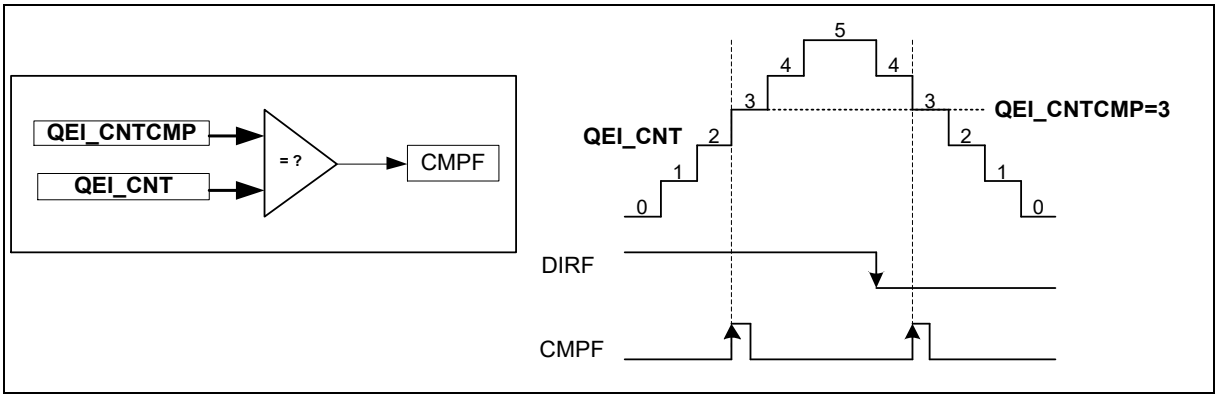


Figure 6.20-8 Compare Operation

6.20.5.10 Reload Counter by Pin IDX

The CNT(QEI_CNT[31:0]) counter can be reset to 0 or reload with the content of CNTMAX (QEI_CNTMAX[31:0]) by the signal CHX (the filtered and polarity-set output of pin IDX) trigger. When the IDX Reload bit IDXRLD_EN(QEI_CTL[27]) is set, a rising edge of CHX causes QEI controller to reset the CNT(QEI_CNT[31:0]) to 0 if the counter is in up-counting; if the counter is in down-counting

the rising edge of CHX causes the QEI controller to reload the CNT(QEI_CNT[31:0]) with the content of CNTMAX (QEI_CNTMAX[31:0]). Refer to Figure 6.20-9 for details.

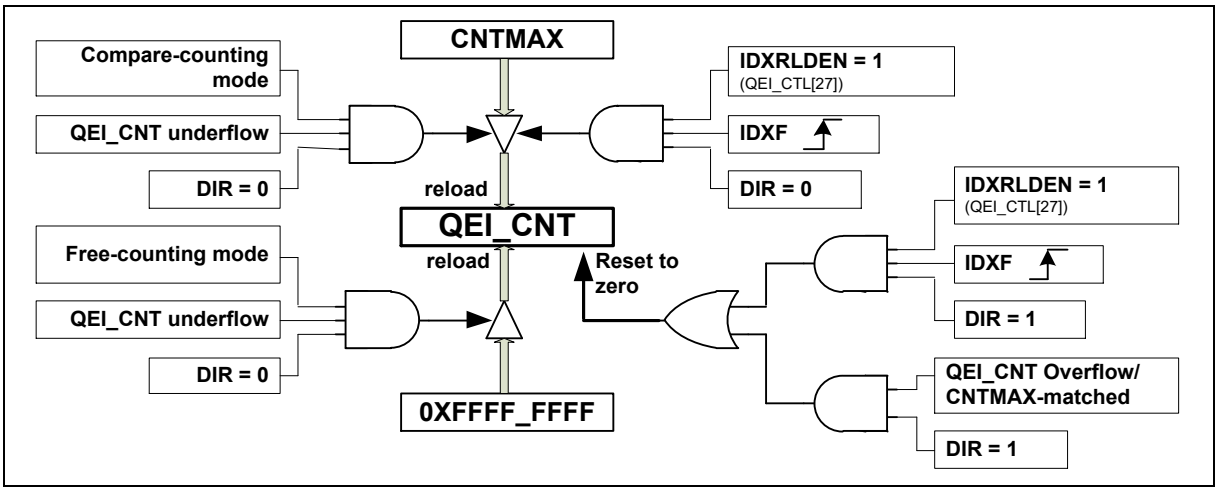


Figure 6.20-9 QEI_CNT Reload/Reset Control

6.20.5.11 Capture QEI Counter

If the bit HOLDCNT(QEI_CTL[24]) is set, the CNT(QEI_CNT[31:0]) content will be captured into QEI Counter Hold Register CNTHOLD(QEI_CNTHOLD[31:0]), the data will be held until the next HOLDCNT (QEI_CTL[24]) trigger comes. The bit HOLDCNT can be set by writing 1 to it or the rising edge of timers interrupt flags TIF (TIMERx_INTSTS[0]).

Note: The bit HOLDCNT is automatically cleared by hardware after CNTHOLD(QEI_CNTHOLD[31:0]) captures the content of QEI counter.

If the bit IDXLATEN (QEI_CTL[25]) is set, the CNT(QEI_CNT[31:0]) content will be latched into QEI Counter Index Latch Register CNTLATCH (QEI_CNTLATCH[31:0]) at every rising edge of CHX signal.

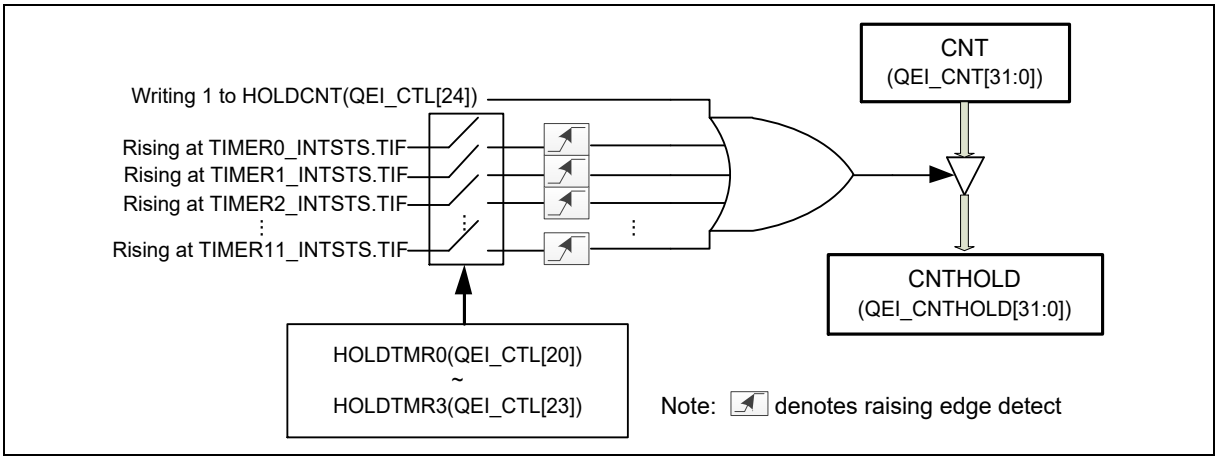
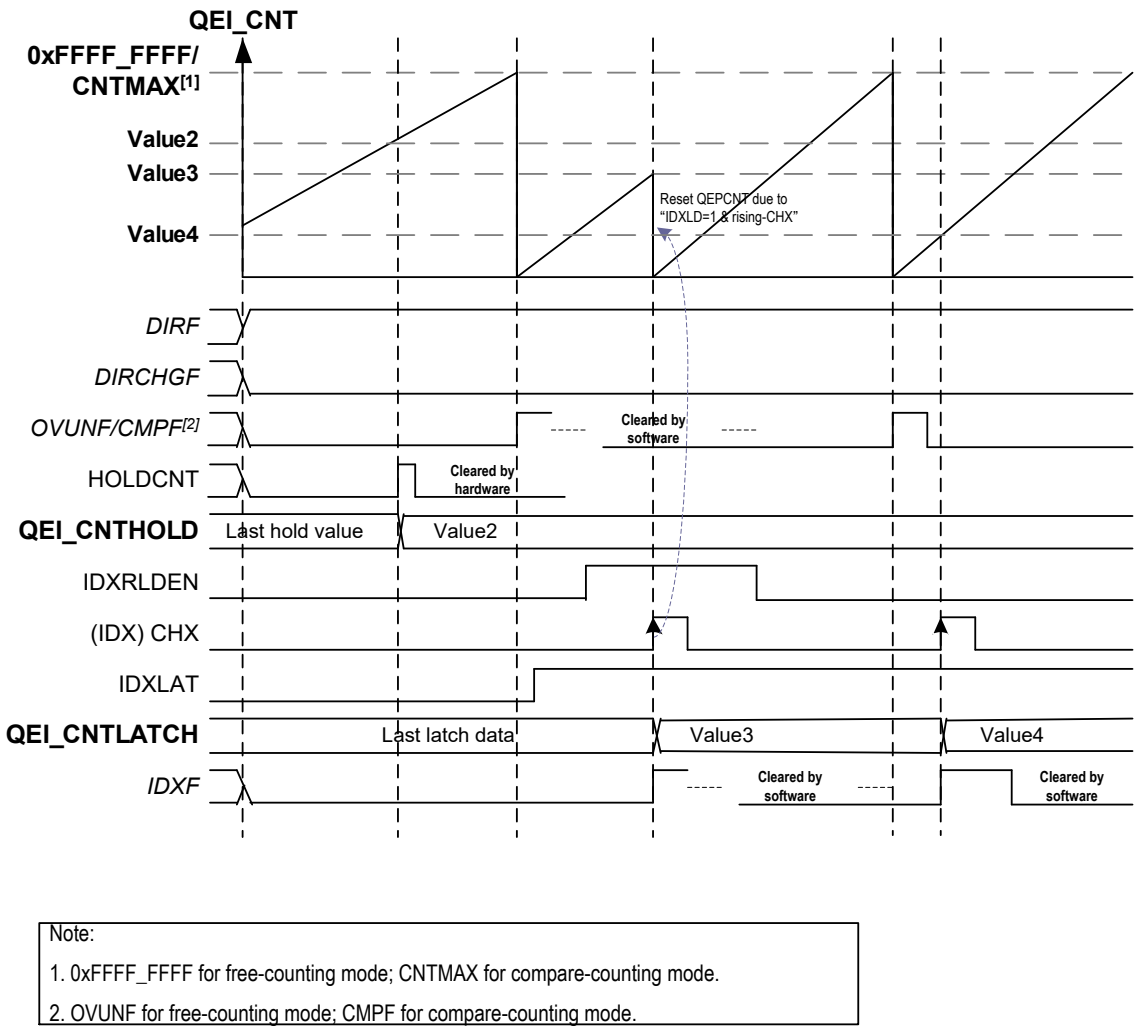


Figure 6.20-10 Trigger Control of Capturing QEI Counter



Note:

1. 0xFFFF_FFFF for free-counting mode; CNTMAX for compare-counting mode.
2. OVUNF for free-counting mode; CMPF for compare-counting mode.

Figure 6.20-11 Capture and Latch QEI Counter

6.20.5.12 QEI Interrupt Architecture

There are four interrupt sources, in which each one has an interrupt flag and enable control bit to trigger QEI Interrupt. When the QEI counter is up-counting and CNT(QEI_CNT[31:0]) overflows or down-counting and underflows, the Overflow/Underflow flag OVUNF (QEI_STATUS[2]) will be set by hardware and it will trigger QEI Interrupt request if bit OVUNIEN (QEI_CTL[16]) is high. When QEI controller detects the encoder rotation change, it toggles the direction indication bit DIRF (QEI_STATUS[8]) and the direction change flag DIRCHGF (QEI_STATUS[3]) will be set by hardware that requests the QEI interrupt if bit DIRIEN (QEI_CTL[17]) is set. When the QEI counter counting value is equal to the value of QEI Counter Compare Register (CNTCMP(QEI_CNTCMP[31:0])), the flag CMPF (QEI_STATUS[1]) will be set by hardware and the QEI Interrupt will be requested if bit CMPIEN (QEI_CTL[18]) is high. When QEI controller detects a rising edge at signal CHX (the filtered and polarity-set output of pin IDX), the flag IDXF(QEI_STATUS[1]) will set by hardware and the QEI interrupt will be requested if bit IDXIEN (QEI_CTL[19]) is set. Note that the four flags, OVUNF(QEI_STATUS[2]), DIRCHGF(QEI_STATUS[3]), CMPF(QEI_STATUS[1]) and IDXF(QEI_STATUS[0]) are set by hardware and must be cleared by software. Figure 6.20-12 demonstrates the architecture of Quadrature Encoder Interface Controller interrupts.

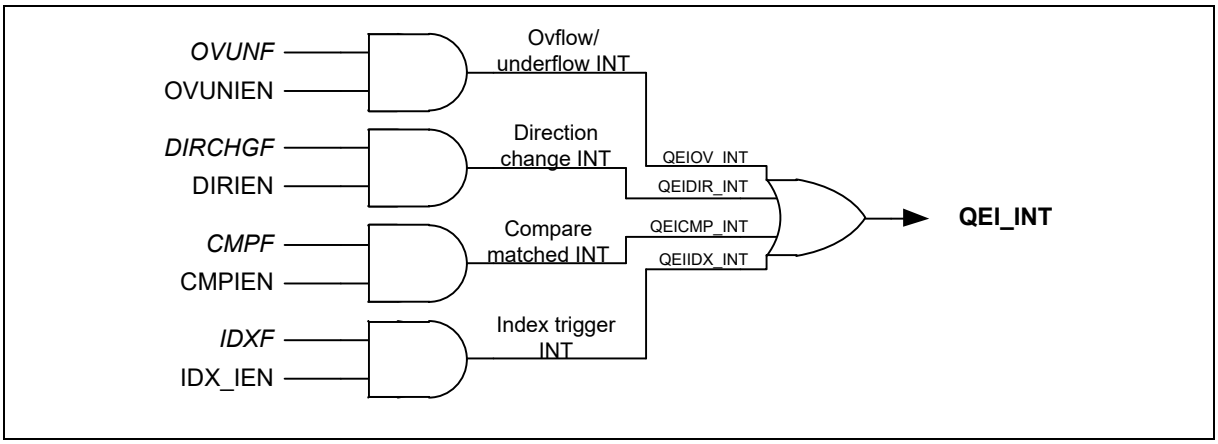


Figure 6.20-12 Quadrature Encoder Interface Interrupt Architecture Diagram

6.20.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
QEI Base Address: **QEI0_BA = 0x40B0_0000** **QEI1_BA = 0x40B1_0000** **QEI2_BA = 0x40B2_0000**				
QEI_CNT **x=0, 1,2**	QEIx_BA+0x00	R/W	QEI Counter Register	0x0000_0000
QEI_CNTHOLD **x=0, 1,2**	QEIx_BA+0x04	R/W	QEI Counter Hold Register	0x0000_0000
QEI_CNTLATCH **x=0,1,2**	QEIx_BA+0x08	R/W	QEI Counter Index Latch Register	0x0000_0000
QEI_CNTCMP **x=0, 1,2**	QEIx_BA+0x0C	R/W	QEI Counter Compare Register	0x0000_0000
QEI_CNTMAX **x=0, 1,2**	QEIx_BA+0x14	R/W	QEI Pre-set Maximum Count Register	0x0000_0000
QEI_CTL **x=0, 1,2**	QEIx_BA+0x18	R/W	QEI Controller Control Register	0x0000_0000
QEI_CTL1 **x=0, 1,2**	QEIx_BA+0x1C	R/W	QEI Controller Control Register1	0x0000_0000
QEI_STATUS **x=0, 1,2**	QEIx_BA+0x2C	R/W	QEI Controller Status Register	0x0000_0000

6.20.7 Register Description

QEI Counter Register (QEI_CNT)

Register	Offset	R/W	Description	Reset Value
QEI_CNT	QEIx_BA+0x00	R/W	QEI Counter Register	0x0000_0000

31	30	29	28	27	26	25	24
CNT							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CNT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CNT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNT							

Bits	Description	
[31:0]	**CNT**	**Quadrature Encoder Interface Counter** A 32-bit up/down counter. When an effective phase pulse is detected, this counter is increased by one if the bit DIRF (QEI_STATUS[8]) is 1 or decreased by one if the bit DIRF(QEI_STATUS[8]) is 0. This register performs an integrator that counts the value proportional to the encoder position. The pulse counter may be initialized to a predetermined value by one of three events occurs: ● Software is written if QEIEN (QEI_CTL[29]) = 0. ● Compare-match event if QEIEN(QEI_CTL[29])=1 and QEI is in compare-counting mode. ● Index signal change if QEIEN(QEI_CTL[29])=1 and IDXRLDEN (QEI_CTL[27])=1.

QEI Counter Hold Register (QEI_CNTHOLD)

Register	Offset	R/W	Description	Reset Value
QEI_CNTHOLD	QEIx_BA+0x04	R/W	QEI Counter Hold Register	0x0000_0000

31	30	29	28	27	26	25	24
CNTHOLD							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CNTHOLD							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CNTHOLD							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNTHOLD							

Bits	Description	
[31:0]	**CNTHOLD**	**Quadrature Encoder Interface Counter Hold** When the bit HOLDCNT (QEI_CTL[24]) goes from low to high, the CNT(QEI_CNT[31:0]) is copied into CNTHOLD (QEI_CNTHOLD[31:0]) register.

QEI Counter Index Latch Register (QEI_CNTLATCH)

Register	Offset	R/W	Description	Reset Value
QEI_CNTLATCH	QEIx_BA+0x08	R/W	QEI Counter Index Latch Register	0x0000_0000

31	30	29	28	27	26	25	24
CNTLATCH							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CNTLATCH							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CNTLATCH							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNTLATCH							

Bits	Description	
[31:0]	**CNTLATCH**	**Quadrature Encoder Interface Counter Index Latch** When the IDXF (QEI_STATUS[0]) bit is set, the CNT(QEI_CNT[31:0]) is copied into CNTLATCH (QEI_CNTLATCH[31:0]) register.

QEI Counter Compare Register (QEI_CNTCMP)

Register	Offset	R/W	Description	Reset Value
QEI_CNTCMP	QEIx_BA+0x0C	R/W	QEI Counter Compare Register	0x0000_0000

31	30	29	28	27	26	25	24
CNTCMP							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CNTCMP							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CNTCMP							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNTCMP							

Bits	Description	
[31:0]	**CNTCMP**	**Quadrature Encoder Interface Counter Compare** If the QEI controller is in the compare-counting mode CMPEN (QEI_CTL[28]) =1, when the value of CNT(QEI_CNT[31:0]) matches CNTCMP(QEI_CNTCMP[31:0]), CMPF will be set. This register is software writable.

QEI Pre-set Maximum Count Register (QEI_CNTMAX)

Register	Offset	R/W	Description	Reset Value
QEI_CNTMAX	QEIx_BA+0x14	R/W	QEI Pre-set Maximum Count Register	0x0000_0000

31	30	29	28	27	26	25	24
CNTMAX							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CNTMAX							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CNTMAX							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNTMAX							

Bits	Description	
[31:0]	**CNTMAX**	**Quadrature Encoder Interface Preset Maximum Count** This register value determined by user stores the maximum value which may be the number of the QEI counter for the QEI controller compare-counting mode.

QEI Controller Control Register (QEI_CTL)

Register	Offset	R/W	Description	Reset Value
QEI_CTL	QEIx_BA+0x18	R/W	QEI Controller Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		QEIEN	CMPEN	IDXRLDEN	Reserved	IDXLATEN	HOLDCNT
23	22	21	20	19	18	17	16
HOLDTMR3	HOLDTMR2	HOLDTMR1	HOLDTMR0	IDXIEN	CMPIEN	DIRIEN	OVUNIEN
15	14	13	12	11	10	9	8
Reserved	IDXINV	CHBINV	CHAINV	Reserved		MODE	
7	6	5	4	3	2	1	0
Reserved	IDXEN	CHBEN	CHAEN	NFDIS	NFCLKSEL		

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29]	**QEIEN**	**Quadrature Encoder Interface Controller Enable Bit** 0 = QEI controller function Disabled. 1 = QEI controller function Enabled.
[28]	**CMPEN**	**The Compare Function Enable Bit** The compare function in QEI controller is to compare the dynamic counting QEI_CNT with the compare register CNTCMP(QEI_CNTCMP[31:0]), if CNT(QEI_CNT[31:0]) reaches CNTCMP(QEI_CNTCMP[31:0]), the flag CMPF will be set. 0 = Compare function Disabled. 1 = Compare function Enabled.
[27]	**IDXRLDEN**	**Index Trigger QEI_CNT Reload Enable Bit** When this bit is high and a rising edge comes on signal CHX, the CNT(QEI_CNT[31:0]) will be reset to 0 if the counter is in up-counting type (DIRF(QEI_STATUS[8]) = 1); while the CNT(QEI_CNT[31:0]) will be reloaded with CNTMAX (QEI_CNTMAX[31:0]) content if the counter is in down-counting type (DIRF(QEI_STATUS[8]) = 0). 0 = Reload function Disabled. 1 = QEI_CNT re-initialized by Index signal Enabled.
[26]	**Reserved**	Reserved.
[25]	**IDXLATEN**	**Index Latch QEI_CNT Enable Bit** If this bit is set to high, the CNT(QEI_CNT[31:0]) content will be latched into CNTLATCH (QEI_CNTLATCH[31:0]) at every rising on signal CHX. 0 = The index signal latch QEI counter function Disabled. 1 = The index signal latch QEI counter function Enabled.

[24]	**HOLDCNT**	**Hold QEI_CNT Control** When this bit is set from low to high, the CNT(QEI_CNT[31:0]) is copied into CNTHOLD(QEI_CNTHOLD[31:0]). This bit may be set by writing 1 to it or Timer0~Timer11 interrupt flag TIF (TIMERx_INTSTS[0]). 0 = No operation. 1 = QEI_CNT content is captured and stored in CNTHOLD(QEI_CNTHOLD[31:0]). **Note:** This bit is automatically cleared after QEI_CNTHOLD holds QEI_CNT value.
[23]	**HOLDTMR3**	**Hold QEI_CNT by Timer 3** 0 = TIF (TIMER3_INTSTS[0]) has no effect on HOLDCNT. 1 = A rising edge of bit TIF(TIMER3_INTSTS[0]) in timer 3 sets HOLDCNT to 1.
[22]	**HOLDTMR2**	**Hold QEI_CNT by Timer 2** 0 = TIF(TIMER2_INTSTS[0]) has no effect on HOLDCNT. 1 = A rising edge of bit TIF(TIMER2_INTSTS[0]) in timer 2 sets HOLDCNT to 1.
[21]	**HOLDTMR1**	**Hold QEI_CNT by Timer 1** 0 = TIF(TIMER1_INTSTS[0]) has no effect on HOLDCNT. 1 = A rising edge of bit TIF (TIMER1_INTSTS[0]) in timer 1 sets HOLDCNT to 1.
[20]	**HOLDTMR0**	**Hold QEI_CNT by Timer 0** 0 = TIF (TIMER0_INTSTS[0]) has no effect on HOLDCNT. 1 = A rising edge of bit TIF(TIMER0_INTSTS[0]) in timer 0 sets HOLDCNT to 1.
[19]	**IDXIEN**	**IDXF Trigger QEI Interrupt Enable Bit** 0 = The IDXF can trigger QEI interrupt Disabled. 1 = The IDXF can trigger QEI interrupt Enabled.
[18]	**CMPIEN**	**CMPF Trigger QEI Interrupt Enable Bit** 0 = CMPF can trigger QEI controller interrupt Disabled. 1 = CMPF can trigger QEI controller interrupt Enabled.
[17]	**DIRIEN**	**DIRCHGF Trigger QEI Interrupt Enable Bit** 0 = DIRCHGF can trigger QEI controller interrupt Disabled. 1 = DIRCHGF can trigger QEI controller interrupt Enabled.
[16]	**OVUNIEN**	**OVUNF Trigger QEI Interrupt Enable Bit** 0 = OVUNF can trigger QEI controller interrupt Disabled. 1 = OVUNF can trigger QEI controller interrupt Enabled.
[15]	**Reserved**	Reserved.
[14]	**IDXINV**	**Inverse IDX Input Polarity** 0 = Not inverse IDX input polarity. 1 = IDX input polarity is inversed to QEI controller.
[13]	**CHBINV**	**Inverse QEB Input Polarity** 0 = Not inverse QEB input polarity. 1 = QEB input polarity is inversed to QEI controller.
[12]	**CHAINV**	**Inverse QEA Input Polarity** 0 = Not inverse QEA input polarity. 1 = QEA input polarity is inversed to QEI controller.
[11:10]	**Reserved**	Reserved.

[9:8]	**MODE**	**QEI Counting Mode Selection** There are four quadrature encoder pulse counter operation modes. 00 = X4 Free-counting Mode. 01 = X2 Free-counting Mode. 10 = X4 Compare-counting Mode. 11 = X2 Compare-counting Mode.
[7]	**Reserved**	Reserved.
[6]	**IDXEN**	**IDX Input to QEI Controller Enable Bit** 0 = IDX input to QEI Controller Disabled. 1 = IDX input to QEI Controller Enabled.
[5]	**CHBEN**	**QEB Input to QEI Controller Enable Bit** 0 = QEB input to QEI Controller Disabled. 1 = QEB input to QEI Controller Enabled.
[4]	**CHAEN**	**QEA Input to QEI Controller Enable Bit** 0 = QEA input to QEI Controller Disabled. 1 = QEA input to QEI Controller Enabled.
[3]	**NFDIS**	**QEI Controller Input Noise Filter Disable Bit** 0 = The noise filter of QEI controller Enabled. 1 = The noise filter of QEI controller Disabled.
[2:0]	**NFCLKSEL**	**Noise Filter Clock Pre-divide Selection** To determine the sampling frequency of the Noise Filter clock. 000 = QEI_CLK. 001 = QEI_CLK/2. 010 = QEI_CLK/4. 011 = QEI_CLK/16. 100 = QEI_CLK/32. 101 = QEI_CLK/64.

QEI Controller Control Register1 (QEI_CTL1)

Register	Offset	R/W	Description	Reset Value
QEI_CTL1	QEIx_BA+0x1C	R/W	QEI Controller Control Register1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
HOLDTMR11	**HOLDTMR10**	**HOLDTMR9**	**HOLDTMR8**	**HOLDTMR7**	**HOLDTMR6**	**HOLDTMR5**	**HOLDTMR4**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**HOLDTMR11**	**Hold QEI_CNT by Timer 11** 0 = TIF (TIMER11_INTSTS[0]) has no effect on HOLDCNT. 1 = A rising edge of bit TIF(TIMER11_INTSTS[0]) in timer 11 sets HOLDCNT to 1.
[6]	**HOLDTMR10**	**Hold QEI_CNT by Timer 10** 0 = TIF (TIMER10_INTSTS[0]) has no effect on HOLDCNT. 1 = A rising edge of bit TIF(TIMER10_INTSTS[0]) in timer 10 sets HOLDCNT to 1.
[5]	**HOLDTMR9**	**Hold QEI_CNT by Timer 9** 0 = TIF (TIMER9_INTSTS[0]) has no effect on HOLDCNT. 1 = A rising edge of bit TIF(TIMER9_INTSTS[0]) in timer 9 sets HOLDCNT to 1.
[4]	**HOLDTMR8**	**Hold QEI_CNT by Timer 8** 0 = TIF (TIMER8_INTSTS[0]) has no effect on HOLDCNT. 1 = A rising edge of bit TIF(TIMER8_INTSTS[0]) in timer 8 sets HOLDCNT to 1.
[3]	**HOLDTMR7**	**Hold QEI_CNT by Timer 7** 0 = TIF (TIMER7_INTSTS[0]) has no effect on HOLDCNT. 1 = A rising edge of bit TIF(TIMER7_INTSTS[0]) in timer 7 sets HOLDCNT to 1.
[2]	**HOLDTMR6**	**Hold QEI_CNT by Timer 6** 0 = TIF (TIMER6_INTSTS[0]) has no effect on HOLDCNT. 1 = A rising edge of bit TIF(TIMER6_INTSTS[0]) in timer 6 sets HOLDCNT to 1.
[1]	**HOLDTMR5**	**Hold QEI_CNT by Timer 5** 0 = TIF (TIMER5_INTSTS[0]) has no effect on HOLDCNT. 1 = A rising edge of bit TIF(TIMER5_INTSTS[0]) in timer 5 sets HOLDCNT to 1.
[0]	**HOLDTMR4**	**Hold QEI_CNT by Timer 4** 0 = TIF (TIMER4_INTSTS[0]) has no effect on HOLDCNT. 1 = A rising edge of bit TIF(TIMER4_INTSTS[0]) in timer 4 sets HOLDCNT to 1.

QEI Controller Status Register (QEI_STATUS)

Register	Offset	R/W	Description	Reset Value
QEI_STATUS	QEIx_BA+0x2C	R/W	QEI Controller Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							DIRF
7	6	5	4	3	2	1	0
Reserved				DIRCHGF	OVUNF	CMPF	IDXF

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8]	**DIRF**	**QEI Counter Counting Direction Indication** 0 = QEI Counter is in down-counting. 1 = QEI Counter is in up-counting. **Note:** This bit is set/reset by hardware according to the phase detection between CHA and CHB.
[7:4]	**Reserved**	Reserved.
[3]	**DIRCHGF**	**Direction Change Flag** Flag is set by hardware while QEI counter counting direction is changed. Software can clear this bit by writing 1 to it. 0 = No change in QEI counter counting direction. 1 = QEI counter counting direction is changed. **Note:** This bit is only cleared by writing 1 to it.
[2]	**OVUNF**	**QEI Counter Overflow or Underflow Flag** Flag is set by hardware while CNT(QEI_CNT[31:0]) overflows from 0xFFFF_FFFF to 0 in free-counting mode or from the CNTMAX (QEI_CNTMAX[31:0]) to 0 in compare-counting mode. Similarly, the flag is set while QEI counter underflows from 0 to 0xFFFF_FFFF or CNTMAX (QEI_CNTMAX[31:0]). 0 = No overflow or underflow occurs in QEI counter. 1 = QEI counter occurs counting overflow or underflow. **Note:** This bit is only cleared by writing 1 to it.
[1]	**CMPF**	**Compare-match Flag** If the QEI compare function is enabled, the flag is set by hardware while QEI counter up or down counts and reach to the CNTCMP(QEI_CNTCMP[31:0]). 0 = QEI counter does not match with CNTCMP(QEI_CNTCMP[31:0]). 1 = QEI counter counts to the same as CNTCMP(QEI_CNTCMP[31:0]). **Note:** This bit is only cleared by writing 1 to it.

[0]	**IDXF**	**IDX Detected Flag** When the QEI controller detects a rising edge on signal CHX it will set flag IDXF to high. 0 = No rising edge detected on signal CHX. 1 = A rising edge occurs on signal CHX. **Note:** This bit is only cleared by writing 1 to it.

6.21 Watchdog Timer (WDT)

6.21.1 Overview

The Watchdog Timer (WDT) is used to perform a system reset when system runs into an unknown state. This prevents system from hanging for an infinite period of time. Besides, this Watchdog Timer supports the function to wake up system from Idle/Power-down mode.

6.21.2 Features

- Three Watchdogs, one for TrustZone Secure (TZS), one for TrustZone Secure/Non-Secure (TZS/TZNS) and the other for SubM
- 20-bit free running up counter for WDT time-out interval.
- Selectable time-out interval (2^4 ~ 2^{20}) and the time-out interval is 0.5 ms ~ 32.768 s if WDT_CLK = 32 kHz.
- System kept in reset state for a period of (1 / WDT_CLK) * 63
- Supports selectable WDT reset delay period, including 1026, 130, 18 or 3 WDT_CLK reset delay period.
- Supports to force WDT1 enabled after chip powered on or reset by setting Power-on Setting [2]
- Supports WDT time-out wake-up function.
- WDT0 can reset both real time Cortex-A35 and real time Cortex-M4 sub-systems.
- WDT1 can reset real time Cortex-A35 sub-system when WDT1RSTAEN (SYS_MISCRFCR[16]) is set.
- WDT1 can reset real time Cortex-M4 sub-system when WDT1RSTMEN (SYS_MISCRFCR[18]) is set.
- WDT2 can reset Cortex-M4 sub-system.
- WDT2 can reset Cortex-A35 sub-system when WDT2RSTAEN(SYS_MISCRFCR[17]) is set.

6.21.3 Block Diagram

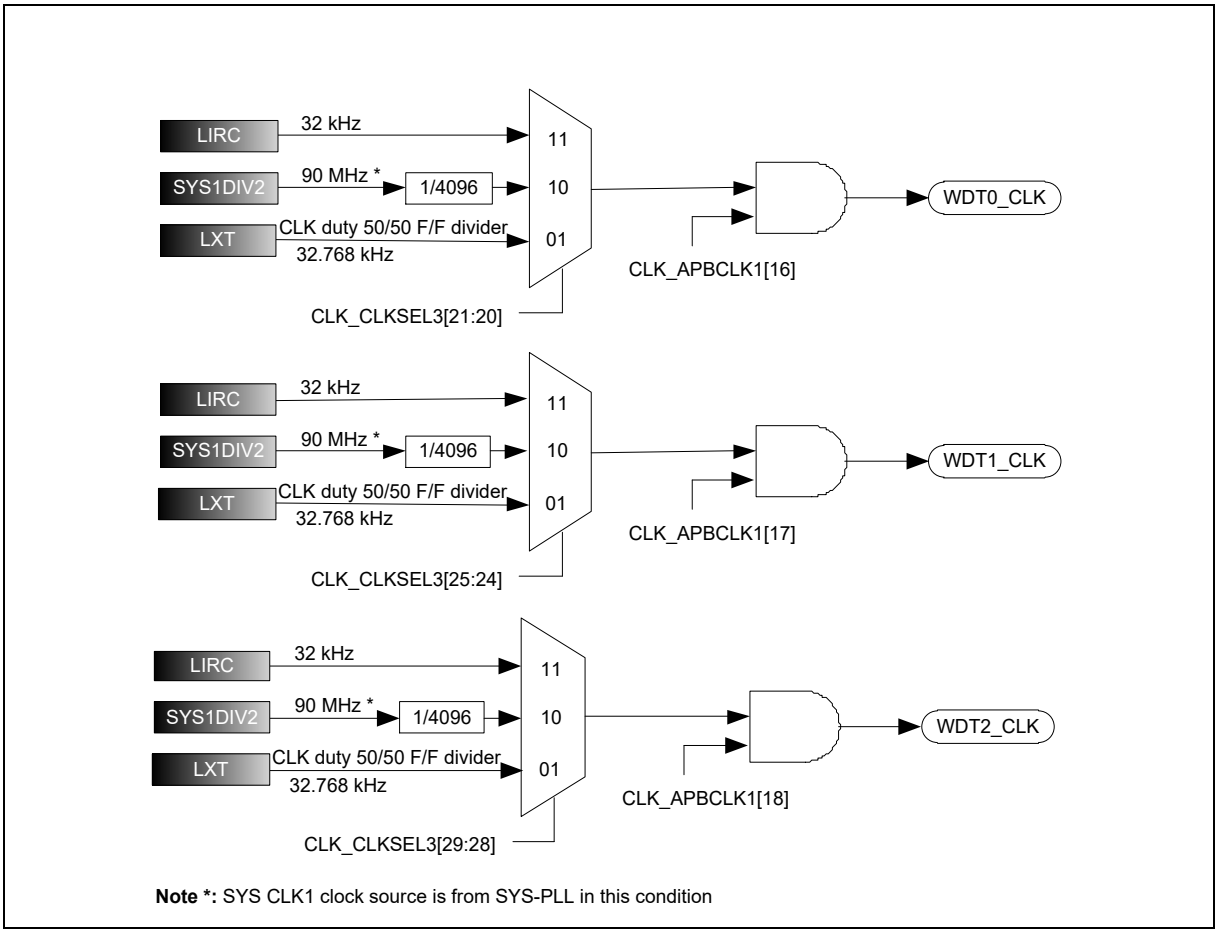


Figure 6.21-1 Watchdog Timer Clock Control

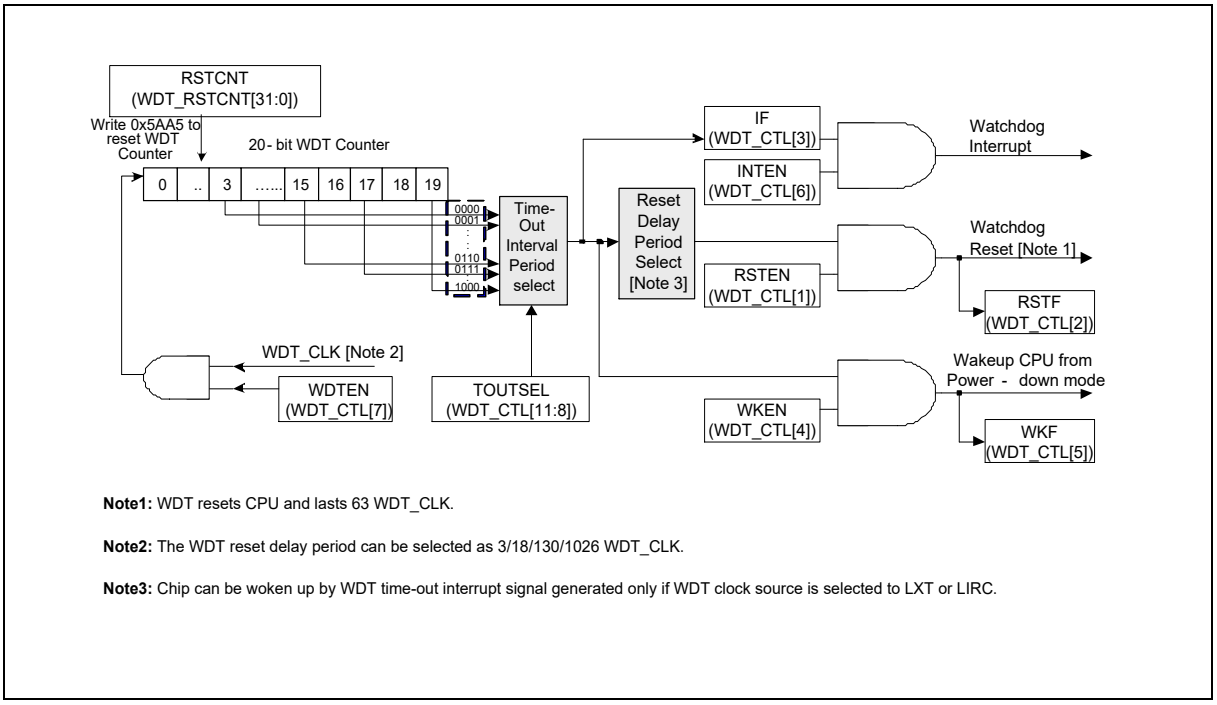


Figure 6.21-2 Watchdog Timer Block Diagram

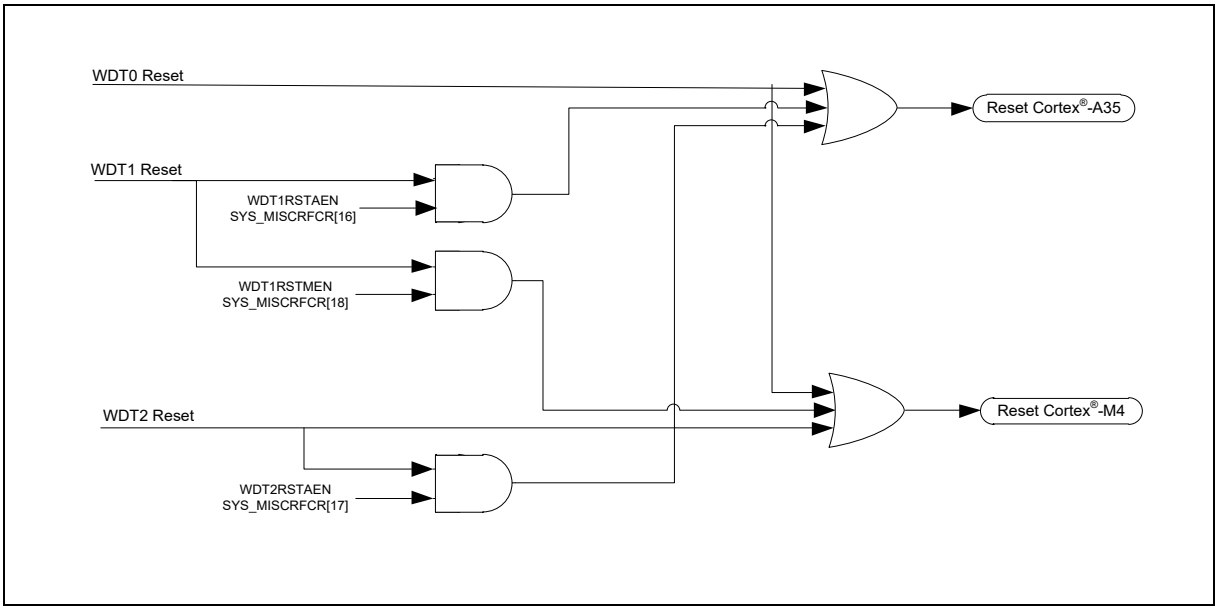


Figure 6.21-3 Watchdog Timer Reset Control Block Diagram

6.21.4 Basic Configuration

- Clock Source Configuration
  - Select the source of WDT0 peripheral clock on WDT0SEL (CLK_CLKSEL3[21:20])
  - Enable WDT0 peripheral clock in WDT0CKEN (CLK_APBCLK1[16]).
  - Select the source of WDT1 peripheral clock on WDT0SEL (CLK_CLKSEL3[25:24])
  - Enable WDT1 peripheral clock in WDT1CKEN (CLK_APBCLK1[17]).
  - Select the source of WDT2 peripheral clock on WDT0SEL (CLK_CLKSEL3[29:28])
  - Enable WDT2 peripheral clock in WDT2CKEN (CLK_APBCLK1[18]).
- Force enable and active WDT1 controller after chip powered on or reset by setting Power-on Setting[2].

6.21.5 Functional Description

The WDT includes an 20-bit free running up counter with programmable time-out intervals. Table 6.21-1 shows the WDT time-out interval period selection and Figure 6.21-1 shows the WDT time-out interval and reset period timing.

6.21.5.1 WDT Time-out Interrupt

Setting WDTEN (WDT_CTL[7]) to 1 will enable the WDT function and the WDT counter to start counting up. The SYNC (WDT_CTL[30]) can be indicated whether enable/disable WDTEN function is completed or not. There are nine time-out interval period can be selected by setting TOUTSEL (WDT_CTL[11:8]). When the WDT up counter reaches the TOUTSEL (WDT_CTL[11:8])settings, WDT time-out interrupt will occur then WDT time-out interrupt flag IF (WDT_CTL[3]) will be set to 1 immediately. If INTEN (WDT_CTL[6]) is enabled, WDT time-out interrupt will inform CPU.

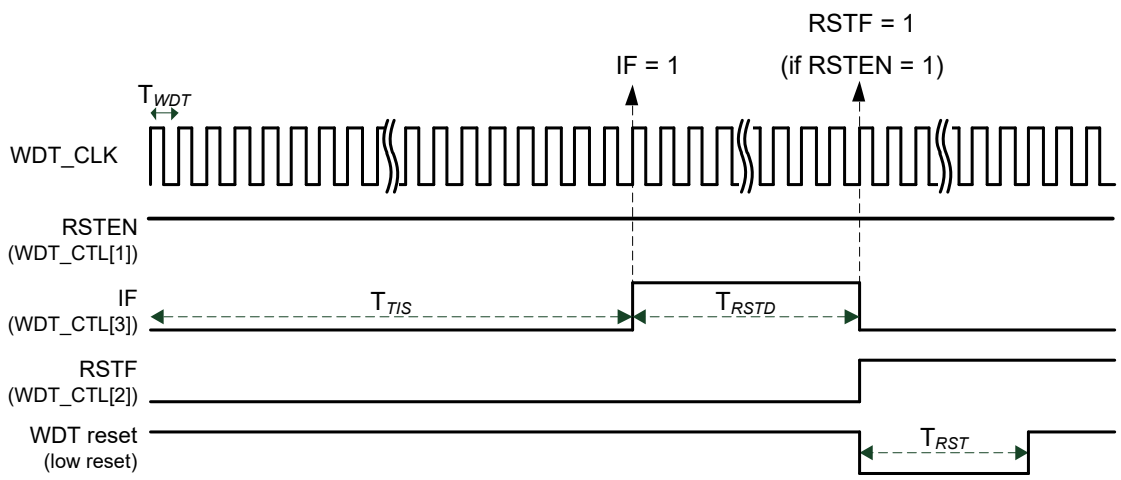
6.21.5.2 WDT Reset Delay Period and Reset System

There is a specified T_{RSTD} reset delay period follows the IF (WDT_CTL[3]) is set to 1. User should program 0x00005AA5 to RSTCNT (WDT_RSTCNT[31:0]) to reset the 20-bit WDT up counter value to avoid generate WDT time-out reset signal before the T_{RSTD} reset delay period expires. Moreover, user

should set RSTDSEL (WDT_ALTCTL [1:0]) to select reset delay period to clear WDT counter. If the WDT up counter value has not been cleared after the specific T_{RSTD} delay period expires, the WDT control will set RSTF (WDT_CTL[2]) to 1 if RSTEN (WDT_CTL[1]) bit is enabled, then chip enters reset state immediately. T_{RST} reset period will keep last 63 WDT clocks then chip restarts executing program from reset vector (0x0000_0000). The RSTF (WDT_CTL[2]) will keep 1 after WDT time-out resets the chip, and user can check RSTF (WDT_CTL[2]) by software to recognize the system has been reset by WDT time-out reset or not.

TOUTSEL	Time-Out Interval Period T_{TIS}	Reset Delay Period T_{RSTD}
0000	$2^4 * T_{WDT}$	(3/18/130/1026) * T_{WDT}
0001	$2^6 * T_{WDT}$	(3/18/130/1026) * T_{WDT}
0010	$2^8 * T_{WDT}$	(3/18/130/1026) * T_{WDT}
0011	$2^{10} * T_{WDT}$	(3/18/130/1026) * T_{WDT}
0100	$2^{12} * T_{WDT}$	(3/18/130/1026) * T_{WDT}
0101	$2^{14} * T_{WDT}$	(3/18/130/1026) * T_{WDT}
0110	$2^{16} * T_{WDT}$	(3/18/130/1026) * T_{WDT}
0111	$2^{18} * T_{WDT}$	(3/18/130/1026) * T_{WDT}
1000	$2^{20} * T_{WDT}$	(3/18/130/1026) * T_{WDT}

Table 6.21-1 Watchdog Timer Time-out Interval Period Selection



- T_{WDT} : *Watchdog Clock Time Period*
- T_{TIS} : *Watchdog Time-out Interval Period ((2^4 ~ 2^{20}) * T_{WDT})*
- T_{RSTD} : *Watchdog Reset Delay Period*
  *- Selectable 3/18/130/1026 * T_{WDT} delay period controlled by RSTDSEL(WDT_ALTCTL [1:0])*
- T_{RST} : *Watchdog Reset Period (63 * T_{WDT})*

Figure 6.21-1 Watchdog Timer Time-out Interval and Reset Period Timing

6.21.5.3 WDT Wake-up

System can be woken up from Power-down mode while WDT time-out interrupt signal is generated and WKEN (WDT_CTL[4]) enabled. In the meanwhile, the WKF (WDT_CTL[5]) will be set to 1 automatically, and user can check WKF (WDT_CTL[5]) status by software to recognize the system has been woken up by WDT time-out interrupt or not. WDT can wakeup system no matter which clock source is selected when SYSPLLAPD(CLK_PWRCTL[11]) is disabled. If SYSPLLAPD(CLK_PWRCTL[11]) is be set, WDT can't wakeup system when clock source is selected SYS1DIV2.

6.21.5.4 WDT ICE Debug

When ICE is connected to MCU, WDT counter is counting or not by ICEDEBUG (WDT_CTL[31]). The default value of ICEDEBUG is 0, WDT counter will stop counting when CPU is held by ICE. If ICEDEBUG is set to 1, the WDT counter will keep counting no matter CPU is held by ICE or not.

6.21.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
WDT Base Address: **WDT0_BA = 0x4040_0000** **WDT1_BA = 0x4044_0000** **WDT2_BA = 0x4098_0000**				
WDT_CTL **n= 0,1,2**	WDTn_BA+0x00	R/W	WDT Control Register	0x0000_0800
WDT_ALTCTL **n= 0,1,2**	WDTn_BA+0x04	R/W	WDT Alternative Control Register	0x0000_0000
WDT_RSTCNT **n= 0,1,2**	WDTn_BA+0x08	W	WDT Reset Counter Register	0x0000_0000

6.21.7 Register Description

WDT Control Register (WDT_CTL)

Register	Offset	R/W	Description	Reset Value
WDT_CTL	WDTn_BA+0x00	R/W	WDT Control Register	0x0000_0800

31	30	29	28	27	26	25	24
ICEDEBUG	**SYNC**	**Reserved**					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**TOUTSEL**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WDTEN	**INTEN**	**WKF**	**WKEN**	**IF**	**RSTF**	**RSTEN**	**Reserved**

Bits	Description	
[31]	**ICEDEBUG**	**ICE Debug Mode Acknowledge Disable Bit (Write Protect)** 0 = ICE debug mode acknowledgement affects WDT counting. WDT up counter will be held while CPU is held by ICE. 1 = ICE debug mode acknowledgement Disabled. WDT up counter will keep going no matter CPU is held by ICE or not. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register
[30]	**SYNC**	**WDT Enable Control SYNC Flag Indicator (Read Only)** If user executes enable/disable WDTEN (WDT_CTL[7]), this flag can be indicated enable/disable WDTEN function is completed or not. 0 = Set WDTEN bit is completed. 1 = Set WDTEN bit is synchronizing and not become active yet. **Note:** Performing enable or disable WDTEN bit needs 2 * WDT_CLK period to become active.
[29:12]	**Reserved**	Reserved.
[11:8]	**TOUTSEL**	**WDT Time-out Interval Selection (Write Protect)** These three bits select the time-out interval period for the WDT. 0000 = 2^4 * WDT_CLK. 0001 = 2^6 * WDT_CLK. 0010 = 2^8 * WDT_CLK. 0011 = 2^{10} * WDT_CLK. 0100= 2^{12} * WDT_CLK. 0101 = 2^{14} * WDT_CLK. 0110 = 2^{16} * WDT_CLK. 0111 = 2^{18} * WDT_CLK. 1000 = 2^{20} * WDT_CLK. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register.

[7]	**WDTEN**	**WDT Enable Bit (Write Protect)** 0 = WDT Disabled (This action will reset the internal up counter value). 1 = WDT Enabled. **Note 1:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register. **Note 2:** If CWDTEN[2:0] (combined by Config0[31] and Config0[4:3]) bits is not configured to 111, this bit is forced as 1 and user cannot change this bit to 0.
[6]	**INTEN**	**WDT Time-out Interrupt Enable Bit (Write Protect)** If this bit is enabled, the WDT time-out interrupt signal is generated and inform to CPU. 0 = WDT time-out interrupt Disabled. 1 = WDT time-out interrupt Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register.
[5]	**WKF**	**WDT Time-out Wake-up Flag** This bit indicates the interrupt wake-up flag status of WDT 0 = WDT does not cause chip wake-up. 1 = Chip wake-up from Idle or Power-down mode if WDT time-out interrupt signal generated. **Note:** This bit is cleared by writing 1 to it.
[4]	**WKEN**	**WDT Time-out Wake-up Function Control (Write Protect)** If this bit is set to 1, while WDT time-out interrupt flag IF (WDT_CTL[3]) is generated to 1 and interrupt enable bit INTEN (WDT_CTL[6]) is enabled, the WDT time-out interrupt signal will generate a wake-up trigger event to chip. 0 = Wake-up trigger event Disabled if WDT time-out interrupt signal generated. 1 = Wake-up trigger event Enabled if WDT time-out interrupt signal generated. **Note 1:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register. **Note 2:** Chip can be woken up by WDT time-out interrupt signal generated only if WDT clock source is selected to 32 kHz internal low speed RC oscillator (LIRC) or LXT.
[3]	**IF**	**WDT Time-out Interrupt Flag** This bit will be set to 1 while WDT up counter value reaches the selected WDT time-out interval. 0 = WDT time-out interrupt did not occur. 1 = WDT time-out interrupt occurred. **Note:** This bit is cleared by writing 1 to it.
[2]	**RSTF**	**WDT Time-out Reset Flag** This bit indicates the system has been reset by WDT time-out reset or not. 0 = WDT time-out reset did not occur. 1 = WDT time-out reset occurred. **Note:** This bit is cleared by writing 1 to it.
[1]	**RSTEN**	**WDT Time-out Reset Enable Bit (Write Protect)** Setting this bit will enable the WDT time-out reset function if the WDT up counter value has not been cleared after the specific WDT reset delay period expires. 0 = WDT time-out reset function Disabled. 1 = WDT time-out reset function Enabled. **Note:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register.
[0]	**Reserved**	Reserved.

WDT Alternative Control Register (WDT_ALTCTL)

Register	Offset	R/W	Description	Reset Value
WDT_ALTCTL	WDTn_BA+0x04	R/W	WDT Alternative Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**RSTDSEL**	

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1:0]	**RSTDSEL**	**WDT Reset Delay Selection (Write Protect)** When WDT time-out happened, user has a time named WDT Reset Delay Period to clear WDT counter by programming 0x5AA5 to RSTCNT to prevent WDT time-out reset happened. User can select a suitable setting of RSTDSEL for different WDT Reset Delay Period. 00 = WDT Reset Delay Period is 1026 * WDT_CLK. 01 = WDT Reset Delay Period is 130 * WDT_CLK. 10 = WDT Reset Delay Period is 18 * WDT_CLK. 11 = WDT Reset Delay Period is 3 * WDT_CLK. **Note 1:** This bit is write protected. Refer to the SYS_RLKTZS register or SYS_RLKTZNS register or SYS_RLKSUBM register. **Note 2:** This register will be reset to 0 if WDT time-out reset happened.

WDT Reset Counter Register (WDT_RSTCNT)

Register	Offset	R/W	Description	Reset Value
WDT_RSTCNT	WDTn_BA+0x08	W	WDT Reset Counter Register	0x0000_0000

31	30	29	28	27	26	25	24
RSTCNT							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RSTCNT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RSTCNT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RSTCNT							

Bits	Description	
[31:0]	**RSTCNT**	**WDT Reset Counter Register** Writing 0x00005AA5 to this field will reset the internal 20-bit WDT up counter value to 0. **Note:** Performing RSTCNT to reset counter needs 2 * WDT_CLK period to become active.

6.22 Window Watchdog Timer (WWDT)

6.22.1 Overview

The Window Watchdog Timer (WWDT) is used to perform a system reset within a specified window period to prevent software running to uncontrollable status by any unpredictable condition.

6.22.2 Features

- Three Window Watchdogs, one for TrustZone Secure (TZS), one for TrustZone Secure/Non-Secure (TZS/TZNS) and the other for SubM
- 6-bit down counter value (CNTDAT, WWDT_CNT[5:0]) and 6-bit compare value (CMPDAT, WWDT_CTL[21:16]) to make the WWDT time-out window period flexible
- Supports 4-bit value (PSCSEL, WWDT_CTL[11:8]) to programmable maximum 11-bit prescale counter period of WWDT counter
- WWDT counter suspends in Idle/Power-down mode
- WWDT0 can reset both real time Cortex-A35 and real time Cortex-M4 sub-systems
- WWDT1 can reset real time Cortex-A35 sub-system when WDT1RSTAEN (SYS_MISCRFCR[16]) is set
- WWDT1 can reset real time Cortex-M4 sub-system when WDT1RSTMEN (SYS_MISCRFCR[18]) is set
- WWDT2 can reset Cortex-M4 sub-system
- WWDT2 can reset Cortex-A35 sub-system when WDT2RSTAEN(SYS_MISCRFCR[17]) is set

6.22.3 Block Diagram

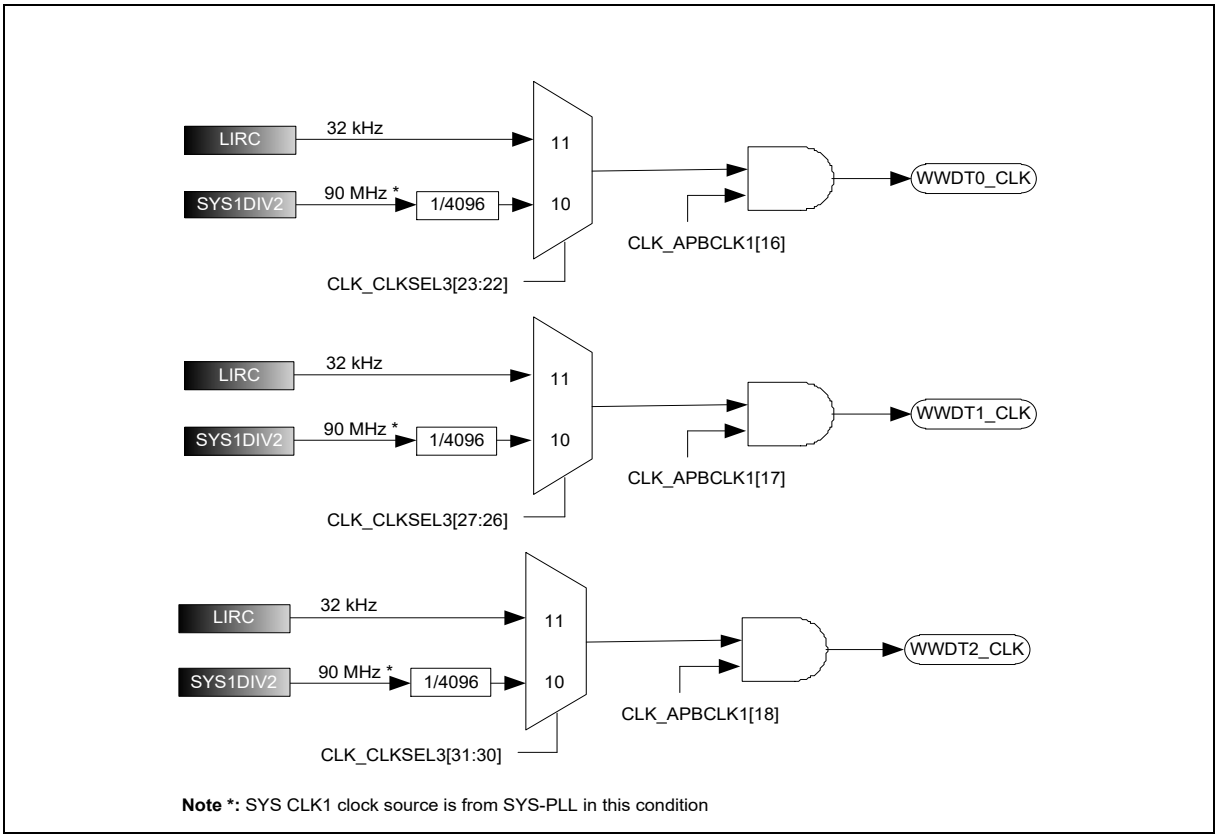


Figure 6.22-1 WWDT Clock Control

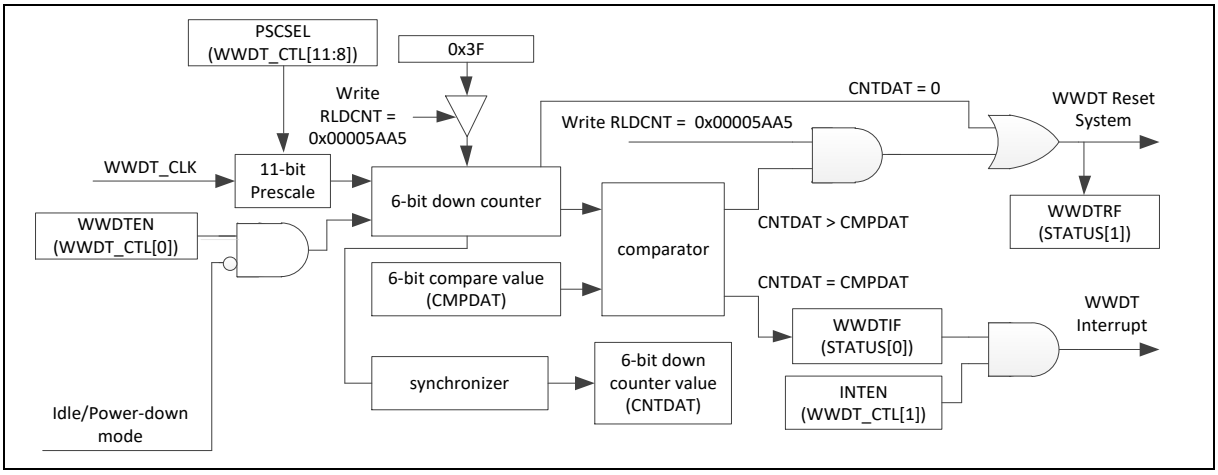


Figure 6.22-2 WWDT Block Diagram

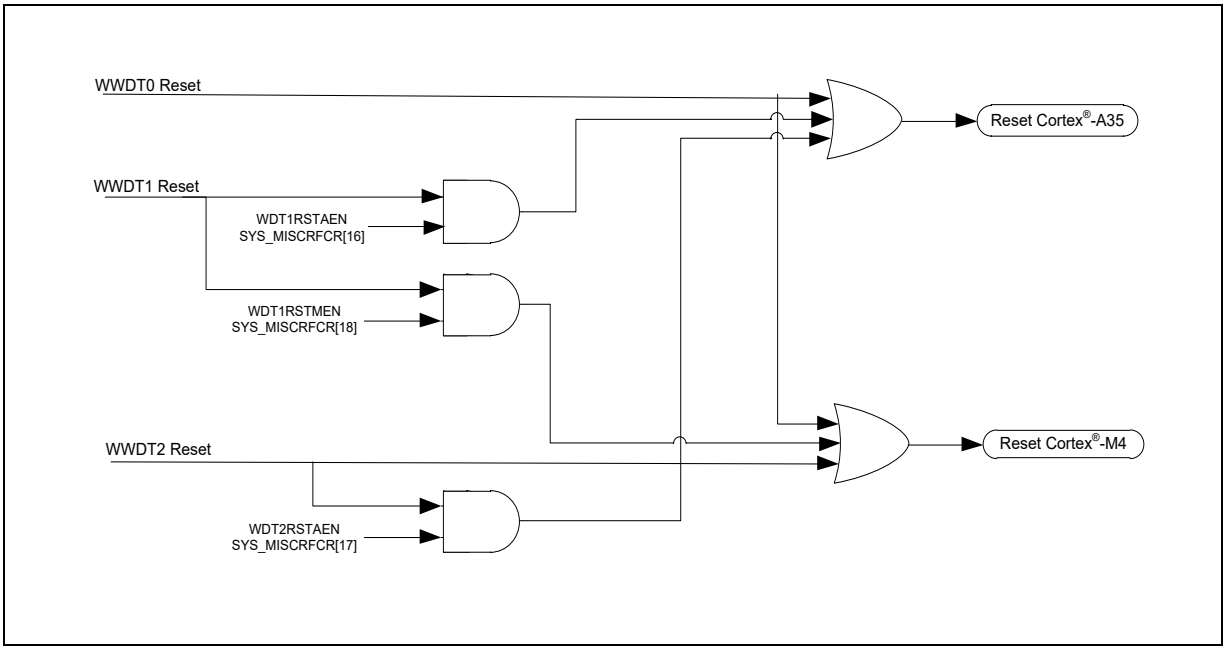


Figure 6.22-3 WWDT Reset Control Block Diagram

6.22.4 Basic Configuration

- Clock Source Configuration
  - Select the source of WWDT0 peripheral clock on WWDT0SEL (CLK_CLKSEL3[23:22])
  - Enable WWDT0 peripheral clock in WDT0CKEN (CLK_APBCLK1[16]).
  - Select the source of WWDT1 peripheral clock on WWDT1SEL (CLK_CLKSEL3[27:26])
  - Enable WWDT1 peripheral clock in WDT1CKEN (CLK_APBCLK1[17]).
  - Select the source of WWDT2 peripheral clock on WWDT2SEL (CLK_CLKSEL3[31:30])
  - Enable WWDT2 peripheral clock in WDT2CKEN (CLK_APBCLK1[18]).

6.22.5 Functional Description

The WWDT includes a 6-bit down counter with programmable prescale value to define different WWDT time-out intervals. The clock sources of 6-bit WWDT are based on SYS1DVI2 divide 4096 and 32 kHz internal low speed RC oscillator (LIRC) with a programmable 11-bit prescale counter value controlled by PSCSEL (WWDT_CTL[11:8]). Also, the correlate of PSCSEL (WWDT_CTL[11:8]) and prescale value are listed in Table 6.22-1.

PSCSEL	Prescaler Value	Max. Time-Out Period	Max. Time-Out Interval (WWDT_CLK = LIRC 32 kHz)
0000	1	1 * 64 * T_{WWDT}	1.95 ms
0001	2	2 * 64 * T_{WWDT}	3.90 ms
0010	4	4 * 64 * T_{WWDT}	7.8125 ms
0011	8	8 * 64 * T_{WWDT}	15.625 ms
0100	16	16 * 64 * T_{WWDT}	31.25 ms
0101	32	32 * 64 * T_{WWDT}	62.5 ms
0110	64	64 * 64 * T_{WWDT}	125 ms

0111	128	128 * 64 * T_{WWDT}	250 ms
1000	192	192 * 64 * T_{WWDT}	375 ms
1001	256	256 * 64 * T_{WWDT}	500 ms
1010	384	384 * 64 * T_{WWDT}	750 ms
1011	512	512 * 64 * T_{WWDT}	1000 ms
1100	768	768 * 64 * T_{WWDT}	1500 ms
1101	1024	1024 * 64 * T_{WWDT}	2000 ms
1110	1536	1536 * 64 * T_{WWDT}	3000 ms

Table 6.22-1 WWDT Prescaler Value Selection

6.22.5.1 WWDT Counting

When the WWDTEN (WWDT_CTL[0]) is set, WWDT down counter will start counting from 0x3F to 0. To prevent program runs to disable WWDT counter counting unexpected, the WWDT_CTL register can only be written once after chip is powered on or reset. User cannot disable WWDT counter counting (WWDTEN), change counter prescale period (PSCSEL) or change window compare value (CMPDAT) while WWDTEN (WWDT_CTL[0]) has been enabled by user unless chip is reset.

To avoid the system is reset while CPU clock is disabled, the WWDT counter will stop counting when CPU enters Idle/Power-down mode. After CPU enters normal mode, the WWDT counter will start down counting.

6.22.5.2 WWDT Compare Match Interrupt

During down counting by the WWDT counter, the WWDTIF (WWDT_STATUS[0]) is set to 1 while the WWDT counter value (CNTDAT) is equal to window compare value (CMPDAT) and WWDTIF can be cleared by user; if INTEN (WWDT_CTL[1]) is also set to 1 by user, the WWDT compare match interrupt signal is generated also while WWDTIF is set to 1 by hardware.

6.22.5.3 WWDT Reset System

Figure 6.22-4 shows three cases of WWDT reset and reload behavior.

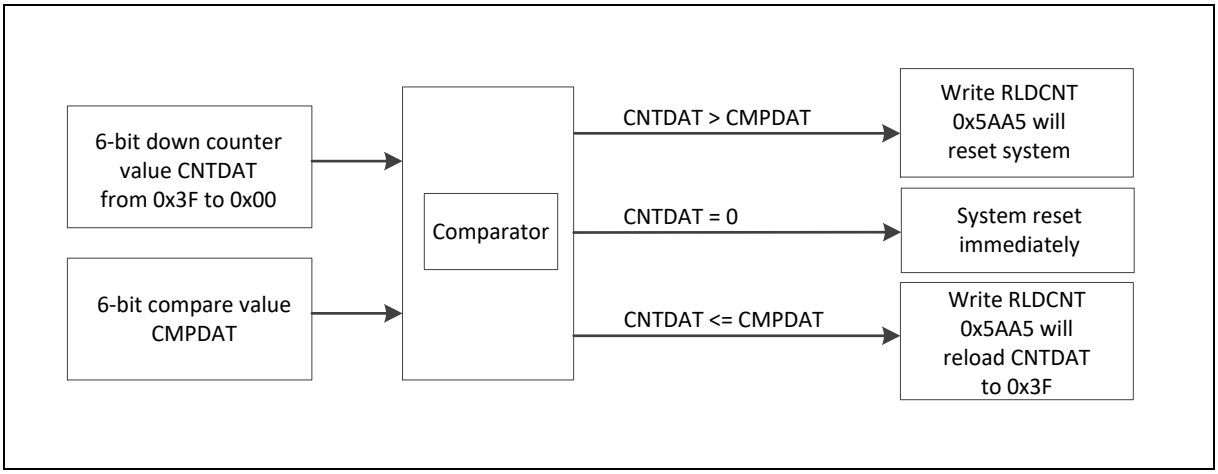


Figure 6.22-4 WWDT Reset and Reload Behavior

If the current CNTDAT (WWDT_CNT[5:0]) is larger than CMPDAT (WWDT_CTL[21:16]) and user writes 0x00005AA5 to the WWDT_RLDCNT register, the WWDT reset system signal will be generated to cause chip reset also. The waveform of WWDT reload counter when CNTDAT > CMPDAT is shown in

Figure 6.22-5.

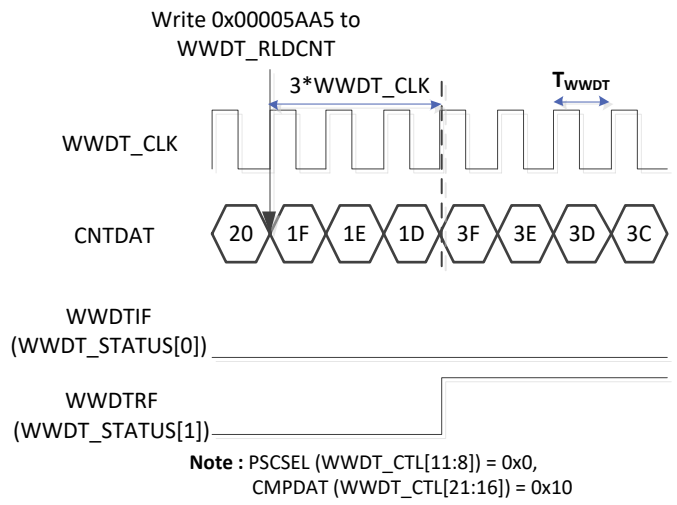


Figure 6.22-5 WWDT Reload Counter When CNTDAT > CMPDAT

When WWDTIF (WWDT_STATUS[0]) is generated, user must reload WWDT counter value to 0x3F by writing 0x00005AA5 to WWDT_RLDCNT register, and also to prevent WWDT counter value reached to 0 and generate WWDT reset system signal to inform system reset. Figure 6.22-6 shows the waveform of WWDT reload counter when CNTDAT < CMPDAT and Figure 6.22-7 shows WWDT generates reset system signal (WWDTRF) if user doesn't write 0x00005AA5 to WWDT_RLDCNT before WWDT counter value reaches 0.

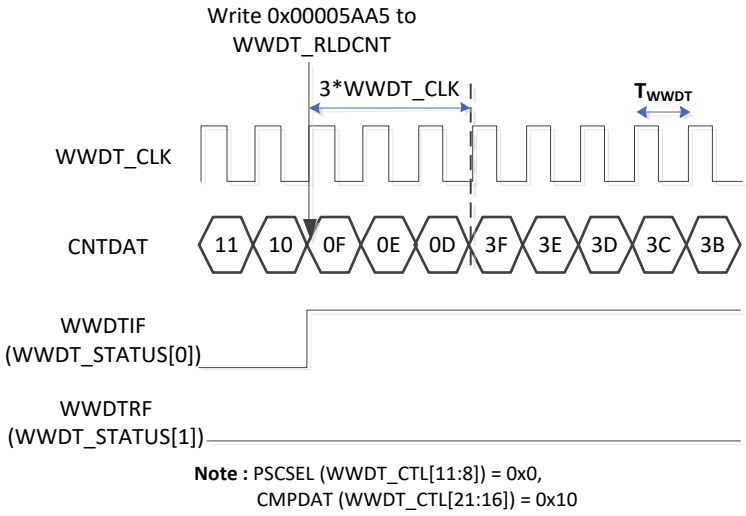


Figure 6.22-6 WWDT Reload Counter When CNTDAT < CMPDAT

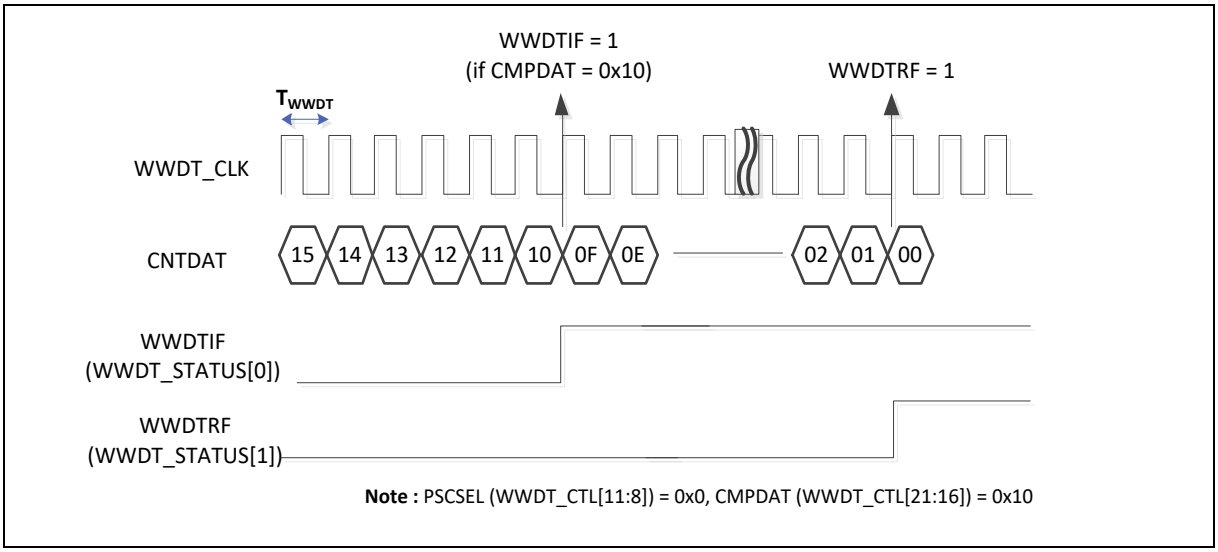


Figure 6.22-7 WWDT Interrupt and Reset Signals

6.22.5.4 WWDT Window Setting Limitation

When user writes 0x00005AA5 to WWDT_RLDCNT register to reload WWDT counter value to 0x3F, it needs 3 WWDT clocks to sync the reload command to actually perform reload action. Note that if user sets PSCSEL (WWDT_CTL[11:8]) to 0000, the counter prescale value should be as 1, and the CMPDAT (WWDT_CTL[21:16]) must be larger than 2. Otherwise, writing WWDT_RLDCNT register to reload WWDT counter value to 0x3F is unavailable, WWDTIF (WWDT_STATUS[0]) is generated, and WWDT reset system event always happened. The WWDT CMPDAT setting limitation is shown in Table 6.22-2.

If user sets CMPDATA as 0x3F and 0x0, the interrupt doesn't occur. The reset occurs when WWDT counts to 0x0, so the interrupt doesn't occur when CMPDATA is 0x0.

PSCSEL	Prescale Value	Valid CMPDAT Value
0000	1	0x3 ~ 0x3E
0001	2	0x2 ~ 0x3E
Others	Others	0x1 ~ 0x3E

Table 6.22-2 CMPDAT Setting Limitation

6.22.5.5 WWDT ICE Debug

When ICE is connected to MCU, the WWDT counter is counting or not by ICEDEBUG (WWDT_CTL[31]). The default value of ICEDEBUG is 0. The WWDT counter will stop counting when CPU is held by ICE. If ICEDEBUG is set to 1, the WWDT counter will keep counting no matter CPU is held by ICE or not.

6.22.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
WWDT Base Address: **WWDT0_BA = 0x4040_0100** **WWDT1_BA = 0x4044_0100** **WWDT2_BA = 0x4098_0100**				
WWDT_RLDCNT **n= 0,1,2**	WWDTn_BA+0x00	W	WWDT Reload Counter Register	0x0000_0000
WWDT_CTL **n= 0,1,2**	WWDTn_BA+0x04	R/W	WWDT Control Register	0x003F_0800
WWDT_STATUS **n= 0,1,2**	WWDTn_BA+0x08	R/W	WWDT Status Register	0x0000_0000
WWDT_CNT **n= 0,1,2**	WWDTn_BA+0x0C	R	WWDT Counter Value Register	0x0000_003F

6.22.7 Register Description

WWDT Reload Counter Register (WWDT_RLDCNT)

Register	Offset	R/W	Description	Reset Value
WWDT_RLDCNT	WWDTn_BA+0x00	W	WWDT Reload Counter Register	0x0000_0000

31	30	29	28	27	26	25	24
RLDCNT							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RLDCNT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RLDCNT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RLDCNT							

Bits	Description	
[31:0]	**RLDCNT**	**WWDT Reload Counter Register** Writing 0x00005AA5 to this register will reload the WWDT counter value to 0x3F. **Note:** User can only write WWDT_RLDCNT register to reload WWDT counter value when current WWDT counter value between 0 and CMPDAT (WWDT_CTL[21:16]). If user writes WWDT_RLDCNT when current WWDT counter value is larger than CMPDAT, WWDT reset signal will be generated.

WWDT Control Register (WWDT_CTL)

Register	Offset	R/W	Description	Reset Value
WWDT_CTL	WWDTn_BA+0x04	R/W	WWDT Control Register	0x003F_0800

Note: This register can be written only one time after chip is powered on or reset.

31	30	29	28	27	26	25	24
ICEDEBUG	Reserved						
23	22	21	20	19	18	17	16
Reserved		CMPDAT					
15	14	13	12	11	10	9	8
Reserved				PSCSEL			
7	6	5	4	3	2	1	0
Reserved						INTEN	WWDTEN

Bits	Description	
[31]	**ICEDEBUG**	**ICE Debug Mode Acknowledge Disable Bit** 0 = ICE debug mode acknowledgement effects WWDT counting. WWDT down counter will be held while CPU is held by ICE. 1 = ICE debug mode acknowledgement Disabled. **Note:** WWDT down counter will keep going no matter CPU is held by ICE or not.
[30:22]	**Reserved**	Reserved.
[21:16]	**CMPDAT**	**WWDT Window Compare Register** Set this register to adjust the valid reload window. **Note:** User can only write WWDT_RLDCNT register to reload WWDT counter value when current WWDT counter value between 0 and CMPDAT. If user writes WWDT_RLDCNT register when current WWDT counter value larger than CMPDAT, WWDT reset signal will generate.
[15:12]	**Reserved**	Reserved.
[11:8]	**PSCSEL**	**WWDT Counter Prescale Period Selection** 0000 = Pre-scale is 1; Max time-out period is 1 * 64 * WWDT_CLK. 0001 = Pre-scale is 2; Max time-out period is 2 * 64 * WWDT_CLK. 0010 = Pre-scale is 4; Max time-out period is 4 * 64 * WWDT_CLK. 0011 = Pre-scale is 8; Max time-out period is 8 * 64 * WWDT_CLK. 0100 = Pre-scale is 16; Max time-out period is 16 * 64 * WWDT_CLK. 0101 = Pre-scale is 32; Max time-out period is 32 * 64 * WWDT_CLK. 0110 = Pre-scale is 64; Max time-out period is 64 * 64 * WWDT_CLK. 0111 = Pre-scale is 128; Max time-out period is 128 * 64 * WWDT_CLK. 1000 = Pre-scale is 192; Max time-out period is 192 * 64 * WWDT_CLK. 1001 = Pre-scale is 256; Max time-out period is 256 * 64 * WWDT_CLK. 1010 = Pre-scale is 384; Max time-out period is 384 * 64 * WWDT_CLK. 1011 = Pre-scale is 512; Max time-out period is 512 * 64 * WWDT_CLK. 1100 = Pre-scale is 768; Max time-out period is 768 * 64 * WWDT_CLK. 1101 = Pre-scale is 1024; Max time-out period is 1024 * 64 * WWDT_CLK.

		1110 = Pre-scale is 1536; Max time-out period is 1536 * 64 * WWDT_CLK. 1111 = Pre-scale is 2048; Max time-out period is 2048 * 64 * WWDT_CLK.
[7:2]	**Reserved**	Reserved.
[1]	**INTEN**	**WWDT Interrupt Enable Bit** If this bit is enabled, the WWDT counter compare match interrupt signal is generated and inform to CPU. 0 = WWDT counter compare match interrupt Disabled. 1 = WWDT counter compare match interrupt Enabled.
[0]	**WWDTEN**	**WWDT Enable Bit** 0 = WWDT counter is stopped. 1 = WWDT counter starts counting.

WWDT Status Register (WWDT_STATUS)

Register	Offset	R/W	Description	Reset Value
WWDT_STATUS	WWDTn_BA+0x08	R/W	WWDT Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						WWDTRF	WWDTIF

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**WWDTRF**	**WWDT Timer-out Reset Flag** This bit indicates the system has been reset by WWDT time-out reset or not. 0 = WWDT time-out reset did not occur. 1 = WWDT time-out reset occurred. **Note:** This bit is cleared by writing 1 to it.
[0]	**WWDTIF**	**WWDT Compare Match Interrupt Flag** This bit indicates the interrupt flag status of WWDT while WWDT counter value matches CMPDAT (WWDT_CTL[21:16]). 0 = No effect. 1 = WWDT counter value matches CMPDAT. **Note:** This bit is cleared by writing 1 to it.

WWDT Counter Value Register (WWDT_CNT)

Register	Offset	R/W	Description	Reset Value
WWDT_CNT	WWDTn_BA+0x0C	R	WWDT Counter Value Register	0x0000_003F

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		CNTDAT					

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5:0]	**CNTDAT**	**WWDT Counter Value** CNTDAT will be updated continuously to monitor 6-bit WWDT down counter value.

6.23 Real Time Clock (RTC)

6.23.1 Overview

The Real Time Clock (RTC) controller provides the real time and calendar message. The RTC offers programmable time tick and alarm match interrupts. The data format of time and calendar messages are expressed in BCD format. A digital frequency compensation feature is available to compensate external crystal oscillator frequency accuracy.

6.23.2 Features

- Supports external power pin V_{BAT}.
- Supports real time counter in RTC_TIME (hour, minute, second) and calendar counter in RTC_CAL (year, month, day) for RTC time and calendar check.
- Supports alarm time (hour, minute, second) and calendar (year, month, day) settings in RTC_TALM and RTC_CALM.
- Supports alarm time (hour, minute, second) and calendar (year, month, day) mask enable in RTC_TAMSK and RTC_CAMSK.
- Selectable 12-hour or 24-hour time scale in RTC_CLKFMT register.
- Supports Leap Year indication in RTC_LEAPYEAR register.
- Supports Day of the Week counter in RTC_WEEKDAY register.
- Frequency of RTC clock source compensate by RTC_FREQADJ register.
- All time and calendar message expressed in BCD format.
- Supports periodic RTC Time Tick interrupt with 8 period interval options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second.
- Supports RTC Time Tick and Alarm Match interrupt.
- Supports 32 kHz clock output.
- Supports chip wake-up from Idle or Power-down mode while a RTC interrupt signal is generated.
- Supports 1 pairs dynamic loop tamper pin or 2 individual tamper pins.
- Built-in LXT frequency monitor.
- Supports 64 bytes spare registers and tamper pins detection to clear the content of these spare registers.

6.23.3 Block Diagram

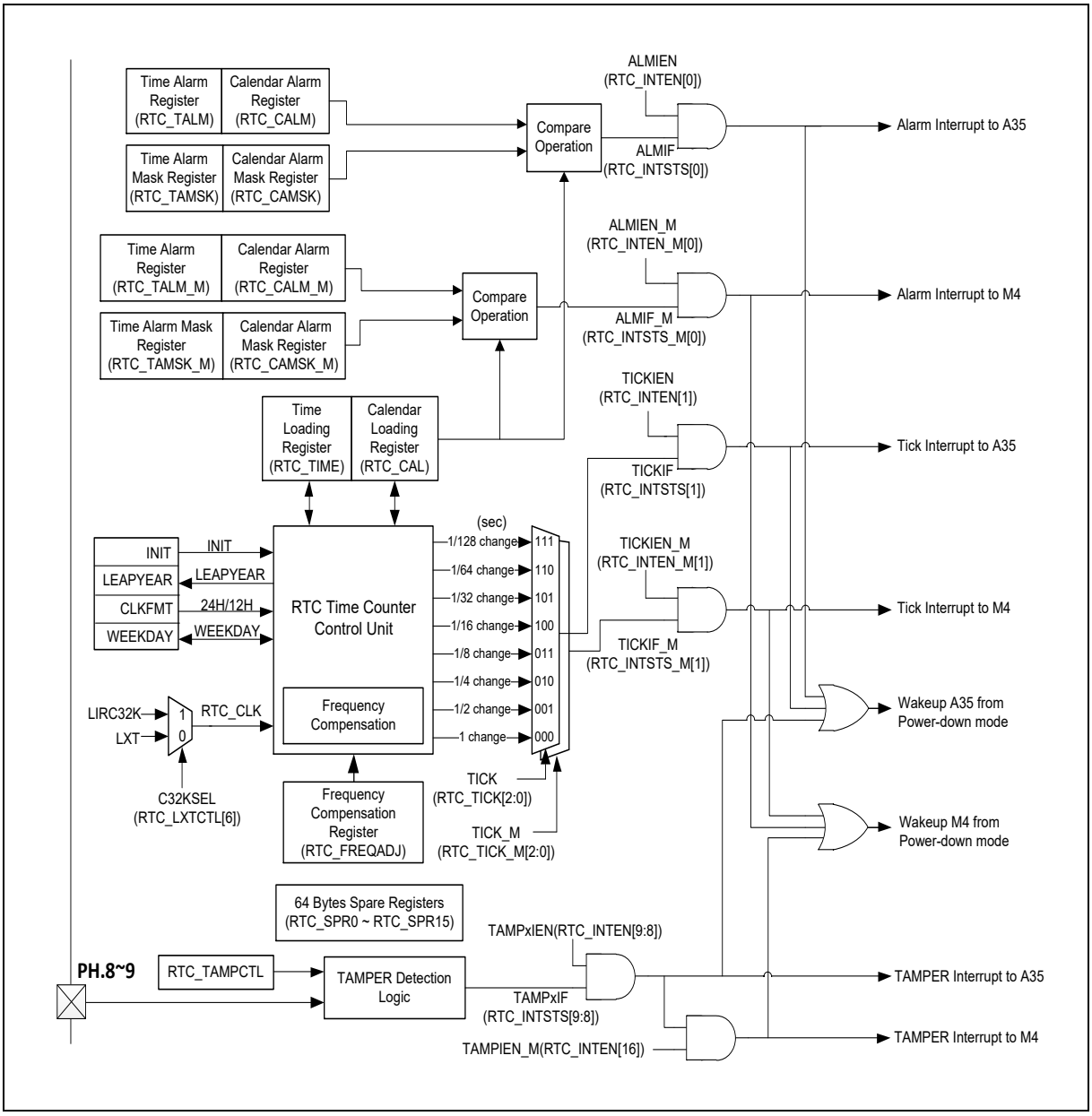


Figure 6.23-1 RTC Block Diagram

6.23.4 Basic Configuration

- Clock Source Configuration
  - The RTC controller clock source is enabled by RTCCKEN (APBCLK0[29]) and RTC Time Counter source is selected by RTC_LXTCTL[6] for LXT or LIRC32K.
- Pin Configuration

Group	Pin Name	GPIO	Other MF
X32	X32_OUT	Dedicated pin	NA
	X32_IN	Dedicated pin	NA
TAMPER	TAMPER0	PH.8	NA

	TAMPER1	PH.9	CLK_32KOUT
Power Control	RTC_RPWR	Dedicated pin	NA
	RTC_nRWAKE	Dedicated pin	NA

6.23.5 Functional Description

6.23.5.1 RTC Initiation

When a RTC block is powered on, RTC is at reset state. User has to write a number 0xa5eb1357 to RTC initial register INIT(RTC_INIT[31:0]) to make RTC leaving reset state. Once the INIT register is written as 0xa5eb1357, the RTC will be in normal active state permanently. User can read INIT[0](RTC_INIT[0]) to check the RTC is at normal active state or reset state.

6.23.5.2 RTC Control Registers Access Attribute

The RTC control registers access attributes are shown in Table 6.23-1.

Register	**ACTIVE = 0**	**ACTIVE = 1**					
		RTCSIAEN(SSPCC_SINFAEN[2]) =0			RTCSIAEN(SSPCC_SINFAEN[2]) =1		
		TZS	TZNS	SubM	TZS	TZNS	SubM
RTC_INIT	Available	R/W	R	R	R/W		
RTC_SINFASTS	Available	R	R	R	R		
RTC_FREQADJ	Available	R/W	R	R	R/W		
RTC_TIME	Not available	R/W	R	R	R/W		
RTC_CAL	Not available	R/W	R	R	R/W		
RTC_CLKFMT	Not available	R/W	R	R	R/W		
RTC_WEEKDAY	Not available	R/W	R	R	R/W		
RTC_TALM	Not available	R/W	R	RAZ/WI	R/W		RAZ/WI
RTC_CALM	Not available	R/W	R	RAZ/WI	R/W		RAZ/WI
RTC_LEAPYEAR	Not available	R	R	R	R		
RTC_INTEN	Available	R/W	R	RAZ/WI	R/W		RAZ/WI
RTC_INTSTS	Available	R/W	R	RAZ/WI	R/W		RAZ/WI
RTC_TICK	Not available	R/W	R	RAZ/WI	R/W		RAZ/WI
RTC_TAMSK	Not available	R/W	R	RAZ/WI	R/W		RAZ/WI
RTC_CAMSK	Not available	R/W	R	RAZ/WI	R/W		RAZ/WI
RTC_SPRCTL	Available	R/W	R	R	R/W		
RTC_SPRx	Available	R/W	R	R	R/W		
RTC_TALM_M	Not available	RAZ/WI		R/W	RAZ/WI		R/W
RTC_CALM_M	Not available	RAZ/WI		R/W	RAZ/WI		R/W
RTC_INTEN_M	Available	RAZ/WI		R/W	RAZ/WI		R/W

RTC_INTSTS_M	Available	RAZ/WI		R/W	RAZ/WI	R/W
RTC_TICK_M	Not available	RAZ/WI		R/W	RAZ/WI	R/W
RTC_TAMSK_M	Not available	RAZ/WI		R/W	RAZ/WI	R/W
RTC_CAMSK_M	Not available	RAZ/WI		R/W	RAZ/WI	R/W
RTC_LXTCTL	Available	R/W	R	R	R/W	
RTC_GPIOCTL0	Available	R/W	R	R	R/W	
RTC_TAMPCTL	Available	R/W	R	RAZ/WI	R/W	RAZ/WI
RTC_TAMPSEED	Available	R/W	R	RAZ/WI	R/W	RAZ/WI
RTC_TAMPTIME	Not available	R	R	RAZ/WI	R	RAZ/WI
RTC_TAMPCAL	Not available	R	R	RAZ/WI	R	RAZ/WI
RTC_CLKDCTL	Available	R/W	R	RAZ/WI	R/W	RAZ/WI
RTC_CLBR	Available	R/W	R	RAZ/WI	R/W	RAZ/WI
RTC_PWRCTL	Available	R/W	RAZ/WI		R/W	RAZ/WI
RTC_PWRSTS	Available	R	RAZ/WI		R	RAZ/WI

Table 6.23-1 RTC Read/Write Enable

6.23.5.3 Frequency Compensation

The Frequency compensation circuit supports dynamic compensation to adjust compensation value without reset the compensation circuit. The enable bit for dynamic compensation is DCOMPEN(RTC_CLKFMT[16]).

If dynamic compensation is enabled, the minimal interval to continuous writing RTC_FREQADJ is 0.24 seconds, and FCRBUSY(RTC_FREQADJ[31]) flag is used to indicate that new register write operation is prohibited. The compensate value will be loaded into the compensation circuit after 0.24~1.16 seconds when writing to RTC_FREQADJ.

The RTC_FREQADJ register allows user to make digital compensation to a clock input. Please follow the example and formula below to write the actual frequency of 32k crystal to RTC_FREQADJ register. Following are the compensation examples for higher or lower than 32768 Hz.

Example 1:

Frequency counter measurement: 32773.65 Hz

Integer Part: 32773 => RTC_FREQADJ[12:8] = 0x15. Refer to the INTEGER(RTC_FREQADJ[12:8]) to get detailed setting value.

Fraction Part: 0.65 X 64 = 41.6(0x2A) => RTC_FREQADJ[5:0]=0x2A

Example 2:

Frequency counter measurement: 32763.25 Hz

Integer part: 32763=> RTC_FREQADJ[12:8] = 0x0B. Refer to the INTEGER(RTC_FREQADJ[12:8]) to get detailed setting value.

Fraction part: 0.25 X 64 = 16(0x10) => RTC_FREQADJ[5:0] = 0x10

Note: The value of RTC_FREQADJ register will be the default value (0x0000_1000) while the compensation is not executed. User can utilize a frequency counter to measure RTC clock source via clock output function in manufacturing. In the meanwhile, user can use clock output function to check

the result of RTC frequency compensation.

6.23.5.4 Time and Calendar Counter

RTC_TIME and RTC_CAL are used to load the real time and calendar. RTC_TALM and RTC_CALM are used for alarm time and calendar setup.

6.23.5.5 12/24 hour Time Scale Selection

The 12/24 hour time scale selection depends on 24HEN (RTC_CLKFMT[0]). When RTC runs as 12-hour time scale mode, RTC_TIME[21] (the high bit of TENHR[1:0]) means AM/PM indication, if RTC_TIME[21] is 1, it indicates PM time message and RTC_TIME[21] is 0 indicates AM time message.) Table 6.23-2 shows RTC_TIME mapping table of 12/24 hour time scale selection.

24-Hour Time Scale (24HEN = 1)		**12-Hour Time Scale (PM Time + 0x20) (24HEN = 0) (PM Time + 0x20)**	
0x00 (AM12)	0x12 (PM12)	0x12 (AM12)	0x32 (PM12)
0x01 (AM01)	0x13 (PM01)	0x01 (AM01)	0x21 (PM01)
0x02 (AM02)	0x14 (PM02)	0x02 (AM02)	0x22 (PM02)
0x03 (AM03)	0x15 (PM03)	0x03 (AM03)	0x23 (PM03)
0x04 (AM04)	0x16 (PM04)	0x04 (AM04)	0x24 (PM04)
0x05 (AM05)	0x17 (PM05)	0x05 (AM05)	0x25 (PM05)
0x06 (AM06)	0x18 (PM06)	0x06 (AM06)	0x26 (PM06)
0x07 (AM07)	0x19 (PM07)	0x07 (AM07)	0x27 (PM07)
0x08 (AM08)	0x20 (PM08)	0x08 (AM08)	0x28 (PM08)
0x09 (AM09)	0x21 (PM09)	0x09 (AM09)	0x29 (PM09)
0x10 (AM10)	0x22 (PM10)	0x10 (AM10)	0x30 (PM10)
0x11 (AM11)	0x23 (PM11)	0x11 (AM11)	0x31 (PM11)
Note: The hour value written into RTC_TIME[21:16], messages are expressed in BCD format.			

Table 6.23-2 12/24 Hour Time Scale Selection

6.23.5.6 Day of the Week Counter

The RTC controller provides day of week in WEEKDAY bits (RTC_WEEKDAY[2:0]). The value is defined from 0 to 6 to represent Sunday to Saturday respectively.

6.23.5.7 Periodic Time Tick Interrupt

The periodic time tick interrupt has 8 period interval options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second that are selected by TICK bits (RTC_TICK[2:0]). When Periodic Time Tick interrupt is enabled by setting TICKIEN (RTC_INTEN[1]) to 1, the Periodic Time Tick interrupt is requested periodically in the period selected by RTC_TICK[2:0] settings.

6.23.5.8 Alarm Interrupt

When the real time and calendar message in RTC_TIME and RTC_CAL registers are equal to alarm time and calendar values in RTC_TALM and RTC_CALM registers, the RTC alarm interrupt flag ALMIF (RTC_INTSTS[0]) is set to 1 and the RTC alarm interrupt signal assert if the alarm interrupt enable ALMIEN (RTC_INTEN[0]) is enabled.

The RTC controller provides time alarm mask register (RTC_TAMSK register) and Calendar Alarm Mask Register (RTC_CAMSK register) to mask the specified digit and generate periodic interrupt without changing the alarm match condition in RTC_TALM and RTC_CALM registers in each alarm interrupt service routine.

6.23.5.9 32 kHz Clock Output

The RTC controller provides 32 kHz clock output to pin PH.9. User can set 32KOUTEN (RTC_LXTCTL[16]) to 1 and enable RTC, 32 kHz clock will output to pin PH.9.

6.23.5.10 Application Note

1. All data in RTC_TALM, RTC_CALM, RTC_TIME and RTC_CAL registers are all expressed in BCD format.
2. User has to make sure that the loaded values are reasonable. For example, Load RTC_CAL as 201a (year), 13 (month), 00 (day), or RTC_CAL does not match with RTC_WEEKDAY, etc.
3. In RTC_CAL and RTC_CALM, only 2 BCD digits are used to express “year”. The 2 BCD digits of xy means 20xy, rather than 19xy or 21xy.
4. Example of 12-Hour Time Setting
   If current RTC time is PM12:59:30 in 12-Hour Time Scale mode, the RTC_TIME setting as:
   (1) HOUR:
   RTC_TIME[21:16]: 0x32 (0x12+0x20) combined by TENHR (RTC_TIME[21:20]) is 0x3, HR (RTC_TIME[19:16]) is 0x2.
   (2) MIN:
   RTC_TIME[14:8]: 0x59 combined by TENMIN (RTC_TIME[14:12]) is 0x5, MIN (RTC_TIME[11:8]) is 0x9.
   (3) SEC:
   RTC_TIME[6:0]: 0x30 combined by TENSEC (RTC_TIME[6:4]) is 0x3, SEC (RTC_TIME[3:0]) is 0x0.

Table 6.23-3 shows registers value after both core power and battery power are first powered on.

Register	Reset State
RTC_INIT	0
RTC_CAL	15/8/8 (year/month/day)
RTC_TIME	00:00:00 (hour : minute : second)
RTC_CALM	00/00/00 (year/month/day)
RTC_TALM	00:00:00 (hour : minute : second)
RTC_CLKFMT	1 (24-hour mode)
RTC_WEEKDAY	6 (Saturday)
RTC_INTEN	0
RTC_INTSTS	0
RTC_LEAPYEAR	0
RTC_TICK	0

Table 6.23-3 Registers Value after Powered On

6.23.5.11 Spare Registers and Tamper Detector

The RTC module is equipped with 64 bytes spare registers to store user's important information. These spare registers are located in RTC domain, user needs to enable SPRRWEN (RTC_SPRCTL[2]) before writing one of 16 spare registers (RTC_SPR0 ~ RTC_SPR15).

When the transition condition defined in RTC_TAMPCTL is detected, tamper detected interrupt flag TAMPxIF (RTC_INTSTS[9:8]) will be generated. Meanwhile, the 64 bytes spare registers (RTC_SPR0 ~ RTC_SPR15) content will be cleared automatically by hardware to prevent the security data be disclosure and current RTC time and calendar will be loaded to RTC_TAMPTIME and RTC_TAMPCAL registers, these values only can be update again when all TAMPxIF are cleared to 0. And if TAMPxIF is set to 1, the interrupt is generated to GIC if the tamper detect interrupt enable RTC_TAMPxIEN (RTC_INTEN[9:8]) is actived also to NVIC if the SubM tamper detect interrupt enable TAMPIENSubM (RTC_INTEN[16]) is actived.

The RTC supports 1 pair dynamic loop tamper pin or 2 individual tamper pins. The DYNRATE (RTC_TAMPCTL[7:5]) determine a reference pattern bit duration as shown in Figure 6.23-2. In Table 6.23-4, setting DYNSRC (RTC_TAMPCTL[3]) can select detect signal as a new value that is generated by hardware random value generator or user defined SEED value (RTC_TAMPSEED[31:0]) for dynamic loop tamper pin. The TAMPxLV (RTC_TAMPCTL[4x+9], x=0, 1) depends on the level attribute of TAMPERx pin for individual tamper detection. The tamper control effection is shown in Table 6.23-5.

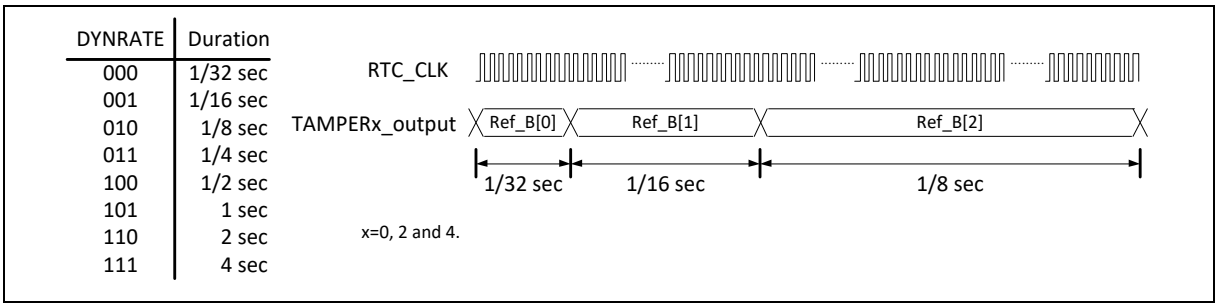


Figure 6.23-2 Dynamic Rate Definition

DYNPRxEN	DYNSRC	The Detect Signal Description
0	x	The TAMPERn is static tamper detection (n=0 ~ 1)
1	0	Generating new random value as reference pattern when the value runs out
1	1	Repeated user defined SEED value as reference pattern when the value runs out

Table 6.23-4 Dynamic Pattern Source Selection

Tamper Configuration			Tamper Control Bit Effection					
DYNPR0EN	TAMPER0EN	TAMPER1EN	TAMP0LV	TAMP1LV	TAMP0DBEN	TAMP1DBEN	TAMPER 0	TAMPER 1
0	0	0	-	-	-	-	-	-
0	0	1	-	V	-	V	-	Static
0	1	0	V	-	V	-	Static	X
0	1	1	V	V	V	V	Static	Static
1	x	0	-	-	-	-	Dynamic Out	-
1	x	1	-	-	-	-	Dynamic Out	Dynamic In

Table 6.23-5 Tamper Control Bit Effection for Pair 0

Static Tamper Programming Sequence Example

(1) Clean the TAMPxIF (RTC_INTSTS[x+8], x=0, 1).

(2) Set TAMPxLV (RTC_TAMPCTL[4x+9], x=0, 1) and TAMPxDBEN (RTC_TAMPCTL[4x+10], x=0, 1).

(3) Enable TAMPxEN (RTC_TAMPCTL[4x+8], x=0, 1).

Dynamic Tamper Programming Sequence Example

(1) Clean the TAMPxIF (RTC_INTSTS[x+8], x=0, 1).

(2) Fill the SEED (RTC_TAMPSEED[31:0]).

(3) Setting DYNSRC (RTC_TAMPCTL[3]) and DYNRATE (RTC_TAMPCTL[7:5]).

(4) Enable DYNPRxEN (RTC_TAMPCTL[8x+16], x=0).

(5) Enable TAMPxEN (RTC_TAMPCTL[4x+8], x=0, 1).

(6) Set SEEDRLD (RTC_TAMPCTL[4]).

6.23.5.12 Backup Domain GPIO Function

When PH.8 and PF.9 pins are not used as tamper function or CLK_32KOUT, they can be used as GPIO pin function. The IOCTLSEL (RTC_LXTCTL[8]) is used to select the PH.8 and PH.9 pins is controlled by RTC or GPIO module. Figure 6.23-3 shows backup I/O control diagram.

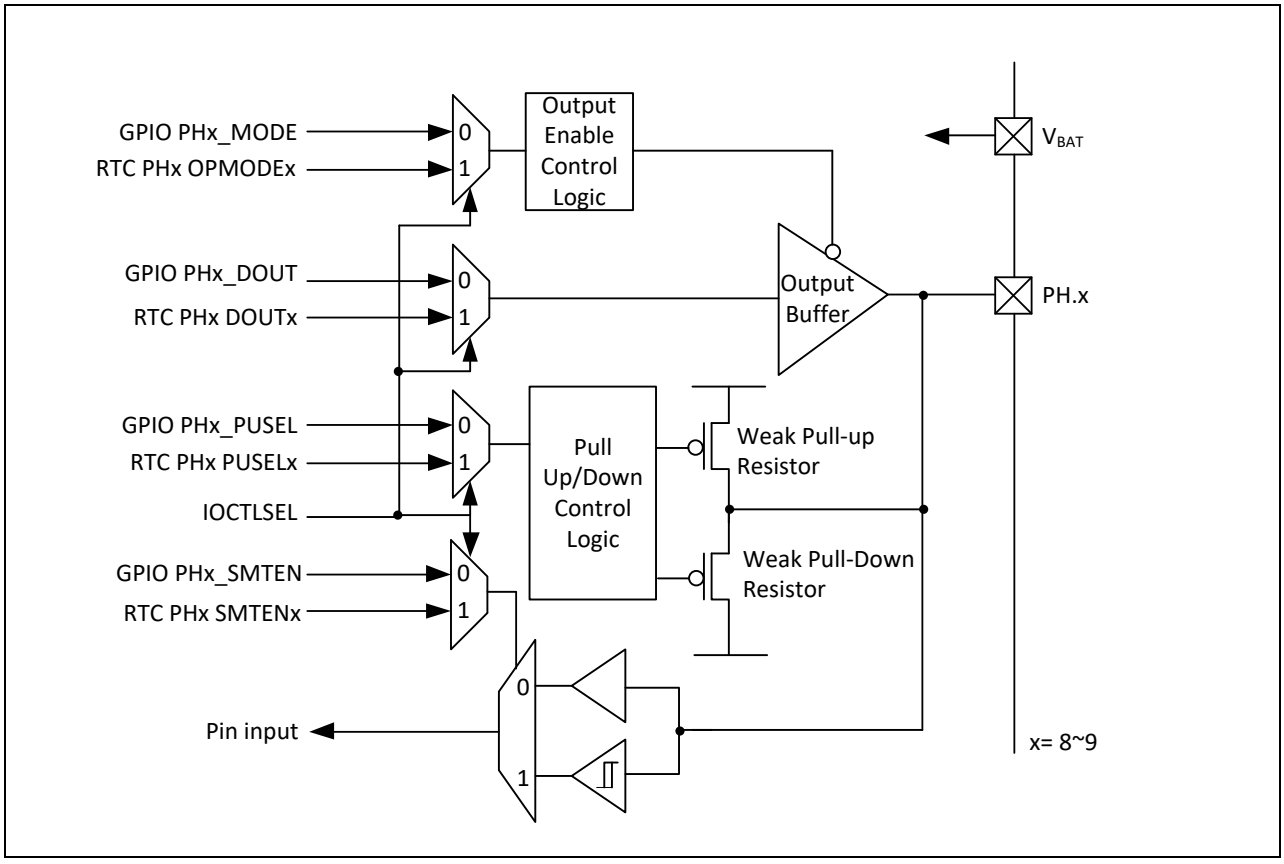


Figure 6.23-3 Backup I/O Control Diagram

6.23.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
RTC Base Address: **RTC_BA = 0x4041_0000**				
RTC_INIT	RTC_BA+0x00	R/W	RTC Initiation Register	0x0000_0000
RTC_SINFASTS	RTC_BA+0x04	R	RTC Shared Information Accessable Status Register	0x0000_0000
RTC_FREQADJ	RTC_BA+0x08	R/W	RTC Frequency Compensation Register	0x0000_1000
RTC_TIME	RTC_BA+0x0C	R/W	RTC Time Loading Register	0x0000_0000
RTC_CAL	RTC_BA+0x10	R/W	RTC Calendar Loading Register	0x0015_0808
RTC_CLKFMT	RTC_BA+0x14	R/W	RTC Time Scale Selection Register	0x0000_0001
RTC_WEEKDAY	RTC_BA+0x18	R/W	RTC Day of the Week Register	0x0000_0006
RTC_TALM	RTC_BA+0x1C	R/W	RTC Time Alarm Register	0x0000_0000
RTC_CALM	RTC_BA+0x20	R/W	RTC Calendar Alarm Register	0x0000_0000
RTC_LEAPYEAR	RTC_BA+0x24	R	RTC Leap Year Indicator Register	0x0000_0000
RTC_INTEN	RTC_BA+0x28	R/W	RTC Interrupt Enable Register	0x0000_0000
RTC_INTSTS	RTC_BA+0x2C	R/W	RTC Interrupt Status Register	0x0000_0000
RTC_TICK	RTC_BA+0x30	R/W	RTC Time Tick Register	0x0000_0000
RTC_TAMSK	RTC_BA+0x34	R/W	RTC Time Alarm Mask Register	0x0000_0000
RTC_CAMSK	RTC_BA+0x38	R/W	RTC Calendar Alarm Mask Register	0x0000_0000
RTC_SPRCTL	RTC_BA+0x3C	R/W	RTC Spare Functional Control Register	0x0000_0000
RTC_SPR0	RTC_BA+0x40	R/W	RTC Spare Register 0	0x0000_0000
RTC_SPR1	RTC_BA+0x44	R/W	RTC Spare Register 1	0x0000_0000
RTC_SPR2	RTC_BA+0x48	R/W	RTC Spare Register 2	0x0000_0000
RTC_SPR3	RTC_BA+0x4C	R/W	RTC Spare Register 3	0x0000_0000
RTC_SPR4	RTC_BA+0x50	R/W	RTC Spare Register 4	0x0000_0000
RTC_SPR5	RTC_BA+0x54	R/W	RTC Spare Register 5	0x0000_0000
RTC_SPR6	RTC_BA+0x58	R/W	RTC Spare Register 6	0x0000_0000
RTC_SPR7	RTC_BA+0x5C	R/W	RTC Spare Register 7	0x0000_0000
RTC_SPR8	RTC_BA+0x60	R/W	RTC Spare Register 8	0x0000_0000
RTC_SPR9	RTC_BA+0x64	R/W	RTC Spare Register 9	0x0000_0000
RTC_SPR10	RTC_BA+0x68	R/W	RTC Spare Register 10	0x0000_0000

RTC_SPR11	RTC_BA+0x6C	R/W	RTC Spare Register 11	0x0000_0000
RTC_SPR12	RTC_BA+0x70	R/W	RTC Spare Register 12	0x0000_0000
RTC_SPR13	RTC_BA+0x74	R/W	RTC Spare Register 13	0x0000_0000
RTC_SPR14	RTC_BA+0x78	R/W	RTC Spare Register 14	0x0000_0000
RTC_SPR15	RTC_BA+0x7C	R/W	RTC Spare Register 15	0x0000_0000
RTC_TALM_M	RTC_BA+0x9C	R/W	RTC Time Alarm Register for SubM (SubM)	0x0000_0000
RTC_CALM_M	RTC_BA+0xA0	R/W	RTC Calendar Alarm Register for SubM (SubM)	0x0000_0000
RTC_INTEN_M	RTC_BA+0xA8	R/W	RTC Interrupt Enable Register for SubM (SubM)	0x0000_0000
RTC_INTSTS_M	RTC_BA+0xAC	R/W	RTC Interrupt Status Register for SubM (SubM)	0x0000_0000
RTC_TICK_M	RTC_BA+0xB0	R/W	RTC Time Tick Register for SubM (SubM)	0x0000_0000
RTC_TAMSK_M	RTC_BA+0xB4	R/W	RTC Time Alarm Mask Register for SubM (SubM)	0x0000_0000
RTC_CAMSK_M	RTC_BA+0xB8	R/W	RTC Calendar Alarm Mask Register for SubM (SubM)	0x0000_0000
RTC_LXTCTL	RTC_BA+0x100	R/W	RTC 32.768 kHz Oscillator Control Register	0x0000_0000
RTC_GPIOCTL0	RTC_BA+0x104	R/W	RTC GPIO Control 0 Register	0x0000_0000
RTC_TAMPCTL	RTC_BA+0x120	R/W	RTC Tamper Pin Control Register	0x0000_0000
RTC_TAMPSEED	RTC_BA+0x128	R/W	RTC Tamper Dynamic Seed Register	0x0000_0000
RTC_TAMPTIME	RTC_BA+0x130	R	RTC Tamper Time Register	0x0000_0000
RTC_TAMPCAL	RTC_BA+0x134	R	RTC Tamper Calendar Register	0x0000_0000
RTC_CLKDCTL	RTC_BA+0x140	R/W	Clock Fail Detector Control Register	0x0000_0000
RTC_CDBR	RTC_BA+0x144	R/W	Clock Frequency Detector Boundary Register	0x00F0_000F
RTC_PWRCTL	RTC_BA+0x180	R/W	RTC Power Control Register (TZS)	0x0000_0000
RTC_PWRSTS	RTC_BA+0x184	R	RTC Power Control Status Register (TZS)	0x0000_0000

6.23.7 Register Description

RTC Initiation Register (RTC_INIT)

Register	Offset	R/W	Description	Reset Value
RTC_INIT	RTC_BA+0x00	R/W	RTC Initiation Register	0x0000_0000

31	30	29	28	27	26	25	24
INIT							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
INIT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
INIT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INIT							ACTIVE

Bits	Description	
[31:1]	**INIT**	**RTC Initiation (Write Only)** When RTC block is powered on, RTC is at reset state. User has to write a number (0x a5eb1357) to INIT to make RTC leave reset state. Once the INIT is written as 0xa5eb1357, the RTC will be in un-reset state permanently. **Note:** The INIT is a write-only field and read value will be always 0.
[0]	**ACTIVE**	**RTC Active Status (Read Only)** 0 = RTC is at reset state. 1 = RTC is at normal active state.

RTC Shared Information Accessable Status Register (RTC_SINFASTS)

Register	Offset	R/W	Description	Reset Value
RTC_SINFASTS	RTC_BA+0x04	R	RTC Shared Information Accessable Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**SINFASTS**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**SINFASTS**	**RTC Shared Information Accessable Status (Read Only)** 0 = RTCSIAEN(SSPCC_SINFAEN[2]) Disabled. 1 = RTCSIAEN(SSPCC_SINFAEN[2]) Enabled. **Note:** For RTC control registers access attribute please refer to the RTC Control Registers Access Attribute section.

RTC Frequency Compensation Register (RTC_FREQADJ)

Register	Offset	R/W	Description	Reset Value
RTC_FREQADJ	RTC_BA+0x08	R/W	RTC Frequency Compensation Register	0x0000_1000

31	30	29	28	27	26	25	24
FCRBUSY	Reserved						
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved			INTEGER				
7	6	5	4	3	2	1	0
Reserved		FRACTION					

Bits	Description	
[31]	**FCRBUSY**	**Frequency Compensation Register Write Operation Busy (Read Only)** 0 = The new register write operation is acceptable. 1 = The last write operation is in progress and new register write operation prohibited. **Note:** This bit is only used when DCOMPEN(RTC_CLKFMT[16]) is enabled.
[30:13]	**Reserved**	Reserved.
[12:8]	**INTEGER**	**Integer Part** 00000 = Integer part of detected value is 32752. 00001 = Integer part of detected value is 32753. 00010 = Integer part of detected value is 32754. 00011 = Integer part of detected value is 32755. 00100 = Integer part of detected value is 32756. 00101 = Integer part of detected value is 32757. 00110 = Integer part of detected value is 32758. 00111 = Integer part of detected value is 32759. 01000 = Integer part of detected value is 32760. 01001 = Integer part of detected value is 32761. 01010 = Integer part of detected value is 32762. 01011 = Integer part of detected value is 32763. 01100 = Integer part of detected value is 32764. 01101 = Integer part of detected value is 32765. 01110 = Integer part of detected value is 32766. 01111 = Integer part of detected value is 32767. 10000 = Integer part of detected value is 32768. 10001 = Integer part of detected value is 32769. 10010 = Integer part of detected value is 32770. 10011 = Integer part of detected value is 32771. 10100 = Integer part of detected value is 32772. 10101 = Integer part of detected value is 32773.

		10110 = Integer part of detected value is 32774. 10111 = Integer part of detected value is 32775. 11000 = Integer part of detected value is 32776. 11001 = Integer part of detected value is 32777. 11010 = Integer part of detected value is 32778. 11011 = Integer part of detected value is 32779. 11100 = Integer part of detected value is 32780. 11101 = Integer part of detected value is 32781. 11110 = Integer part of detected value is 32782. 11111 = Integer part of detected value is 32783.
[7:6]	**Reserved**	Reserved.
[5:0]	**FRACTION**	**Fraction Part** Formula: FRACTION = (fraction part of detected value) x 64. **Note:** Digit in FCR must be expressed as hexadecimal number.

Note: If dynamic compensation is not enabled, i.e. DCOMPEN(RTC_CLKFMT[16]) = 0, FREQADJ's counter will be reset for starting to compensate when writing RTC_FREQADJ. Imply RTC Time will be restarted.

RTC Time Loading Register (RTC_TIME)

Register	Offset	R/W	Description	Reset Value
RTC_TIME	RTC_BA+0x0C	R/W	RTC Time Loading Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		**TENHR**		**HR**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**TENMIN**			**MIN**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**TENSEC**			**SEC**			

Bits	Description	
[31:22]	**Reserved**	Reserved.
[21:20]	**TENHR**	**10-Hour Time Digit** When RTC runs as 12-hour time scale mode, RTC_TIME[21] (the high bit of TENHR[1:0]) means AM/PM indication (If RTC_TIME[21] is 1, it indicates PM time message.) **Note:** The reasonable value range is 0~2.
[19:16]	**HR**	**1-Hour Time Digit** **Note:** The reasonable value range is 0~9.
[15]	**Reserved**	Reserved.
[14:12]	**TENMIN**	**10-Min Time Digit** **Note:** The reasonable value range is 0~5.
[11:8]	**MIN**	**1-Min Time Digit** **Note:** The reasonable value range is 0~9.
[7]	**Reserved**	Reserved.
[6:4]	**TENSEC**	**10-Sec Time Digit** **Note:** The reasonable value range is 0~5.
[3:0]	**SEC**	**1-Sec Time Digit** **Note:** The reasonable value range is 0~9.

Note:

1. RTC_TIME is a BCD digit counter and RTC will not check loaded data.
2. The reasonable value range is listed in the parenthesis.
3. FREQADJ's counter will be reset for starting to Compensatie when writing RTC_TIME, RTC_CAL, RTC_WEEKDAY. Imply RTC Time will be restarted.

RTC Calendar Loading Register (RTC_CAL)

Register	Offset	R/W	Description	Reset Value
RTC_CAL	RTC_BA+0x10	R/W	RTC Calendar Loading Register	0x0015_0808

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
TENYEAR				YEAR			
15	14	13	12	11	10	9	8
Reserved			TENMON	MON			
7	6	5	4	3	2	1	0
Reserved		TENDAY		DAY			

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:20]	**TENYEAR**	**10-Year Calendar Digit** **Note:** The reasonable value range is 0~9.
[19:16]	**YEAR**	**1-Year Calendar Digit** **Note:** The reasonable value range is 0~9.
[15:13]	**Reserved**	Reserved.
[12]	**TENMON**	**10-Month Calendar Digit** **Note:** The reasonable value range is 0~1.
[11:8]	**MON**	**1-Month Calendar Digit** **Note:** The reasonable value range is 0~9.
[7:6]	**Reserved**	Reserved.
[5:4]	**TENDAY**	**10-Day Calendar Digit** **Note:** The reasonable value range is 0~3.
[3:0]	**DAY**	**1-Day Calendar Digit** **Note:** The reasonable value range is 0~9.

Note:

1. RTC_CAL is a BCD digit counter and RTC will not check loaded data.
2. The reasonable value range is listed in the parenthesis.
3. FREQADJ's counter will be reset for starting to Compensatie when writing RTC_TIME, RTC_CAL, RTC_WEEKDAY. Imply RTC Time will be restarted.

RTC Time Scale Selection Register (RTC_CLKFMT)

Register	Offset	R/W	Description	Reset Value
RTC_CLKFMT	RTC_BA+0x14	R/W	RTC Time Scale Selection Register	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							DCOMPEN
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							24HEN

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**DCOMPEN**	**Dynamic Compensation Enable Bit** 0 = Dynamic Compensation Disabled. 1 = Dynamic Compensation Enabled.
[15:1]	**Reserved**	Reserved.
[0]	**24HEN**	**24-hour / 12-hour Time Scale Selection** Indicates that RTC_TIME and RTC_TALM are in 24-hour time scale or 12-hour time scale 0 = 12-hour time scale with AM and PM indication selected. 1 = 24-hour time scale selected.

RTC Day of the Week Register (RTC_WEEKDAY)

Register	Offset	R/W	Description	Reset Value
RTC_WEEKDAY	RTC_BA+0x18	R/W	RTC Day of the Week Register	0x0000_0006

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**WEEKDAY**		

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2:0]	**WEEKDAY**	**Day of the Week Register** 000 = Sunday. 001 = Monday. 010 = Tuesday. 011 = Wednesday. 100 = Thursday. 101 = Friday. 110 = Saturday. 111 = Reserved.

Note: FREQADJ's counter will be reset for starting to Compensatie when writing RTC_TIME, RTC_CAL, RTC_WEEKDAY. Imply RTC Time will be restarted.

RTC Time Alarm Register (RTC_TALM)

Register	Offset	R/W	Description	Reset Value
RTC_TALM	RTC_BA+0x1C	R/W	RTC Time Alarm Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		TENHR		HR			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	TENMIN			MIN			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	TENSEC			SEC			

Bits	Description	
[31:22]	**Reserved**	Reserved.
[21:20]	**TENHR**	**10-Hour Time Digit of Alarm Setting** When RTC runs as 12-hour time scale mode, RTC_TIME[21] (the high bit of TENHR[1:0]) means AM/PM indication (If RTC_TIME[21] is 1, it indicates PM time message.) **Note:** The reasonable value range is 0~2.
[19:16]	**HR**	**1-Hour Time Digit of Alarm Setting** **Note:** The reasonable value range is 0~9.
[15]	**Reserved**	Reserved.
[14:12]	**TENMIN**	**10-Min Time Digit of Alarm Setting** **Note:** The reasonable value range is 0~5.
[11:8]	**MIN**	**1-Min Time Digit of Alarm Setting** **Note:** The reasonable value range is 0~9.
[7]	**Reserved**	Reserved.
[6:4]	**TENSEC**	**10-Sec Time Digit of Alarm Setting** **Note:** The reasonable value range is 0~5.
[3:0]	**SEC**	**1-Sec Time Digit of Alarm Setting** **Note:** The reasonable value range is 0~9.

Note:

1. RTC_TALM is a BCD digit counter.
2. The reasonable value range is listed in the parenthesis.

RTC Calendar Alarm Register (RTC_CALM)

Register	Offset	R/W	Description	Reset Value
RTC_CALM	RTC_BA+0x20	R/W	RTC Calendar Alarm Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TENYEAR				YEAR			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			TENMON	MON			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		TENDAY		DAY			

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:20]	**TENYEAR**	**10-Year Calendar Digit of Alarm Setting** **Note:** The reasonable value range is 0~9.
[19:16]	**YEAR**	**1-Year Calendar Digit of Alarm Setting** **Note:** The reasonable value range is 0~9.
[15:13]	**Reserved**	Reserved.
[12]	**TENMON**	**10-Month Calendar Digit of Alarm Setting** **Note:** The reasonable value range is 0~1.
[11:8]	**MON**	**1-Month Calendar Digit of Alarm Setting** **Note:** The reasonable value range is 0~9.
[7:6]	**Reserved**	Reserved.
[5:4]	**TENDAY**	**10-Day Calendar Digit of Alarm Setting** **Note:** The reasonable value range is 0~3.
[3:0]	**DAY**	**1-Day Calendar Digit of Alarm Setting** **Note:** The reasonable value range is 0~9.

Note:

1. RTC_CALM is a BCD digit counter.
2. The reasonable value range is listed in the parenthesis.

RTC Leap Year Indication Register (RTC_LEAPYEAR)

Register	Offset	R/W	Description	Reset Value
RTC_LEAPYEAR	RTC_BA+0x24	R	RTC Leap Year Indicator Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**LEAPYEAR**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**LEAPYEAR**	**Leap Year Indication (Read Only)** 0 = This year is not a leap year. 1 = This year is leap year.

RTC Interrupt Enable Register (RTC_INTEN)

Register	Offset	R/W	Description	Reset Value
RTC_INTEN	RTC_BA+0x28	R/W	RTC Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						CLKSTIEN	CLKFIEN
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							TAMPIENSubM
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						TAMP1IEN	TAMP0IEN
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				KPRSIEN	PSWIEN	TICKIEN	ALMIEN

Bits	Description	
[31:26]	**Reserved**	Reserved.
[25]	**CLKSTIEN**	**LXT Clock Frequency Monitor Stop Interrupt Enable Bit** 0 = LXT Frequency Stop interrupt Disabled. 1 = LXT Frequency Stop interrupt Enabled.
[24]	**CLKFIEN**	**LXT Clock Frequency Monitor Fail Interrupt Enable Bit** 0 = LXT Frequency Fail interrupt Disabled. 1 = LXT Frequency Fail interrupt Enabled.
[23:17]	**Reserved**	Reserved.
[16]	**TAMPIENSubM**	**Tamper Interrupt for SubM Enable Bit** 0 = Tamper interrupt for SubM Disabled. 1 = Tamper interrupt for SubM Enabled.
[15:10]	**Reserved**	Reserved.
[9]	**TAMP1IEN**	**Tamper 1 or Pair 0 Interrupt Enable Bit** Setting TAMP1IEN to 1 can also enable chip wake-up function when tamper 1 interrupt event is generated. 0 = Tamper 1 or Pair 0 interrupt Disabled. 1 = Tamper 1 or Pair 0 interrupt Enabled.
[8]	**TAMP0IEN**	**Tamper 0 Interrupt Enable Bit** Setting TAMP0IEN to 1 can also enable chip wake-up function when tamper 0 interrupt event is generated. 0 = Tamper 0 interrupt Disabled. 1 = Tamper 0 interrupt Enabled.
[7:4]	**Reserved**	Reserved.
[3]	**KPRSIEN**	**Key Pressed for 1 Second Interrupt Enable Bit** 0 = Key Pressed for 1 Second interrupt Disabled. 1 = Key Pressed for 1 Second interrupt Enabled.

[2]	**PSWIEN**	**Power Switch Interrupt Enable Bit** 0 = RTC Power Switch interrupt Disabled. 1 = RTC Power Switch interrupt Enabled.
[1]	**TICKIEN**	**Time Tick Interrupt Enable Bit** Setting TICKIEN to 1 can also enable chip wake-up function when RTC tick interrupt event is generated. 0 = RTC Time Tick interrupt Disabled. 1 = RTC Time Tick interrupt Enabled.
[0]	**ALMIEN**	**Alarm Interrupt Enable Bit** Setting ALMIEN to 1 can also enable chip wake-up function when RTC alarm interrupt event is generated. 0 = RTC Alarm interrupt Disabled. 1 = RTC Alarm interrupt Enabled.

RTC Interrupt Status Register (RTC_INTSTS)

Register	Offset	R/W	Description	Reset Value
RTC_INTSTS	RTC_BA+0x2C	R/W	RTC Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						CLKSTIF	CLKFIF
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						TAMP1IF	TAMP0IF
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				KPRSST	PSWST	TICKIF	ALMIF

Bits	Description	
[31:26]	**Reserved**	Reserved.
[25]	**CLKSTIF**	**LXT Clock Frequency Monitor Stop Interrupt Flag** 0 = LXT frequency is normal. 1 = LXT frequency is almost stop. **Note:** Write 1 to clear the bit to 0.
[24]	**CLKFIF**	**LXT Clock Frequency Monitor Fail Interrupt Flag** 0 = LXT frequency is normal. 1 = LXT frequency is abnormal. **Note:** Write 1 to clear the bit to 0.
[23:10]	**Reserved**	Reserved.
[9]	**TAMP1IF**	**Tamper 1 or Pair 0 Interrupt Flag** This bit is set when TAMP1_PIN detected level non-equal TAMP1LV (RTC_TAMPCTL[13]) or TAMP0_PIN and TAMP1_PIN disconnected during DYNPR0EN (RTC_TAMPCTL[15]) is activated. 0 = No Tamper 1 or Pair 0 interrupt flag is generated. 1 = Tamper 1 or Pair 0 interrupt flag is generated. **Note 1:** Write 1 to clear this bit. **Note 2:** This interrupt flag will be generated again when Tamper setting condition is not restoration. **Note 3:** Clear all TAPMxIF will clear RTC_TAMPTIME and RTC_TAMPCAL automatically.
[8]	**TAMP0IF**	**Tamper 0 Interrupt Flag** This bit is set when TAMP0_PIN detected level non-equal TAMP0LV (RTC_TAMPCTL[9]). 0 = No Tamper 0 interrupt flag is generated. 1 = Tamper 0 interrupt flag is generated. **Note 1:** Write 1 to clear this bit. **Note 2:** This interrupt flag will be generated again when Tamper setting condition is not restoration. **Note 3:** Clear all TAPMxIF will clear RTC_TAMPTIME and RTC_TAMPCAL automatically.

[7:4]	**Reserved**	Reserved.
[3]	**KPRSST**	**Key Pressed for 1 Second Interrupt Status** This bit high indicates the RTC_nRWAKE kept in low (user pressed the key) for 1 second when RTC_RPWR is high (stage machine is in Power_On state). **Note:** Write 1 to clear this bit.
[2]	**PSWST**	**Power Switch Interrupt Status** This bit high indicates the RTC_nRWAKE transition from high to low (user pressed the key) when RTC_RPWR is high (stage machine is in Power_On state). When state machine is not in Power_On state, RTC_nRWAKE transition from high to low will not set this bit high. **Note:** Write 1 to clear this bit.
[1]	**TICKIF**	**RTC Time Tick Interrupt Flag** 0 = Tick condition did not occur. 1 = Tick condition occurred. **Note:** Write 1 to clear this bit.
[0]	**ALMIF**	**RTC Alarm Interrupt Flag** 0 = Alarm condition is not matched. 1 = Alarm condition is matched. **Note:** Write 1 to clear this bit.

RTC Time Tick Register (RTC_TICK)

Register	Offset	R/W	Description	Reset Value
RTC_TICK	RTC_BA+0x30	R/W	RTC Time Tick Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**TICK**		

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2:0]	**TICK**	**Time Tick Register** These bits are used to select RTC time tick period for Periodic Time Tick Interrupt request. 000 = Time tick is 1 second. 001 = Time tick is 1/2 second. 010 = Time tick is 1/4 second. 011 = Time tick is 1/8 second. 100 = Time tick is 1/16 second. 101 = Time tick is 1/32 second. 110 = Time tick is 1/64 second. 111 = Time tick is 1/128 second.

RTC Time Alarm MASK Register (RTC_TAMSK)

Register	Offset	R/W	Description	Reset Value
RTC_TAMSK	RTC_BA+0x34	R/W	RTC Time Alarm Mask Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		MTENHR	MHR	MTENMIN	MMIN	MTENSEC	MSEC

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**MTENHR**	**Mask 10-Hour Time Digit of Alarm Setting**
[4]	**MHR**	**Mask 1-Hour Time Digit of Alarm Setting**
[3]	**MTENMIN**	**Mask 10-Min Time Digit of Alarm Setting**
[2]	**MMIN**	**Mask 1-Min Time Digit of Alarm Setting**
[1]	**MTENSEC**	**Mask 10-Sec Time Digit of Alarm Setting**
[0]	**MSEC**	**Mask 1-Sec Time Digit of Alarm Setting**

Note:

1. RTC_TALM is a BCD digit counter and RTC will not check loaded data.
2. The reasonable value range is listed in the parenthesis.
3. MTENHR/MHR base on 24 hour Time Scale.

RTC Calendar Alarm MASK Register (RTC_CAMSK)

Register	Offset	R/W	Description	Reset Value
RTC_CAMSK	RTC_BA+0x38	R/W	RTC Calendar Alarm Mask Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**MTENYEAR**	**MYEAR**	**MTENMON**	**MMON**	**MTENDAY**	**MDAY**

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**MTENYEAR**	**Mask 10-Year Calendar Digit of Alarm Setting**
[4]	**MYEAR**	**Mask 1-Year Calendar Digit of Alarm Setting**
[3]	**MTENMON**	**Mask 10-Month Calendar Digit of Alarm Setting**
[2]	**MMON**	**Mask 1-Month Calendar Digit of Alarm Setting**
[1]	**MTENDAY**	**Mask 10-Day Calendar Digit of Alarm Setting**
[0]	**MDAY**	**Mask 1-Day Calendar Digit of Alarm Setting**

Note:

1. RTC_CALM is a BCD digit counter and RTC will not check loaded data.
2. The reasonable value range is listed in the parenthesis.

RTC Spare Functional Control Register (RTC_SPRCTL)

Register	Offset	R/W	Description	Reset Value
RTC_SPRCTL	RTC_BA+0x3C	R/W	RTC Spare Functional Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							**LXTFCLR**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**SPRCSTS**	**Reserved**		**SPRRWEN**	**Reserved**	

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**LXTFCLR**	**LXT Clock Fail/Stop to Clear Spare Enable Bit** 0 = LXT Fail/Stop to clear Spare register content Disabled.. 1 = LXT Fail/Stop to clear Spare register content Enabled.
[15:6]	**Reserved**	Reserved.
[5]	**SPRCSTS**	**SPR Clear Flag** This bit indicates if the RTC_SPR0 ~RTC_SPR19 content is cleared when specify tamper event is detected. 0 = Spare register content is not cleared. 1 = Spare register content is cleared. **Note 1:** Write 1 to clear this bit. **Note 2:** This bit keeps 1 when RTC_INTSTS[13:8] is not equal to 0.
[4:3]	**Reserved**	Reserved.
[2]	**SPRRWEN**	**Spare Register Enable Bit** 0 = Spare register Disabled. 1 = Spare register Enabled. **Note:** When spare register is disabled, RTC_SPR0 ~ RTC_SPR19 cannot be accessed.
[1:0]	**Reserved**	Reserved.

RTC Spare Register (RTC_SPRx)

Register	Offset	R/W	Description	Reset Value
RTC_SPR0	RTC_BA+0x40	R/W	RTC Spare Register 0	0x0000_0000
RTC_SPR1	RTC_BA+0x44	R/W	RTC Spare Register 1	0x0000_0000
RTC_SPR2	RTC_BA+0x48	R/W	RTC Spare Register 2	0x0000_0000
RTC_SPR3	RTC_BA+0x4C	R/W	RTC Spare Register 3	0x0000_0000
RTC_SPR4	RTC_BA+0x50	R/W	RTC Spare Register 4	0x0000_0000
RTC_SPR5	RTC_BA+0x54	R/W	RTC Spare Register 5	0x0000_0000
RTC_SPR6	RTC_BA+0x58	R/W	RTC Spare Register 6	0x0000_0000
RTC_SPR7	RTC_BA+0x5C	R/W	RTC Spare Register 7	0x0000_0000
RTC_SPR8	RTC_BA+0x60	R/W	RTC Spare Register 8	0x0000_0000
RTC_SPR9	RTC_BA+0x64	R/W	RTC Spare Register 9	0x0000_0000
RTC_SPR10	RTC_BA+0x68	R/W	RTC Spare Register 10	0x0000_0000
RTC_SPR11	RTC_BA+0x6C	R/W	RTC Spare Register 11	0x0000_0000
RTC_SPR12	RTC_BA+0x70	R/W	RTC Spare Register 12	0x0000_0000
RTC_SPR13	RTC_BA+0x74	R/W	RTC Spare Register 13	0x0000_0000
RTC_SPR14	RTC_BA+0x78	R/W	RTC Spare Register 14	0x0000_0000
RTC_SPR15	RTC_BA+0x7C	R/W	RTC Spare Register 15	0x0000_0000

31	30	29	28	27	26	25	24
SPARE							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SPARE							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SPARE							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SPARE							

Bits	Description	
[31:0]	**SPARE**	**Spare Register** This field is used to store back-up information defined by user. This field will be cleared by hardware automatically in the following conditions, a tamper pin event is detected, LXT clock fail/stop event occurs if LXTFCLR(RTC_SPRCTL[16]) is 1, or after Flash mass operation.

RTC Time Alarm Register for SubM (RTC_TALM_M)

Register	Offset	R/W	Description	Reset Value
RTC_TALM_M	RTC_BA+0x9C	R/W	RTC Time Alarm Register for SubM (SubM)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		TENHRM		HRM			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	TENMINM			MINM			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	TENSECM			SECM			

Bits	Description	
[31:22]	**Reserved**	Reserved.
[21:20]	**TENHRM**	**10-Hour Time Digit of Alarm Setting** When RTC runs as 12-hour time scale mode, RTC_TIME[21] (the high bit of TENHR[1:0]) means AM/PM indication (If RTC_TIME[21] is 1, it indicates PM time message.) **Note:** The reasonable value range is 0~2.
[19:16]	**HRM**	**1-Hour Time Digit of Alarm Setting** **Note:** The reasonable value range is 0~9.
[15]	**Reserved**	Reserved.
[14:12]	**TENMINM**	**10-Min Time Digit of Alarm Setting** **Note:** The reasonable value range is 0~5.
[11:8]	**MINM**	**1-Min Time Digit of Alarm Setting** **Note:** The reasonable value range is 0~9.
[7]	**Reserved**	Reserved.
[6:4]	**TENSECM**	**10-Sec Time Digit of Alarm Setting** **Note:** The reasonable value range is 0~5.
[3:0]	**SECM**	**1-Sec Time Digit of Alarm Setting** **Note:** The reasonable value range is 0~9.

Note:

1. RTC_TALM_M is a BCD digit counter.
2. The reasonable value range is listed in the parenthesis.

RTC Calendar Alarm Register for SubM (RTC_CALM_M)

Register	Offset	R/W	Description	Reset Value
RTC_CALM_M	RTC_BA+0xA0	R/W	RTC Calendar Alarm Register for SubM (SubM)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
TENYEARM				YEARM			
15	14	13	12	11	10	9	8
Reserved			TENMONM	MONM			
7	6	5	4	3	2	1	0
Reserved		TENDAYM		DAYM			

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:20]	**TENYEARM**	**10-Year Calendar Digit of Alarm Setting** **Note:** The reasonable value range is 0~9.
[19:16]	**YEARM**	**1-Year Calendar Digit of Alarm Setting** **Note:** The reasonable value range is 0~9.
[15:13]	**Reserved**	Reserved.
[12]	**TENMONM**	**10-Month Calendar Digit of Alarm Setting** **Note:** The reasonable value range is 0~1.
[11:8]	**MONM**	**1-Month Calendar Digit of Alarm Setting** **Note:** The reasonable value range is 0~9.
[7:6]	**Reserved**	Reserved.
[5:4]	**TENDAYM**	**10-Day Calendar Digit of Alarm Setting** **Note:** The reasonable value range is 0~3.
[3:0]	**DAYM**	**1-Day Calendar Digit of Alarm Setting** **Note:** The reasonable value range is 0~9.

Note:

1. RTC_CALM_M is a BCD digit counter.
2. The reasonable value range is listed in the parenthesis.

RTC Interrupt Enable Register for SubM (RTC_INTEN_M)

Register	Offset	R/W	Description	Reset Value
RTC_INTEN_M	RTC_BA+0xA8	R/W	RTC Interrupt Enable Register for SubM (SubM)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						TICKIENM	ALMIENM

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**TICKIENM**	**Time Tick Interrupt Enable Bit** Setting TICKIEN to 1 can also enable chip wake-up function when RTC tick interrupt event is generated. 0 = RTC Time Tick interrupt Disabled. 1 = RTC Time Tick interrupt Enabled.
[0]	**ALMIENM**	**Alarm Interrupt Enable Bit** Setting ALMIEN to 1 can also enable chip wake-up function when RTC alarm interrupt event is generated. 0 = RTC Alarm interrupt Disabled. 1 = RTC Alarm interrupt Enabled.

RTC Interrupt Status Register for SubM (RTC_INTSTS_M)

Register	Offset	R/W	Description	Reset Value
RTC_INTSTS_M	RTC_BA+0xAC	R/W	RTC Interrupt Status Register for SubM (SubM)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**TICKIFM**	**ALMIFM**

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**TICKIFM**	**RTC Time Tick Interrupt Flag** 0 = Tick condition did not occur. 1 = Tick condition occurred. **Note:** Write 1 to clear this bit.
[0]	**ALMIFM**	**RTC Alarm Interrupt Flag** 0 = Alarm condition is not matched. 1 = Alarm condition is matched. **Note:** Write 1 to clear this bit.

RTC Time Tick Register for SubM (RTC_TICK_M)

Register	Offset	R/W	Description	Reset Value
RTC_TICK_M	RTC_BA+0xB0	R/W	RTC Time Tick Register for SubM (SubM)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					TICKM		

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2:0]	**TICKM**	**Time Tick Register** These bits are used to select RTC time tick period for Periodic Time Tick Interrupt request. 000 = Time tick is 1 second. 001 = Time tick is 1/2 second. 010 = Time tick is 1/4 second. 011 = Time tick is 1/8 second. 100 = Time tick is 1/16 second. 101 = Time tick is 1/32 second. 110 = Time tick is 1/64 second. 111 = Time tick is 1/128 second.

RTC Time Alarm MASK Register for SubM (RTC_TAMSK_M)

Register	Offset	R/W	Description	Reset Value
RTC_TAMSK_M	RTC_BA+0xB4	R/W	RTC Time Alarm Mask Register for SubM (SubM)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		MTENHRM	MHRM	MTENMINM	MMINM	MTENSECM	MSECM

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**MTENHRM**	**Mask 10-Hour Time Digit of Alarm Setting**
[4]	**MHRM**	**Mask 1-Hour Time Digit of Alarm Setting**
[3]	**MTENMINM**	**Mask 10-Min Time Digit of Alarm Setting**
[2]	**MMINM**	**Mask 1-Min Time Digit of Alarm Setting**
[1]	**MTENSECM**	**Mask 10-Sec Time Digit of Alarm Setting**
[0]	**MSECM**	**Mask 1-Sec Time Digit of Alarm Setting**

Note:

1. RTC_TALM_M is a BCD digit counter and RTC will not check loaded data.
2. The reasonable value range is listed in the parenthesis.
3. MTENHR/MHR base on 24 hour Time Scale.

RTC Calendar Alarm MASK Register for SubM (RTC_CAMSK_M)

Register	Offset	R/W	Description	Reset Value
RTC_CAMSK_M	RTC_BA+0xB8	R/W	RTC Calendar Alarm Mask Register for SubM (SubM)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		MTENYEARM	MYEARM	MTENMONM	MMONM	MTENDAYM	MDAYM

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**MTENYEARM**	**Mask 10-Year Calendar Digit of Alarm Setting**
[4]	**MYEARM**	**Mask 1-Year Calendar Digit of Alarm Setting**
[3]	**MTENMONM**	**Mask 10-Month Calendar Digit of Alarm Setting**
[2]	**MMONM**	**Mask 1-Month Calendar Digit of Alarm Setting**
[1]	**MTENDAYM**	**Mask 10-Day Calendar Digit of Alarm Setting**
[0]	**MDAYM**	**Mask 1-Day Calendar Digit of Alarm Setting**

Note:

1. RTC_CALM_M is a BCD digit counter and RTC will not check loaded data.
2. The reasonable value range is listed in the parenthesis.

RTC 32K Oscillator Control Register (RTC_LXTCTL)

Register	Offset	R/W	Description	Reset Value
RTC_LXTCTL	RTC_BA+0x100	R/W	RTC 32.768 kHz Oscillator Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							32KOUTEN
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							IOCTLSEL
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	C32KSEL	Reserved					

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**32KOUTEN**	**Clock 32 kHz Output Enable Bit** When TAMP1EN is disabled, PH.9 pin(TAMPERx pin) can be used as clock 32 kHz output. 0 = Clock 32 kHz output to PH.9 pin Disabled. 1 = Clock 32 kHz output to PH.9 pin Enabled.
[15:9]	**Reserved**	Reserved.
[8]	**IOCTLSEL**	**I/O Pin Backup Control Selection** When TAMPxEN or 32KOUTEN is disabled, PH.8 pin and PH.9 pin(TAMPERx pin) can be used as GPIO function. User can program IOCTLSEL to decide PH.8~9 I/O function is controlled by system power domain GPIO module or V_{BAT} power domain RTC_GPIOCTL0 control register. 0 = PH.8~9 pin I/O function is controlled by GPIO module. 1 = PH.8~9 pin I/O function is controlled by V_{BAT} power domain. **Note:** IOCTLSEL will automatically be set by hardware to 1 when system power is off and any writable RTC registers has been written at RTCCKEN(CLK_APBCLK0[29]) enabled.
[7]	**Reserved**	Reserved.
[6]	**C32KSEL**	**Clock 32K Source Selection** 0 = Clock source from external low speed crystal oscillator (LXT). 1 = Clock source from internal low speed RC 32K oscillator (LIRC32K).
[5:0]	**Reserved**	Reserved.

RTC GPIO Control Register0 (RTC_GPIOCTL0)

Register	Offset	R/W	Description	Reset Value
RTC_GPIOCTL0	RTC_BA+0x104	R/W	RTC GPIO Control 0 Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	SMTEN1	PUSEL1		DINOFF1	DOUT1	OPMODE1	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	SMTEN0	PUSEL0		DINOFF0	DOUT0	OPMODE0	

Bits	Description	
[31:15]	**Reserved**	Reserved.
[14]	**SMTEN1**	**Input Schmitt Trigger Enable Bit** 0 = PH.9 input schmitt trigger function Disabled. 1 = PH.9 input schmitt trigger function Enabled.
[13:12]	**PUSEL1**	**I/O Pull-up and Pull-down Enable Bits** Determine PH.9 I/O pull-up or pull-down. 00 = PH.9 pull-up and pull-down Disabled. 01 = PH.9 pull-up Enabled. 10 = PH.9 pull-down Enabled. 11 = PH.9 pull-up and pull-down Disabled. **Note:** Basically, the pull-up control and pull-down control has following behavior limitation. The independent pull-up / pull-down control register only valid when OPMODE1 set as input tri-state and open-drain mode.
[11]	**DINOFF1**	**I/O Pin Digital Input Path Disable Bit** 0 = PH.9 digital input path Enabled. 1 = PH.9 digital input path Disabled (digital input tied to low).
[10]	**DOUT1**	**I/O Output Data** 0 = PH.9 output low. 1 = PH.9 output high.
[9:8]	**OPMODE1**	**I/O Operation Mode** 00 = PH.9 is input only mode. 01 = PH.9 is output push pull mode. 10 = PH.9 is open drain mode. 11 = Reserved.
[7]	**Reserved**	Reserved.
[6]	**SMTEN0**	**Input Schmitt Trigger Enable Bit**

		0 = PH.8 input schmitt trigger function Disabled. 1 = PH.8 input schmitt trigger function Enabled.
[5:4]	**PUSEL0**	**I/O Pull-up and Pull-down Enable Bits** Determine PH.8 I/O pull-up or pull-down. 00 = PH.8 pull-up and pull-down Disabled. 01 = PH.8 pull-up Enabled. 10 = PH.8 pull-down Enabled. 11 = PH.8 pull-up and pull-down Disabled. **Note:** Basically, the pull-up control and pull-down control has following behavior limitation. The independent pull-up / pull-down control register only valid when OPMODE0 set as input tri-state and open-drain mode.
[3]	**DINOFF0**	**I/O Pin Digital Input Path Disable Bit** 0 = PH.8 digital input path Enabled. 1 = PH.8 digital input path Disabled (digital input tied to low).
[2]	**DOUT0**	**I/O Output Data** 0 = PH.8 output low. 1 = PH.8 output high.
[1:0]	**OPMODE0**	**I/O Operation Mode** 00 = PH.8 is input only mode. 01 = PH.8 is output push pull mode. 10 = PH.8 is open drain mode. 11 = Reserved.

Note: RTC GPIO will be input mode when RTC registers have never been programmed. User needs to avoid the pin floating problem.

RTC Tamper Pin Control Register (RTC_TAMPCTL)

Register	Offset	R/W	Description	Reset Value
RTC_TAMPCTL	RTC_BA+0x120	R/W	RTC Tamper Pin Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
DYNPR0EN	TAMP1DEN	TAMP1LV	TAMP1EN	Reserved	TAMP0DEN	TAMP0LV	TAMP0EN
7	6	5	4	3	2	1	0
DYNRATE			SEEDRLD	DYNSRC	Reserved		

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15]	**DYNPR0EN**	**Dynamic Pair 0 Enable Bit** 0 = Static detect. 1 = Dynamic detect.
[14]	**TAMP1DEN**	**Tamper 1 De-bounce Enable Bit** 0 = Tamper 1 de-bounce Disabled. 1 = Tamper 1 de-bounce Enabled, tamper detection pin will sync 1 RTC clock.
[13]	**TAMP1LV**	**Tamper 1 Level** This bit depends on level attribute of tamper pin for static tamper detection. 0 = Detect voltage level is low. 1 = Detect voltage level is high.
[12]	**TAMP1EN**	**Tamper 1 Detect Enable Bit** 0 = Tamper 1 detect Disabled. 1 = Tamper 1 detect Enabled. **Note:** The reference is RTC-clock. Tamper detector need sync 2 ~ 3 RTC-clock.
[11]	**Reserved**	Reserved.
[10]	**TAMP0DEN**	**Tamper 0 De-bounce Enable Bit** 0 = Tamper 0 de-bounce Disabled. 1 = Tamper 0 de-bounce Enabled; tamper detection pin will sync 1 RTC clock.
[9]	**TAMP0LV**	**Tamper 0 Level** This bit depends on level attribute of tamper pin for static tamper detection. 0 = Detect voltage level is low. 1 = Detect voltage level is high.
[8]	**TAMP0EN**	**Tamper0 Detect Enable Bit** 0 = Tamper 0 detect Disabled. 1 = Tamper 0 detect Enabled.

		Note: The reference is RTC-clock. Tamper detector need sync 2 ~ 3 RTC-clock.
[7:5]	**DYNRATE**	**Dynamic Change Rate** This item is choice the dynamic tamper output change rate. 000 = 210 * RTC_CLK. 001 = 211 * RTC_CLK. 010 = 212 * RTC_CLK. 011 = 213 * RTC_CLK. 100 = 214 * RTC_CLK. 101 = 215 * RTC_CLK. 110 = 216 * RTC_CLK. 111 = 217 * RTC_CLK. **Note:** After revising this field, set SEEDRLD (RTC_TAMPCTL[4]) can reload change rate immediately.
[4]	**SEEDRLD**	**Reload New Seed for PRNG Engine** Setting this bit, the tamper configuration will be reloaded. 0 = Generating key based on the current seed. 1 = Reload new seed. **Note 1:** Before this bit is set, the tamper configuration should be set to complete and this bit will be auto clear to 0 after reload new seed completed. **Note 2:** The reference is RTC-clock. Tamper detector need sync 2 ~ 3 RTC-clock.
[3]	**DYNSRC**	**Dynamic Reference Pattern** This fields determine the new reference pattern when the current pattern runs out in dynamic pair mode. 0 = The new reference pattern is generated by random number generator when the reference pattern run out. 1 = The new reference pattern is repeated from SEED (RTC_TAMPSEED[31:0]) when the reference pattern run out. **Note:** After this bit is modified, the SEEDRLD (RTC_TAMPCTL[4]) should be set.
[2:0]	**Reserved**	Reserved.

RTC Tamper Dynamic Seed Register (RTC_TAMPSEED)

Register	Offset	R/W	Description	Reset Value
RTC_TAMPSEED	RTC_BA+0x128	R/W	RTC Tamper Dynamic Seed Register	0x0000_0000

31	30	29	28	27	26	25	24
SEED							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SEED							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SEED							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SEED							

Bits	Description	
[31:0]	**SEED**	Seed Value

RTC Tamper Time Register (RTC_TAMPTIME)

Register	Offset	R/W	Description	Reset Value
RTC_TAMPTIME	RTC_BA+0x130	R	RTC Tamper Time Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		TENHR		HR			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	TENMIN			MIN			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	TENSEC			SEC			

Bits	Description	
[31:22]	**Reserved**	Reserved.
[21:20]	**TENHR**	**10-Hour Time Digit of TAMPER Time** **Note 1:** The reasonable value range is 0~2. **Note 2:** 24-hour time scale only.
[19:16]	**HR**	**1-Hour Time Digit of TAMPER Time** **Note:** The reasonable value range is 0~9.
[15]	**Reserved**	Reserved.
[14:12]	**TENMIN**	**10-Min Time Digit of TAMPER Time** **Note:** The reasonable value range is 0~5.
[11:8]	**MIN**	**1-Min Time Digit of TAMPER Time** **Note:** The reasonable value range is 0~9.
[7]	**Reserved**	Reserved.
[6:4]	**TENSEC**	**10-Sec Time Digit of TAMPER Time** **Note:** The reasonable value range is 0~5.
[3:0]	**SEC**	**1-Sec Time Digit of TAMPER Time** **Note:** The reasonable value range is 0~9.

Note:

1. RTC_TAMPTIME is a BCD digit counter.
2. This fields can't update until all TAMPxIF are cleared.

RTC Tamper Calendar Register (RTC_TAMPCAL)

Register	Offset	R/W	Description	Reset Value
RTC_TAMPCAL	RTC_BA+0x134	R	RTC Tamper Calendar Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
TENYEAR				YEAR			
15	14	13	12	11	10	9	8
Reserved			TENMON	MON			
7	6	5	4	3	2	1	0
Reserved		TENDAY		DAY			

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:20]	**TENYEAR**	**10-Year Calendar Digit of TAMPER Calendar** **Note:** The reasonable value range is 0~9.
[19:16]	**YEAR**	**1-Year Calendar Digit of TAMPER Calendar** **Note:** The reasonable value range is 0~9.
[15:13]	**Reserved**	Reserved.
[12]	**TENMON**	**10-Month Calendar Digit of TAMPER Calendar** **Note:** The reasonable value range is 0~1.
[11:8]	**MON**	**1-Month Calendar Digit of TAMPER Calendar** **Note:** The reasonable value range is 0~9.
[7:6]	**Reserved**	Reserved.
[5:4]	**TENDAY**	**10-Day Calendar Digit of TAMPER Calendar** **Note:** The reasonable value range is 0~3.
[3:0]	**DAY**	**1-Day Calendar Digit of TAMPER Calendar** **Note:** The reasonable value range is 0~9.

Note:

1. RTC_TAMPCAL is a BCD digit counter.
2. This fields can't be updated until all TAMPxIF are cleared

RTC Clock Fail Detector Control Register (RTC_CLKDCTL)

Register	Offset	R/W	Description	Reset Value
RTC_CLKDCTL	RTC_BA+0x140	R/W	Clock Fail Detector Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						LXTSLOWF	SWLIRCF
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					LXTSTSW	LXTFSW	LXTFDEN

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17]	**LXTSLOWF**	**LXT Slower Than LIRC32K Flag (Read Only)** 0 = LXT frequency faster than LIRC32K. 1 = LXT frequency is slowly. **Note:** LXTSLOWF is vaild during CLKSTIF (RTC_INTSTS[25]) or CLKFIF (RTC_INTSTS[24]) rising.
[16]	**SWLIRCF**	**LXT Clock Detector Fail/Stop Switch LIRC32K Flag (Read Only)** 0 = RTC clock source from LXT. 1 = RTC clock source from LIRC32K.
[15:3]	**Reserved**	Reserved.
[2]	**LXTSTSW**	**LXT Clock Stop Detector Switch LIRC32K Enable Bit** 0 = LXT clock Stop switch LIRC32K Disabled. 1 = LXT clock Stop detector rise, RTC clock source switched from LIRC32K.
[1]	**LXTFSW**	**LXT Clock Fail Detector Switch LIRC32K Enable Bit** 0 = LXT clock Fail switch LIRC32K Disabled. 1 = LXT clock Fail detector rise, RTC clock source switched from LIRC32K.
[0]	**LXTFDEN**	**LXT Clock Fail/Stop Detector Enable Bit** 0 = LXT clock Fail/Stop detector Disabled. 1 = LXT clock Fail/Stop detector Enabled.

RTC Clock Frequency Detector Boundary Register (RTC_CDBR)

Register	Offset	R/W	Description	Reset Value
RTC_CDBR	RTC_BA+0x144	R/W	Clock Frequency Detector Boundary Register	0x00F0_000F

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
FAILBD							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
STOPBD							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**FAILBD**	**LXT Clock Frequency Detector Fail Boundary** The bits define the fail value of frequency monitor window. When LXT frequency monitor counter lower than FAILBD, the LXT frequency detect fail interrupt flag will set to 1. **Note:** The boundary is defined as the minimun value of LXT among 256 LIRC32K clock time.
[15:8]	**Reserved**	Reserved.
[7:0]	**STOPBD**	**LXT Clock Stop Frequency Detector Stop Boundary** The bits define the stop value of frequency monitor window. When LXT frequency monitor counter lower than STOPBD, the LXT frequency detect Stop interrupt flag will set to 1. **Note:** The boundary is defined as the maximun value of LXT among 256 LIRC32K clock time.

RTC Power Control Register (RTC_PWRCTL)

Register	Offset	R/W	Description	Reset Value
RTC_PWRCTL	RTC_BA+0x180	R/W	RTC Power Control Register (TZS)	0x0000_0000

31	30	29	28	27	26	25	24
PWRSTCLR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PWRSTCLR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ISORLS	Reserved	PWROFFTIME		PWRONTIME			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PWRKEY	PWRST	EDGETRIG	Reserved		HWPCLREN	SWPCLR	PWRON

Bits	Description	
[31:16]	**PWRSTCLR**	**Power Control Status Clear** User has to write a number (0x5aa5) to PWRSTCLR to set or clear PWRST.
[15]	**ISORLS**	**Isolation Release Indicated Bit (Read Only)** 0= The V_{BAT} domain is isolated, write to RTC register is ignored. 1= The V_{BAT} domain is not isolated. **Note:** When core power is power on(RTC_RPWR high), isolation will be hold for 10ms, user can read this bit to check.
[14]	**Reserved**	Reserved.
[13:12]	**PWROFFTIME**	**Power Off Time** This field defines the time from RTC_nRWAKE pressed to RTC_RPWR off. The acture power-off time is the PWROFFTIME+4. **Note:** The unit is second.
[11:8]	**PWRONTIME**	**Power On Time** This field defines the period that RTC_nRWAKE pressed to turn on core power (RTC_RPWR high). The period is from 1/4096 to 4 seconds calculate with $2^{(PWRONTIME -12)}$. **Note 1:** PWRONTIME = 4'b1111 is Reserved. **Note 2:** The unit is second.
[7]	**PWRKEY**	**The key status of RTC_nRWAKE.**
[6]	**PWRST**	**Power Control Status** 0 = Power Control Disabled. 1 = Power Control Enabled. **Note 1:** PWR_ST will set to high automatically once PWRKEY going low. **Note 2:** User has to write a number (0x5aa5) to PWRSTCLR to set or clear
[5]	**EDGETRIG**	**RTC_nRWAKE Pin Trigger Mode** 0 = Level trigger mode. RTC_RPWR is high when RTC_nRWAKE pressed longer than PWRONTIME. 1 = Edge trigger mode. RTC_RPWR is high when RTC_nRWAKE pressed longer than PWRONTIME and then released.

[4:3]	**Reserved**	Reserved.
[2]	**HWPCLREN**	**Hardware Power Clear Enable Bit** 0 = Hardware did not take any active on RTC_RPWR pin. 1 = Hardware set RTC_RPWR pin to low automatically after PWROFFTIME expired.
[1]	**SWPCLR**	**Software Power Clear Enable Bit** When RTC_nRWAKE is low (user pressed the key), Software can write this bit high to RTC_RPWR to be low directly. When user released the key (RTC_nRWAKE is high), the RTC will clear this bit to low automatically. 0 = No operation. 1 = Force RTC_RPWR to low.
[0]	**PWRON**	**Software Power ON Control** 0 = No operation or clear it to turn off RTC_RPWR while PWRON is set. 1 = To keep RTC_RPWR in high state when RTC_nRWAKE released.

RTC Power Control Status Register (RTC_PWRSTS)

Register	Offset	R/W	Description	Reset Value
RTC_PWRSTS	RTC_BA+0x184	R	RTC Power Control Status Register (TZS)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					PWROFFCNT		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved	PWRONCNT						
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PWRONCNT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:27]	**Reserved**	Reserved.
[26:24]	**PWROFFCNT**	**Power Off Counter Current Value (Read Only)** This field shows the current value of power off counter. This field is read only and write to this field will not take any effect.
[23]	**Reserved**	Reserved.
[22:8]	**PWRONCNT**	**Power On Counter Current Value (Read Only)** This field shows the current value of power on counter. This field is read only and write to this field will not take any effect.
[7:0]	**Reserved**	Reserved.

6.24 UART Interface Controller (UART)

6.24.1 Overview

The chip provides seventeen channels of Universal Asynchronous Receiver/Transmitters (UART). The UART controller performs serial-to-parallel conversion on data received from the peripheral and parallel-to-serial conversion on data transmitted from the CPU. Each UART controller channel supports ten types of interrupts. The UART controller supports flow control function. The UART controller also supports IrDA SIR, RS-485, and Single-wire function modes and auto-baud rate measuring function.

6.24.2 Features

- Full-duplex asynchronous communications
- Separates receive and transmit 16/16 bytes (UART0 only) or 32/32 bytes entry FIFO for data payloads
- Supports hardware auto-flow control
- Programmable receiver buffer trigger level
- Supports programmable baud rate generator for each channel individually
- Supports nCTS, incoming data, Received Data FIFO reached threshold and RS-485 Address Match (AAD mode) wake-up function
- Supports 8-bit receiver buffer time-out detection function
- Programmable transmitting data delay time between the last stop and the next start bit by setting DLY (UART_TOUT [15:8])
- Supports Auto-Baud Rate measurement and baud rate compensation function
  - Support 9600 bps for UART_CLK is selected LXT.
- Supports break error, frame error, parity error and receive/transmit buffer overflow detection function
- Fully programmable serial-interface characteristics
  - Programmable number of data bit, 5-, 6-, 7-, 8- bit character
  - Programmable parity bit, even, odd, no parity or stick parity bit generation and detection
  - Programmable stop bit, 1, 1.5, or 2 stop bit generation
- Supports IrDA SIR function mode
  - Supports for 3/16 bit duration for normal mode
- Supports RS-485 function mode
  - Supports RS-485 9-bit mode
  - Supports hardware or software enables to program nRTS pin to control RS-485 transmission direction
- Supports PDMA transfer function
- Support Single-wire function mode.

UART Feature	UART0	UART1 ~ UART16
FIFO	16 Bytes	32 Bytes
Processor	A35	M4

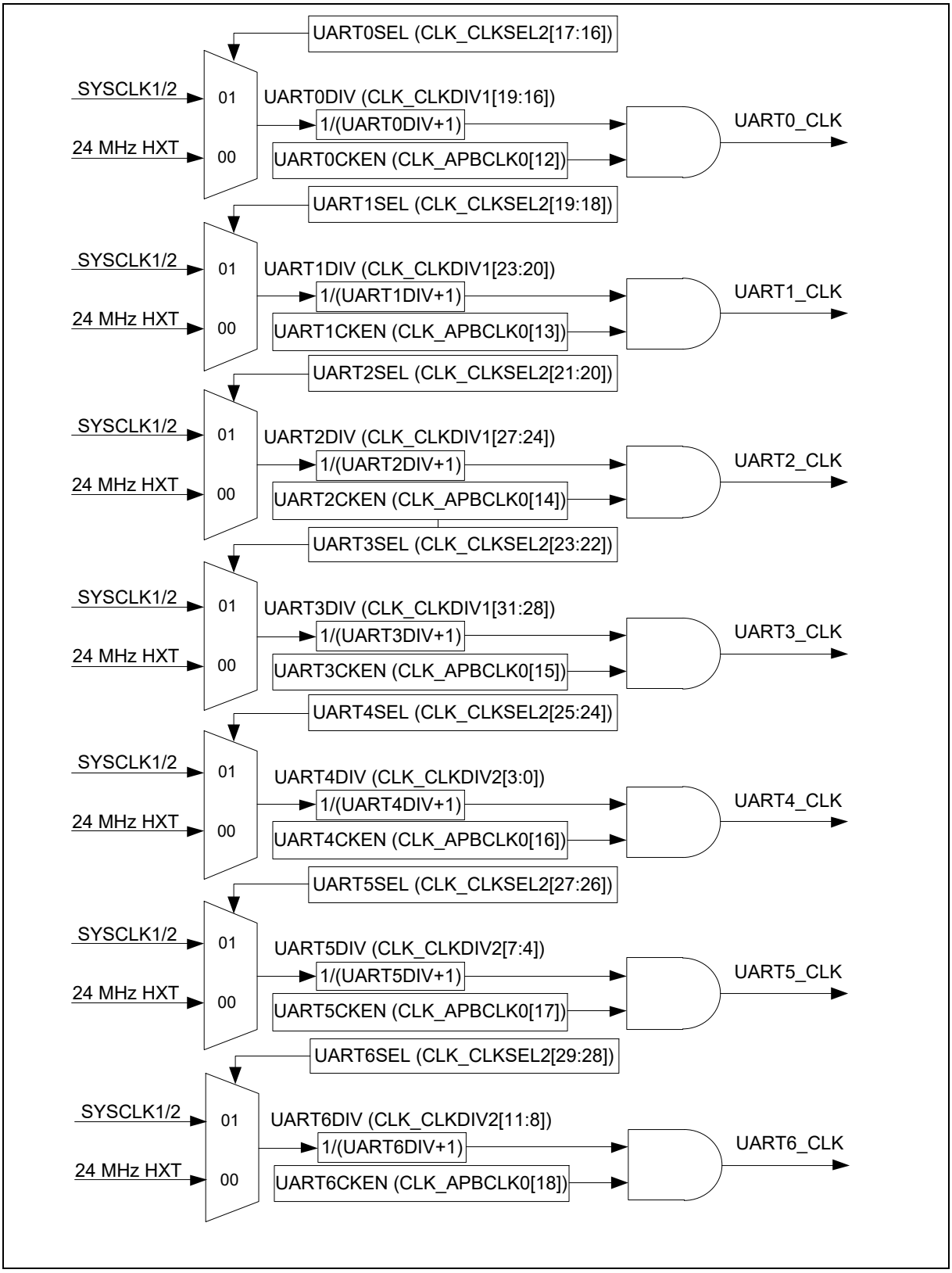
Auto Flow Control (CTS/RTS)	√	√
IrDA	√	√
RS-485 Function Mode	√	√
nCTS Wake-up	√	√
Incoming Data Wake-up	√	√
Received Data FIFO reached threshold Wake-up	√	√
RS-485 Address Match (AAD mode) Wake-up	√	√
Received Data FIFO reached threshold Time-out Wake-up	√	√
Baud Rate Compensation	√	√
Auto-Baud Rate Measurement	√	√
STOP Bit Length	1, 1.5, 2 bits	1, 1.5, 2 bits
Word Length	5, 6, 7, 8 bits	5, 6, 7, 8 bits
Even / Odd Parity	√	√
Stick Bit	√	√

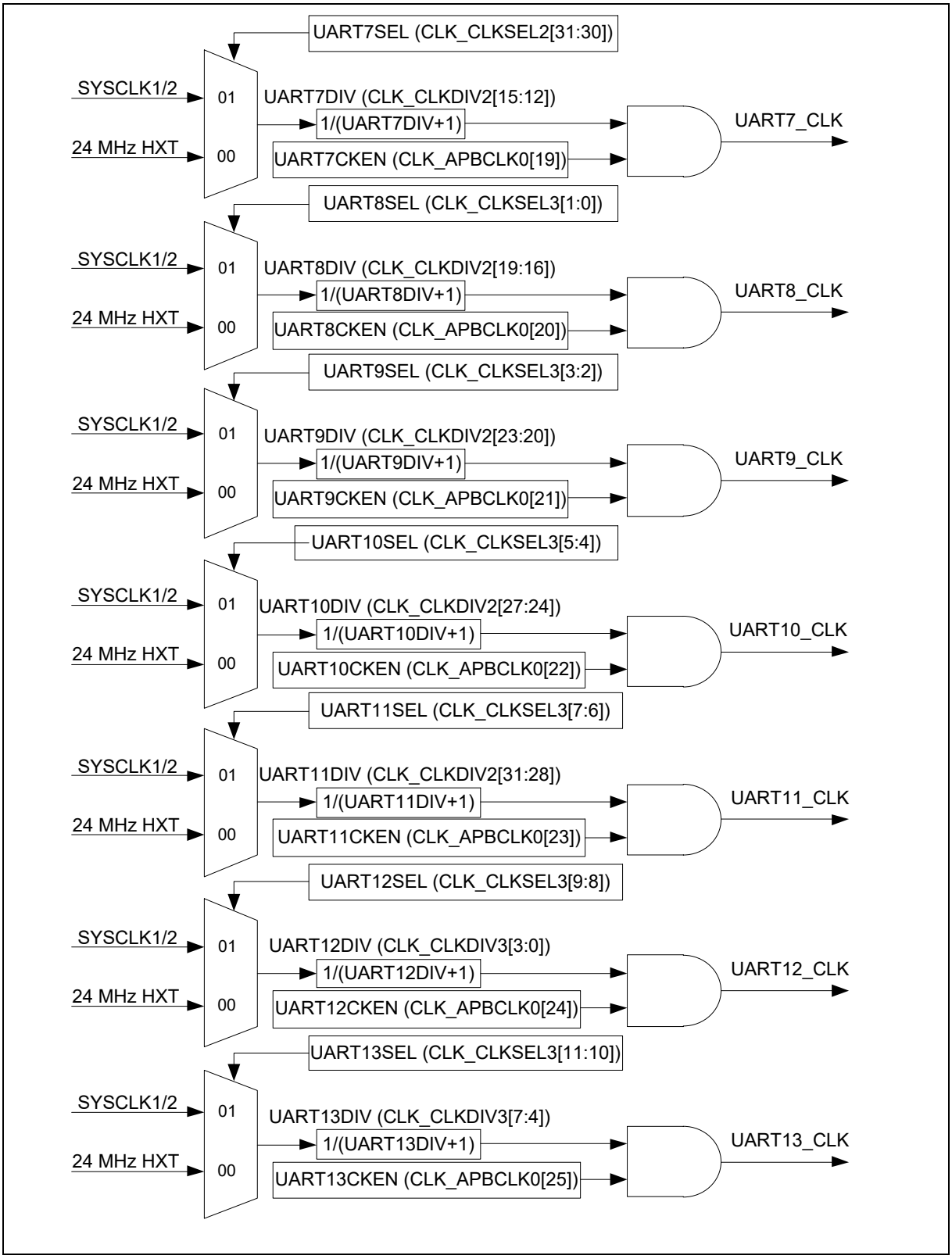
Table 6.24-1 NuMicro MA35D1 Series UART Features

6.24.3 Block Diagram

The UART clock control and block diagram are shown in Figure 6.24-1 and Figure 6.24-2 respectively.

Note: The frequency of UARTx_CLK should not be greater than 30 times HCLK.


UART0SEL (CLK_CLKSEL2[17:16])
SYSCLK1/2
24 MHz HXT
01
00
UART0DIV (CLK_CLKDIV1[19:16])
1/(UART0DIV+1)
UART0CKEN (CLK_APBCLK0[12])
UART0_CLK
UART1SEL (CLK_CLKSEL2[19:18])
UART1DIV (CLK_CLKDIV1[23:20])
1/(UART1DIV+1)
UART1CKEN (CLK_APBCLK0[13])
UART1_CLK
UART2SEL (CLK_CLKSEL2[21:20])
UART2DIV (CLK_CLKDIV1[27:24])
1/(UART2DIV+1)
UART2CKEN (CLK_APBCLK0[14])
UART2_CLK
UART3SEL (CLK_CLKSEL2[23:22])
UART3DIV (CLK_CLKDIV1[31:28])
1/(UART3DIV+1)
UART3CKEN (CLK_APBCLK0[15])
UART3_CLK
UART4SEL (CLK_CLKSEL2[25:24])
UART4DIV (CLK_CLKDIV2[3:0])
1/(UART4DIV+1)
UART4CKEN (CLK_APBCLK0[16])
UART4_CLK
UART5SEL (CLK_CLKSEL2[27:26])
UART5DIV (CLK_CLKDIV2[7:4])
1/(UART5DIV+1)
UART5CKEN (CLK_APBCLK0[17])
UART5_CLK
UART6SEL (CLK_CLKSEL2[29:28])
UART6DIV (CLK_CLKDIV2[11:8])
1/(UART6DIV+1)
UART6CKEN (CLK_APBCLK0[18])
UART6_CLK


UART7SEL (CLK_CLKSEL2[31:30])
SYSCLK1/2
24 MHz HXT
01
00
UART7DIV (CLK_CLKDIV2[15:12])
1/(UART7DIV+1)
UART7CKEN (CLK_APBCLK0[19])
UART7_CLK
UART8SEL (CLK_CLKSEL3[1:0])
UART8DIV (CLK_CLKDIV2[19:16])
1/(UART8DIV+1)
UART8CKEN (CLK_APBCLK0[20])
UART8_CLK
UART9SEL (CLK_CLKSEL3[3:2])
UART9DIV (CLK_CLKDIV2[23:20])
1/(UART9DIV+1)
UART9CKEN (CLK_APBCLK0[21])
UART9_CLK
UART10SEL (CLK_CLKSEL3[5:4])
UART10DIV (CLK_CLKDIV2[27:24])
1/(UART10DIV+1)
UART10CKEN (CLK_APBCLK0[22])
UART10_CLK
UART11SEL (CLK_CLKSEL3[7:6])
UART11DIV (CLK_CLKDIV2[31:28])
1/(UART11DIV+1)
UART11CKEN (CLK_APBCLK0[23])
UART11_CLK
UART12SEL (CLK_CLKSEL3[9:8])
UART12DIV (CLK_CLKDIV3[3:0])
1/(UART12DIV+1)
UART12CKEN (CLK_APBCLK0[24])
UART12_CLK
UART13SEL (CLK_CLKSEL3[11:10])
UART13DIV (CLK_CLKDIV3[7:4])
1/(UART13DIV+1)
UART13CKEN (CLK_APBCLK0[25])
UART13_CLK

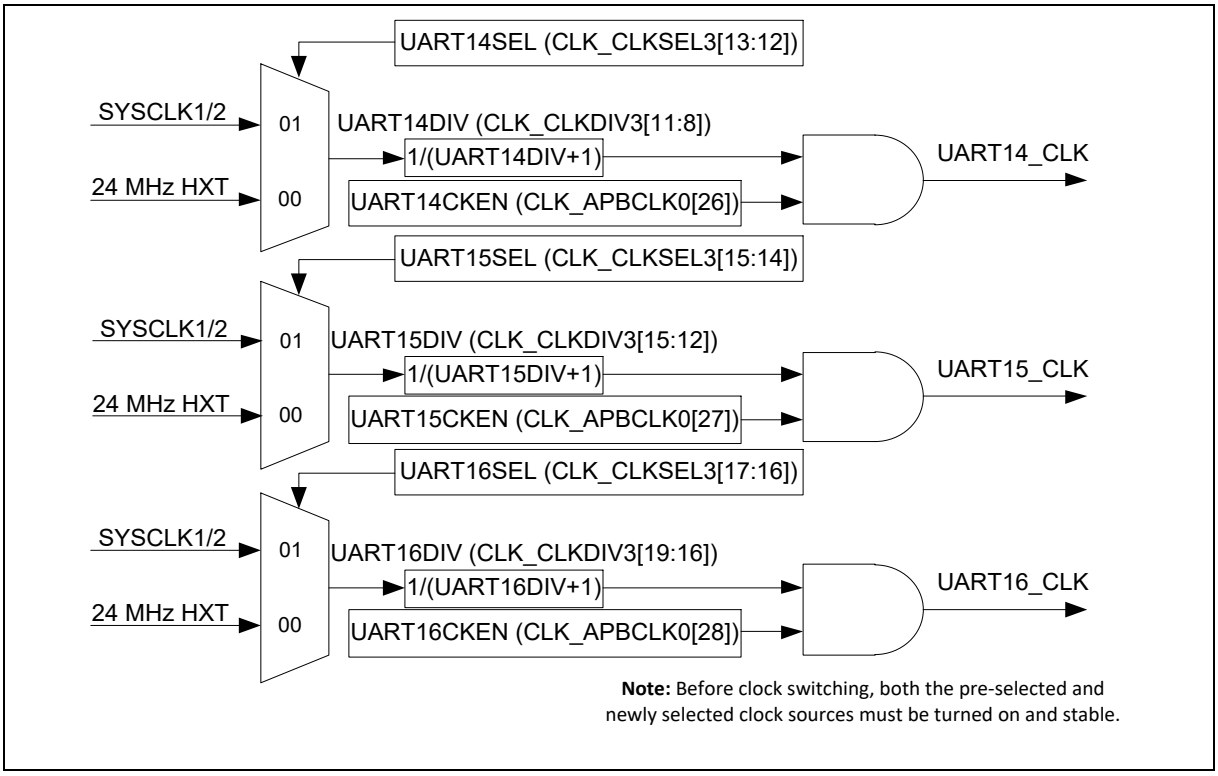


Figure 6.24-1 UART Clock Control Diagram

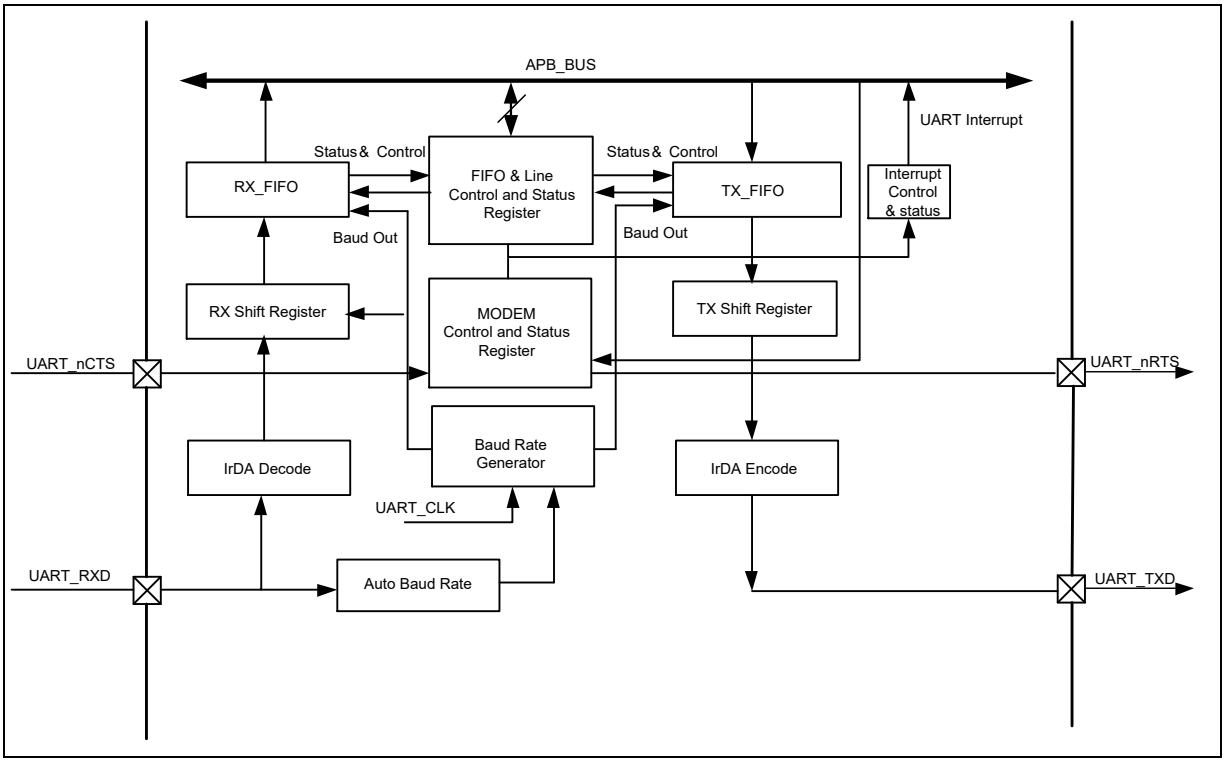


Figure 6.24-2 UART Block Diagram

Each block is described in detail as follows:

TX_FIFO

The transmitter is buffered with a 16 bytes (UART0 only) or 32 bytes FIFO to reduce the number of interrupts presented to the CPU.

RX_FIFO

The receiver is buffered with a 16 bytes (UART0 only) or 32 bytes FIFO (plus three error bits, BIF (UART_FIFOSTS[6]), FEF (UART_FIFOSTS[5]), PEF (UART_FIFOSTS[4])) to reduce the number of interrupts presented to the CPU.

TX Shift Register

This block is responsible for shifting out the transmitting data serially.

RX Shift Register

This block is responsible for shifting in the receiving data serially.

Modem Control and Status Register

This register controls the interface to the MODEM or data set (or a peripheral device emulating a MODEM).

Baud Rate Generator

Divide the external clock by the divisor to get the desired baud rate clock. Refer to baud rate equation.

IrDA Encode

This block is IrDA encoding control block.

IrDA Decode

This block is IrDA decoding control block.

FIFO and Line Control and Status Register

This field is register set that including the FIFO control register (UART_FIFO), FIFO status register (UART_FIFOSTS), and line control register (UART_LINE) for transmitter and receiver. The time-out register (UART_TOUT) identifies the condition of time-out interrupt.

Auto-Baud Rate Measurement

This block is responsible for auto-baud rate measurement.

Interrupt Control and Status Register

There are ten types of interrupts. Interrupt enable register (UART_INTEN) enable or disable the responding interrupt and interrupt status register (UART_INTSTS) identifying the occurrence of the responding interrupt.

Interrupt	Description
RDAINT	Receive Data Available Interrupt.
THREINT	Transmit Holding Register Empty Interrupt.
TXENDINT	Transmitter Empty Interrupt.
RLSINT	Receive Line Status Interrupt (parity error or frame error or break error).
MODEMINT	MODEM Status Interrupt.
RXTOINT	Receiver Buffer Time-out Interrupt.
BUFERRINT	Buffer Error Interrupt.
WKINT	Wake-up Interrupt.
ABRINT	Auto-Baud Rate Interrupt.

SWBEINT	Single-wire Bit Error Detect Interrupt.

Table 6.24-2 UART Interrupt

6.24.4 Basic Configuration

6.24.4.1 UART0 Basic Configuration

- Clock Source Configuration
  - Select the source of UART0 peripheral clock on UART0SEL (CLK_CLKSEL2[17:16]).
  - Select the clock divider number of UART0 peripheral clock on UART0DIV (CLK_CLKDIV1[19:16]).
  - Enable UART0 peripheral clock in UART0CKEN (CLK_APBCLK0[12]).
- Reset Configuration
  - Reset UART0 controller in UART0RST (SYS_IPRST1[16]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
UART0	UART0_RXD	PE.15*	MFP1
	UART0_TXD	PE.14	MFP1
Note *: This GPIO PE.15 (UART0_RXD) pin must be pulled to high level through an external 10kΩ resistor or an internal pull-up resistor in the external device (such as a transceiver or MCU).			

6.24.4.2 UART1 Basic Configuration

- Clock Source Configuration
  - Select the source of UART1 peripheral clock on UART1SEL (CLK_CLKSEL2[19:18]).
  - Select the clock divider number of UART1 peripheral clock on UART1DIV (CLK_CLKDIV1[23:20]).
  - Enable UART1 peripheral clock in UART1CKEN (CLK_APBCLK0[13]).
- Reset Configuration
  - Reset UART1 controller in UART1RST (SYS_IPRST1[17]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
UART1	UART1_RXD	PA.2	MFP2
		PJ.12	MFP3
		PD.6	MFP2
		PL.4	MFP3
		PK.14	MFP2
		PF.0	MFP3
		PB.8	MFP3
	UART1_TXD	PA.3	MFP2

		PJ.13	MFP3
		PD.7	MFP2
		PL.5	MFP3
		PK.15	MFP2
		PF.1	MFP3
		PB.9	MFP3
	UART1_nCTS	PA.0	MFP2
		PD.4	MFP2
		PK.12	MFP2
	UART1_nRTS	PA.1	MFP2
		PD.5	MFP2
		PK.13	MFP2

6.24.4.3 UART2 Basic Configuration

- Clock Source Configuration
  - Select the source of UART2 peripheral clock on UART2SEL (CLK_CLKSEL2[21:20]).
  - Select clock divider number of UART2 peripheral clock on UART2DIV (CLK_CLKDIV1[27:24]).
  - Enable UART2 peripheral clock in UART2CKEN (CLK_APBCLK0[14]).
- Reset Configuration
  - Reset UART2 controller in UART2RST (SYS_IPRST1[18]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
UART2	UART2_RXD	PA.4	MFP3
		PJ.14	MFP2
		PL.6	MFP2
		PD.4	MFP3
		PF.2	MFP2
		PB.10	MFP2
	UART2_TXD	PA.5	MFP3
		PJ.15	MFP2
		PL.7	MFP2
		PD.5	MFP3
		PF.3	MFP2
		PB.11	MFP2
	UART2_nCTS	PJ.12	MFP2

		PL.4	MFP2
		PF.0	MFP2
		PB.8	MFP2
	UART2_nRTS	PJ.13	MFP2
		PL.5	MFP2
		PF.1	MFP2
		PB.9	MFP2

6.24.4.4 UART3 Basic Configuration

- Clock Source Configuration
  - Select the source of UART3 peripheral clock on UART3SEL (CLK_CLKSEL2[23:22]).
  - Select the clock divider number of UART3 peripheral clock on UART3DIV (CLK_CLKDIV1[31:28]).
  - Enable UART3 peripheral clock in UART3CKEN (CLK_APBCLK0[15]).
- Reset Configuration
  - Reset UART3 controller in UART3RST (SYS_IPRST1[19]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
UART3	UART3_RXD	PA.6	MFP2
		PI.8	MFP3
		PD.2	MFP2
		PE.4	MFP3
		PB.12	MFP3
	UART3_TXD	PA.7	MFP2
		PI.9	MFP3
		PD.3	MFP2
		PE.5	MFP3
		PB.13	MFP3
	UART3_nCTS	PA.4	MFP2
		PD.0	MFP2
	UART3_nRTS	PA.5	MFP2
		PD.1	MFP2

6.24.4.5 UART4 Basic Configuration

- Clock Source Configuration
  - Select the source of UART4 peripheral clock on UART4SEL (CLK_CLKSEL2[25:24]).
  - Select the clock divider number of UART4 peripheral clock on UART4DIV (CLK_CLKDIV2[3:0]).

- Enable UART4 peripheral clock in UART4CKEN (CLK_APBCLK0[16]).

- Reset Configuration
  - Reset UART4 controller in UART4RST (SYS_IPRST1[20]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
UART4	UART4_RXD	PB.14	MFP2
		PA.8	MFP3
		PI.10	MFP2
		PD.0	MFP3
		PE.6	MFP2
	UART4_TXD	PB.15	MFP2
		PA.9	MFP3
		PI.11	MFP2
		PD.1	MFP3
		PE.7	MFP2
	UART4_nCTS	PI.8	MFP2
		PE.4	MFP2
		PB.12	MFP2
	UART4_nRTS	PI.9	MFP2
		PE.5	MFP2
		PB.13	MFP2

6.24.4.6 UART5 Basic Configuration

- Clock Source Configuration
  - Select the source of UART5 peripheral clock on UART5SEL (CLK_CLKSEL2[27:26]).
  - Select the clock divider number of UART5 peripheral clock on UART5DIV (CLK_CLKDIV2[7:4]).
  - Enable UART5 peripheral clock in UART5CKEN (CLK_ APBCLK0[17]).
- Reset Configuration
  - Reset UART5 controller in UART5RST (SYS_IPRST1[21]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
UART5	UART5_RXD	PA.10	MFP2
		PG.6	MFP2
		PI.12	MFP3
	UART5_TXD	PA.11	MFP2

		PG.7	MFP2
		PI.13	MFP3
	UART5_nCTS	PA.8	MFP2
		PG.4	MFP2
	UART5_nRTS	PA.9	MFP2
		PG.5	MFP2

6.24.4.7 UART6 Basic Configuration

- Clock Source Configuration
  - Select the source of UART6 peripheral clock on UART6SEL (CLK_CLKSEL2[29:28]).
  - Select the clock divider number of UART6 peripheral clock on UART6DIV (CLK_CLKDIV2[11:8]).
  - Enable UART6 peripheral clock in UART6CKEN (CLK_ APBCLK0[18]).
- Reset Configuration
  - Reset UART6 controller in UART6RST (SYS_IPRST1[22]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
UART6	UART6_RXD	PA.15	MFP3
		PG.4	MFP3
		PI.14	MFP2
		PN.14	MFP2
	UART6_TXD	PG.1	MFP3
		PG.5	MFP3
		PI.15	MFP2
		PN.15	MFP2
	UART6_nCTS	PI.12	MFP2
		PN.12	MFP2
	UART6_nRTS	PI.13	MFP2
		PN.13	MFP2

6.24.4.8 UART7 Basic Configuration

- Clock Source Configuration
  - Select the source of UART7 peripheral clock on UART7SEL (CLK_CLKSEL2[31:30]).
  - Select the clock divider number of UART7 peripheral clock on UART7DIV (CLK_CLKDIV2[15:12]).
  - Enable UART7 peripheral clock in UART7CKEN (CLK_ APBCLK0[19]).
- Reset Configuration
  - Reset UART7 controller in UART7RST (SYS_IPRST1[23]).

- Pin Configuration

Group	Pin Name	GPIO	MFP
UART7	UART7_RXD	PL.14	MFP2
		PA.14	MFP2
		PJ.0	MFP3
		PH.0	MFP3
	UART7_TXD	PL.15	MFP2
		PG.0	MFP2
		PJ.1	MFP3
		PH.1	MFP3
	UART7_nCTS	PL.12	MFP2
		PA.12	MFP2
	UART7_nRTS	PL.13	MFP2
		PA.13	MFP2

6.24.4.9 UART8 Basic Configuration

- Clock Source Configuration
  - Select the source of UART8 peripheral clock on UART8SEL (CLK_CLKSEL3[1:0]).
  - Select the clock divider number of UART8 peripheral clock on UART8DIV (CLK_CLKDIV2[19:16]).
  - Enable UART8 peripheral clock in UART8CKEN (CLK_ APBCLK0[20]).
- Reset Configuration
  - Reset UART8 controller in UART8RST (SYS_IPRST3[8]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
UART8	UART8_RXD	PA.12	MFP3
		PJ.2	MFP2
		PH.2	MFP2
		PE.0	MFP3
	UART8_TXD	PA.13	MFP3
		PJ.3	MFP2
		PH.3	MFP2
		PE.1	MFP3
	UART8_nCTS	PJ.0	MFP2
		PH.0	MFP2
	UART8_nRTS	PJ.1	MFP2

		PH.1	MFP2

6.24.4.10 UART9 Basic Configuration

- Clock Source Configuration
  - Select the source of UART9 peripheral clock on UART9SEL (CLK_CLKSEL3[3:2]).
  - Select the clock divider number of UART9 peripheral clock on UART9DIV (CLK_CLKDIV2[23:20]).
  - Enable UART9 peripheral clock in UART9CKEN (CLK_ APBCLK0[21]).
- Reset Configuration
  - Reset UART9 controller in UART9RST (SYS_IPRST3[9]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
UART9	UART9_RXD	PG.2	MFP2
		PH.4	MFP3
		PE.2	MFP2
	UART9_TXD	PG.3	MFP2
		PH.5	MFP3
		PE.3	MFP2
	UART9_nCTS	PA.15	MFP2
		PE.0	MFP2
	UART9_nRTS	PG.1	MFP2
		PE.1	MFP2

6.24.4.11 UART10 Basic Configuration

- Clock Source Configuration
  - Select the source of UART10 peripheral clock on UART10SEL (CLK_CLKSEL3[5:4]).
  - Select the clock divider number of UART10 peripheral clock on UART10DIV (CLK_CLKDIV2[27:24]).
  - Enable UART10 peripheral clock in UART10CKEN (CLK_ APBCLK0[22]).
- Reset Configuration
  - Reset UART10 controller in UART10RST (SYS_IPRST3[10]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
UART10	UART10_RXD	PD.12	MFP3
		PM.14	MFP2
		PL.0	MFP3
		PH.6	MFP2

		PF.4	MFP3
	UART10_TXD	PD.13	MFP3
		PM.15	MFP2
		PL.1	MFP3
		PH.7	MFP2
		PF.5	MFP3
	UART10_nCTS	PM.12	MFP2
		PH.4	MFP2
	UART10_nRTS	PM.13	MFP2
		PH.5	MFP2

6.24.4.12 UART11 Basic Configuration

- Clock Source Configuration
  - Select the source of UART11 peripheral clock on UART11SEL (CLK_CLKSEL3[7:6]).
  - Select the clock divider number of UART11 peripheral clock on UART11DIV (CLK_CLKDIV2[31:28]).
  - Enable UART11 peripheral clock in UART11CKEN (CLK_ APBCLK0[23]).
- Reset Configuration
  - Reset UART11 controller in UART11RST (SYS_IPRST3[11]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
UART11	UART11_RXD	PD.13	MFP2
		PM.12	MFP4
		PI.0	MFP3
		PL.2	MFP2
		PC.12	MFP3
		PF.6	MFP2
	UART11_TXD	PD.12	MFP2
		PM.13	MFP4
		PI.1	MFP3
		PL.3	MFP2
		PC.13	MFP3
		PF.7	MFP2
	UART11_nCTS	PD.14	MFP2
		PL.0	MFP2
		PF.4	MFP2

	UART11_nRTS	PD.15	MFP2
		PL.1	MFP2
		PF.5	MFP2

6.24.4.13 UART12 Basic Configuration

- Clock Source Configuration
  - Select the source of UART12 peripheral clock on UART12SEL (CLK_CLKSEL3[9:8]).
  - Select the clock divider number of UART12 peripheral clock on UART12DIV (CLK_CLKDIV3[3:0]).
  - Enable UART12 peripheral clock in UART12CKEN (CLK_ APBCLK0[24]).
- Reset Configuration
  - Reset UART12 controller in UART12RST (SYS_IPRST3[12]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
UART12	UART12_RXD	PI.2	MFP2
		PG.8	MFP2
		PK.6	MFP2
		PC.14	MFP2
		PN.12	MFP3
		PE.8	MFP3
	UART12_TXD	PI.3	MFP2
		PG.9	MFP2
		PK.7	MFP2
		PC.15	MFP2
		PN.13	MFP3
		PE.9	MFP3
	UART12_nCTS	PI.0	MFP2
		PK.4	MFP2
		PC.12	MFP2
	UART12_nRTS	PI.1	MFP2
		PG.10	MFP2
		PK.5	MFP2
		PC.13	MFP2

6.24.4.14 UART13 Basic Configuration

- Clock Source Configuration
  - Select the source of UART13 peripheral clock on UART13SEL

(CLK_CLKSEL3[11:10]).

- Select the clock divider number of UART13 peripheral clock on UART13DIV (CLK_CLKDIV3[7:4]).
- Enable UART13 peripheral clock in UART13CKEN (CLK_ APBCLK0[25]).

- Reset Configuration
  - Reset UART13 controller in UART13RST (SYS_IPRST3[13]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
UART13	UART13_RXD	PI.4	MFP3
		PL.8	MFP3
		PK.4	MFP3
		PH.12	MFP3
		PK.12	MFP3
		PF.8	MFP2
	UART13_TXD	PI.5	MFP3
		PL.9	MFP3
		PG.10	MFP3
		PK.5	MFP3
		PH.13	MFP3
		PK.13	MFP3
		PF.9	MFP2
	UART13_nCTS	PE.8	MFP2
		PF.10	MFP2
	UART13_nRTS	PE.9	MFP2
		PF.11	MFP2

6.24.4.15 UART14 Basic Configuration

- Clock Source Configuration
  - Select the source of UART14 peripheral clock on UART14SEL (CLK_CLKSEL3[13:12]).
  - Select the clock divider number of UART14 peripheral clock on UART14DIV (CLK_CLKDIV3[11:8]).
  - Enable UART14 peripheral clock in UART14CKEN (CLK_ APBCLK0[26]).
- Reset Configuration
  - Reset UART14 controller in UART14RST (SYS_IPRST3[14]).
- Pin Configuration

Group	Pin Name	GPIO	MFP

UART14	UART14_RXD	PL.12	MFP4
		PI.6	MFP2
		PL.10	MFP2
		PH.14	MFP2
		PE.10	MFP3
	UART14_TXD	PL.13	MFP4
		PI.7	MFP2
		PL.11	MFP2
		PH.15	MFP2
		PE.11	MFP3
	UART14_nCTS	PI.4	MFP2
		PL.8	MFP2
		PH.12	MFP2
	UART14_nRTS	PI.5	MFP2
		PL.9	MFP2
		PH.13	MFP2

6.24.4.16 UART15 Basic Configuration

- Clock Source Configuration
  - Select the source of UART15 peripheral clock on UART15SEL (CLK_CLKSEL3[15:14]).
  - Select the clock divider number of UART15 peripheral clock on UART15DIV (CLK_CLKDIV3[15:12]).
  - Enable UART15 peripheral clock in UART15CKEN (CLK_ APBCLK0[27]).
- Reset Configuration
  - Reset UART15 controller in UART15RST (SYS_IPRST3[15]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
UART15	UART15_RXD	PK.0	MFP3
		PD.8	MFP3
		PE.12	MFP2
	UART15_TXD	PK.1	MFP3
		PD.9	MFP3
		PE.13	MFP2
	UART15_nCTS	PE.10	MFP2
	UART15_nRTS	PE.11	MFP2

6.24.4.17 UART16 Basic Configuration

- Clock Source Configuration
  - Select the source of UART16 peripheral clock on UART16SEL (CLK_CLKSEL3[17:16]).
  - Select the clock divider number of UART16 peripheral clock on UART16DIV (CLK_CLKDIV3[19:16]).
  - Enable UART16 peripheral clock in UART16CKEN (CLK_ APBCLK0[28]).
- Reset Configuration
  - Reset UART16 controller in UART16RST (SYS_IPRST3[16]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
UART16	UART16_RXD	PA.0	MFP3
		PK.2	MFP2
		PD.10	MFP2
	UART16_TXD	PA.1	MFP3
		PK.3	MFP2
		PD.11	MFP2
	UART16_nCTS	PK.0	MFP2
		PD.8	MFP2
	UART16_nRTS	PK.1	MFP2
		PD.9	MFP2

UART Interface Controller Pin description is shown in Table 6.24-3:

Pin	Type	Description
UARTx_TXD	Output	UARTx transmit
UARTx_RXD	Input	UARTx receive
UARTx_nCTS	Input	UARTx modem clear to send
UARTx_nRTS	Output	UARTx modem request to send

Table 6.24-3 UART Interface Controller Pin

6.24.5 Functional Description

The UART controller supports four function modes including UART, IrDA, RS-485, and Single-wire mode. User can select a function by setting the UART_FUNCSEL register. The four function modes will be described in following section.

6.24.5.1 UART Controller Baud Rate Generator

The UART controller includes a programmable baud rate generator capable of dividing clock input by divisors to produce the serial clock that transmitter and receiver need. Table 6.24-4 list the UART baud rate equations in the various conditions. Table 6.24-5 and Table 6.24-6 list the UART baud rate parameter and register setting example. In IrDA function mode, the baud rate generator must be set in mode 0. More detailed register description is shown in UART_BAUD register. There are three setting

modes. Mode 0 is set by UART_BAUD[29:28] with 00. Mode 1 is set by UART_BAUD[29:28] with 10. Mode 2 is set by UART_BAUD[29:28] with 11.

Mode	BAUDM1	BAUDM0	Baud Rate Equation
Mode 0	0	0	UART_CLK / [16 * (BRD+2)].
Mode 1	1	0	UART_CLK / [(EDIVM1+1) * (BRD+2)], EDIVM1 must >= 8.
Mode 2	1	1	UART_CLK / (BRD+2). If UART_CLK <= 3*HCLK, BRD must >= 8. If UART_CLK > 3*HCLK, BRD must >= 3*N – 1. N is the smallest integer larger than or equal to the ratio of UART_CLK /HCLK. For example, if 3*HCLK < UART_CLK =< 4*HCLK, BRD must >=11. if 4*HCLK < UART_CLK =< 5*HCLK, BRD must >=14. (If the UART_CLK is selected from LXT, BRD can be greater than or equal to 1)

Table 6.24-4 UART controller Baud Rate Equation Table

UART Peripheral Clock = 12 MHz			
Baud Rate	Mode 0	Mode 1	Mode 2
921600	Not support	Not recommended	BRD=11
460800	Not recommended	BRD=0, EDIVM1 =12	BRD=24
230400	Not recommended	BRD =2, EDIVM1 =12	BRD =50
115200	Not recommended	BRD =6, EDIVM1 =12	BRD =102
57600	BRD =11	BRD =14, EDIVM1 =12	BRD =206
38400	BRD =18	BRD =22, EDIVM1 =12	BRD =311
19200	BRD =37	BRD =61, EDIVM1 =9	BRD =623
9600	BRD =76	BRD =123, EDIVM1 =9	BRD =1248
4800	BRD =154	BRD =248, EDIVM1 =9	BRD =2498

Table 6.24-5 UART controller Baud Rate Parameter Setting Example Table

UART Peripheral Clock = 12 MHz			
Baud Rate	UART_BAUD Value		
	Mode 0	Mode 1	Mode 2
921600	Not support	Not recommended	0x3000_000B
460800	Not recommended	0x2C00_0000	0x3000_0018
230400	Not recommended	0x2C00_0002	0x3000_0032
115200	Not recommended	0x2C00_0006	0x3000_0066
57600	0x0000_000B	0x2C00_000E	0x3000_00CE
38400	0x0000_0012	0x2C00_0016	0x3000_0137

19200	0x0000_0025	0x2900_003D	0x3000_026F
9600	0x0000_004C	0x2900_007B	0x3000_04E0
4800	0x0000_009A	0x2900_00F8	0x3000_09C2

Table 6.24-6 UART controller Baud Rate Register Setting Example Table

6.24.5.2 UART Controller Baud Rate Compensation

The UART controller supports baud rate compensation function. It is used to optimize the precision in each bit. The precision of the compensation is half of UART module clock because there is BRCOMPDEC (UART_BRCOMP[31]) to define the positive or negative compensation in each bit. If the BRCOMPDEC (UART_BRCOMP[31]) = 0, it is positive compensation for each bit, one more module clock will be append in the compensated bit. If the BRCOMPDEC (UART_BRCOMP[31]) = 1, it is negative compensation for each bit, decrease one module clock in the compensated bit.

There is 9-bits location, BRCOMP[8:0] (UART_BRCOMP[8:0]), can be configured by user to define the relative bit is compensated or not. BRCOMP[7:0] is used to define the compensation of DAT (UART_DAT[7:0]) and BRCOMP[8] is used to define PARITY (UART_DAT[8]).

Example:

1. UART's peripheral clock = 32.768 kHz and baud rate is 9600 bps

Baud rate is 9600 bps, UART peripheral clock is 32.768 kHz → 3.413 peripheral clock/bit

if the baud divider is set 1 (3 peripheral clock/bit), the inaccuracy of each bit is -0.413 peripheral clock and BRCOMPDEC (UART_BRCOMP[31]) = 0, so that the BRCOMP (UART_BRCOMP[8:0]) can be set as 9'b010100101 = 0xa5.

Bit	Name	Total INACCURACY	BRCOMP Compensated	Final Inaccuracy
0	Start	-0.413	x	-0.413
1	UART_DAT[0]	-0.826(-0.413-0.413)	1	0.174
2	UART_DAT[1]	-0.239(0.174-0.413)	0	-0.239
3	UART_DAT[2]	-0.652(-0.239-0.413)	1	0.348
4	UART_DAT[3]	-0.065(0.348-0.413)	0	-0.065
5	UART_DAT[4]	-0.478(-0.065-0.413)	0	-0.478
6	UART_DAT[5]	-0.891(-0.478-0.413)	1	0.109
7	UART_DAT[6]	-0.304(0.109-0.413)	0	-0.304
8	UART_DAT[7]	-0.717(-0.304-0.413)	1	0.283
9	Parity	-0.130(0.283-0.413)	0	-0.13

Table 6.24-7 Baud Rate Compensation Example Table 1

2. UART's peripheral clock = 32.768 kHz and baud rate is 4800 bps

Baud rate is 4800 bps, UART peripheral clock is 32.768 kHz → 6.827 peripheral clock/bit

if the baud divider is set 5 (7 peripheral clock/bit), the inaccuracy of each bit is 0.173 peripheral clock and BRCOMPDEC (UART_BRCOMP[31]) = 1, so that the BRCOMP (UART_BRCOMP[8:0]) can be set as 9'b010000010 = 0x82.

Bit	Name	Total INACCURACY	BRCOMP	Final Inaccuracy

			Compensated	
0	Start	0.173	x	0.173
1	UART_DAT[0]	0.346(0.173+0.173)	0	0.346
2	UART_DAT[1]	0.519(0.346+0.173)	1	-0.481
3	UART_DAT[2]	-0.308(-0.481+0.173)	0	-0.308
4	UART_DAT[3]	-0.135(-0.308+0.173)	0	-0.135
5	UART_DAT[4]	-0.038(-0.135+0.173)	0	0.038
6	UART_DAT[5]	0.211(0.038+0.173)	0	0.211
7	UART_DAT[6]	0.384(0.211+0.173)	0	0.384
8	UART_DAT[7]	0.557(0.384+0.173)	1	-0.443
9	Parity	-0.270(-0.443+0.173)	0	-0.270

Table 6.24-8 Baud Rate Compensation Example Table 2

UART Controller Auto-Baud Rate Function Mode

Auto-Baud Rate function can measure baud rate of receiving data from UART RX pin automatically. When the Auto-Baud Rate measurement is finished, the measuring baud rate is loaded to BRD (UART_BAUD[15:0]). Both of the BAUDM1 (UART_BAUD[29]) and BAUDM0 (UART_BAUD[28]) are set to 1 automatically. UART RX data from Start bit to 1st rising edge time is set by $2^{ABRDBITS}$ bit time in Auto-Baud Rate function detection frame.

$2^{ABRDBITS}$ bit time from START bit to the 1st rising edge is calculated by setting ABRDBITS (UART_ALTCTL[20:19]). Set ABRDEN (UART_ALTCTL[18]) to enable auto-baud rate function. In the beginning when no data transfer, the UART RX is held high. Once the falling edge is detected, START bit is received. The auto-baud rate counter will be reset and then start counting. The auto-baud rate counter will be stop when the 1st rising edge is detected. Then, the auto-baud rate counter value divided by ABRDBITS (UART_ALTCTL[20:19]) is loaded to BRD (UART_BAUD[15:0]) automatically. ABRDEN (UART_ALTCTL[18]) is cleared. The Auto-Baud measurement is shown in Figure 6.24-3. Once the auto-baud rate measurement is finished, the ABRDIF (UART_FIFOSTS[1]) is set. When auto-baud rate counter is overflow, ABRDTOIF (UART_FIFOSTS[2]) is set. If ABRDIF (UART_FIFOSTS[1]) or ABRDTOIF (UART_FIFOSTS[2]) is set, the auto-baud rate flag ABRIF (UART_ALTCTL[17]) is generated. If ABRIEN (UART_INTEN[18]) is enabled, the auto-baud rate interrupt ABRINT (UART_INTSTS[31]) is generated when ABRIF (UART_ALTCTL[17]) is set.

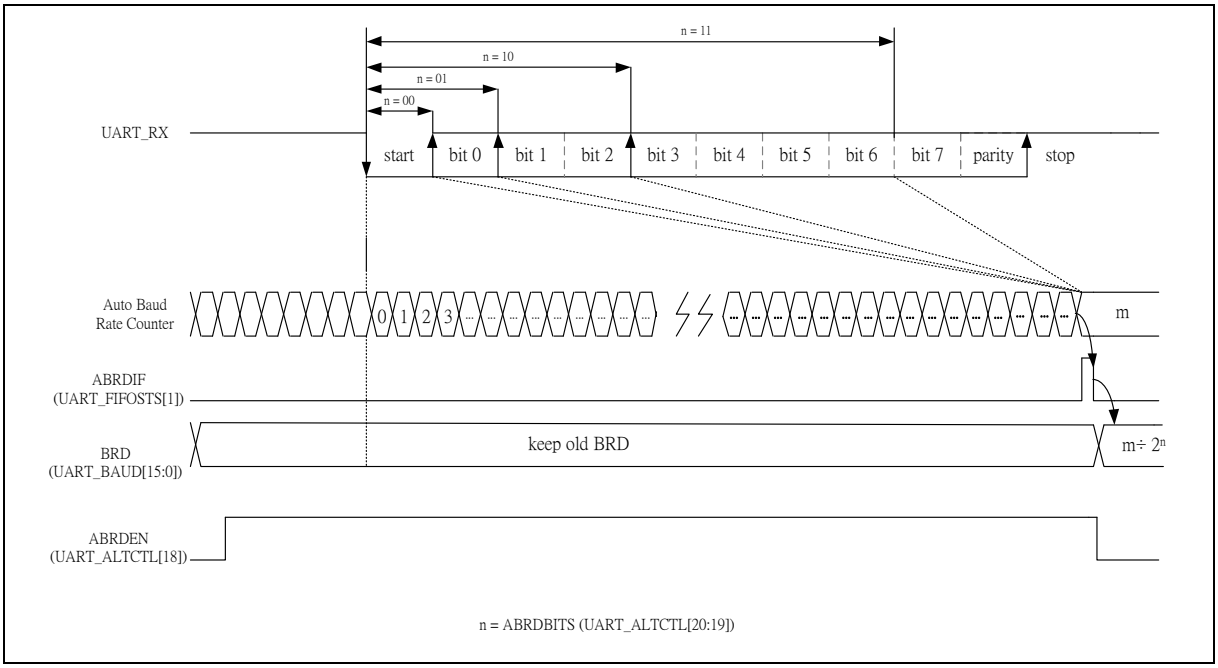


Figure 6.24-3 Auto-Baud Rate Measurement

6.24.5.3 Programming Sequence Example:

1. Program ABRDBITS (UART_ALTCTL[20:19]) to determines UART RX data 1st rising edge time from Start by $2^{ABRDBITS}$ bit time.
2. Set ABRIEN (UART_INTEN[18]) to enable auto-baud rate function interrupt.
3. Set ABRDEN (UART_ALTCTL[18]) to enable auto-baud rate function.
4. ABRDIF (UART_FIFOSTS[1]) is set, the auto-baud rate measurement is finished.
5. Operate UART transmit and receive action.
6. ABRDTOIF (UART_FIFOSTS[2]) is set, if auto-baud rate counter is overflow.
7. Go to Step 3.

6.24.5.4 UART Controller Transmit Delay Time Value

The UART controller programs DLY (UART_TOUT [15:8]) to control the transfer delay time between the last stop bit and next start bit in transmission. The unit is baud. The operation is shown in Figure 6.24-4.

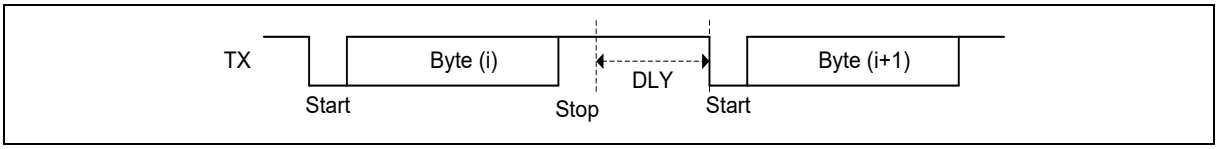


Figure 6.24-4 Transmit Delay Time Operation

6.24.5.5 UART Controller FIFO Control and Status

The UART controller is built-in with a 32 bytes transmitter FIFO (TX_FIFO) and a 32 bytes receiver FIFO (RX_FIFO) that reduces the number of interrupts presented to the CPU. The CPU can read the status of the UART at any time during operation. The reported status information includes condition of the transfer operations being performed by the UART, as well as 3 error conditions (parity error, framing error, break interrupt) occur if receiving data has parity, frame or break error. UART, IrDA and RS-485 mode support FIFO control and status function.

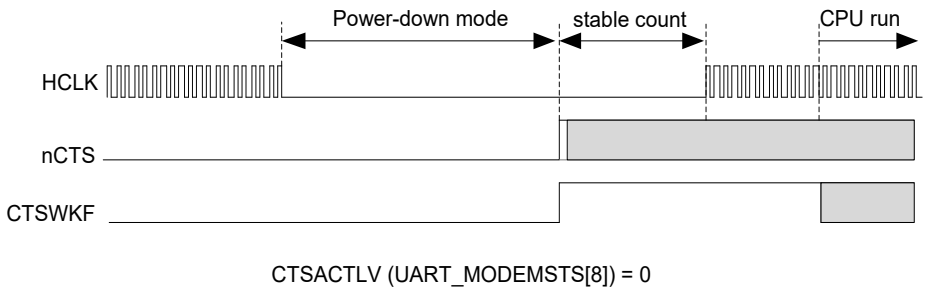
6.24.5.6 UART Controller Wake-up Function

The UART controller supports wake-up system function. The wake-up function includes nCTS pin, incoming data wake-up, Received Data FIFO reached threshold wake-up, RS-485 Address Match (AAD mode) wake-up and Received Data FIFO threshold time-out wake-up function. CTSWKF (UART_WKSTS[0]), DATWKF (UART_WKSTS[1]), RFRTWKF (UART_WKSTS[2]), RS485WKF (UART_WKSTS[3]) or TOUTWKF (UART_WKSTS[4]) cause the wake-up interrupt flag WKIF(UART_INTSTS[6]) generated. If the WKIEN (UART_INTEN[6]) is enabled, the wake-up interrupt flag WKIF(UART_INTSTS[6]) cause the wake-up interrupt WKINT (UART_INTSTS[14]) generated.

nCTS pin wake-up :

When the system is in Power-down mode and WKCTSEN (UART_WKCTL[0]) is set, the toggle of nCTS pin can wake-up system. If the WKCTSEN (UART_WKCTL[0]) is enabled, the toggle of nCTS pin cause the nCTS wake-up flag CTSWKF (UART_WKSTS[0]) generated. The nCTS wake-up is shown in Figure 6.24-5 and Figure 6.24-6.

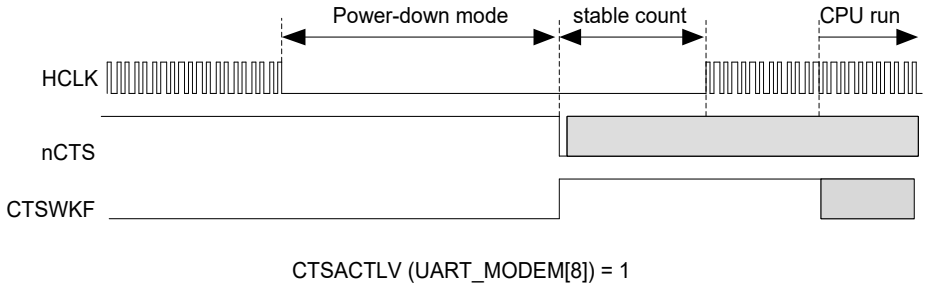
nCTS Wake-up Case 1 (nCTS transition from low to high)



Note: Stable count means HCLK source recovery stable count.

Figure 6.24-5 UART nCTS Wake-up Case1

nCTS Wake-up Case 2 (nCTS transition from high to low)



Note: Stable count means HCLK source recovery stable count.

Figure 6.24-6 UART nCTS Wake-up Case2

Incoming Data Wake-up

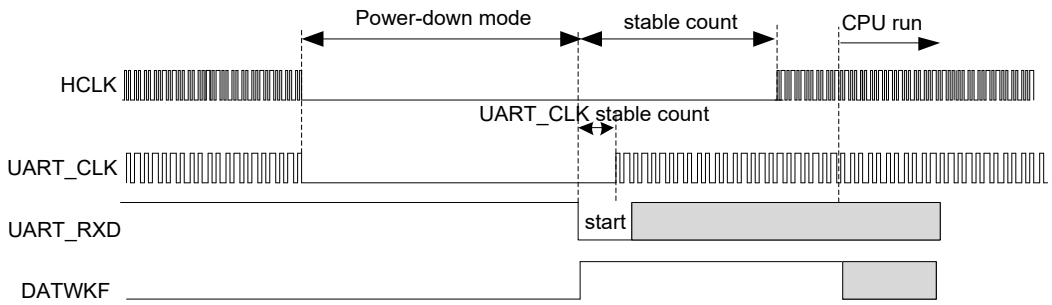
When system is in Power-down mode and the WKDATEN (UART_WKCTL [1]) is set, the toggle of incoming data (UART_RXD) pin can wake-up the system. In order to receive the incoming data after the system wake-up, the STCOMP (UART_DWKCOMP[15:0]) shall be set. These bits field of STCOMP indicate how many clock cycle selected by UART_CLK do the UART controller can get the 1st bit (start bit) when the system is wakeup from Power-down mode.

When incoming data wakes system up, the incoming data will be received and stored in FIFO. If the WKDATEN (UART_WKCTL[1]) is enabled, the toggle of incoming data (UART_RXD) pin cause the incoming data wake-up flag DATWKF (UART_WKSTS[1]) is generated. The incoming data wake-up is

shown in Figure 6.24-7.

Note1: The UART controller clock source should be selected as HIRC and the compensation time for start bit is about 8.602us. It means that the value of STCOMP (UART_DWKCOMP[15:0]) can be set as 0x68.

Note2: The value of BRD(UART_BAUD[15:0]) should be greater than STCOMP (UART_DWKCOMP[15:0]).



Note 1: Stable count means HCLK source recovery stable count.

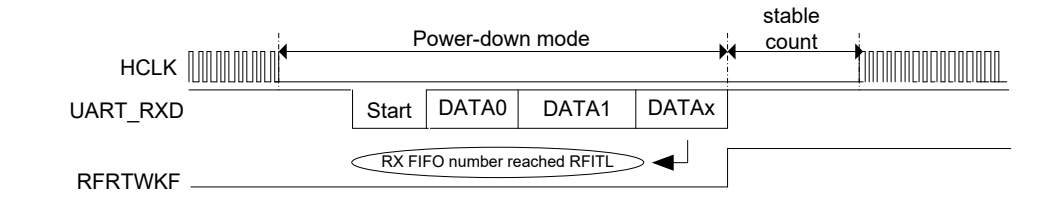
Note 2: UART_CLK stable count means UART clock source recovery stable count.

Figure 6.24-7 UART Data Wake-up

Received Data FIFO Reached Threshold Wake-up

The received data FIFO threshold reached wake-up function is enabled by setting WKRFRTEN (UART_WKCTL[2]). In Power-down mode, when the number of received data in RX FIFO reaches the threshold value RFITL (UART_FIFO[7:4]), it can wake-up the system. If the WKRFRTEN (UART_WKCTL[2]) is enabled, the number of received data in RX FIFO reaches the threshold value RFITL (UART_FIFO[7:4]) cause the received data FIFO reached threshold wake-up flag RFRTWKF (UART_WKSTS[2]) is generated. The Received Data FIFO reached threshold wake-up is shown in Figure 6.24-8.

Note: The UART controller clock source should be selected as LXT in Power-down mode to receive data.



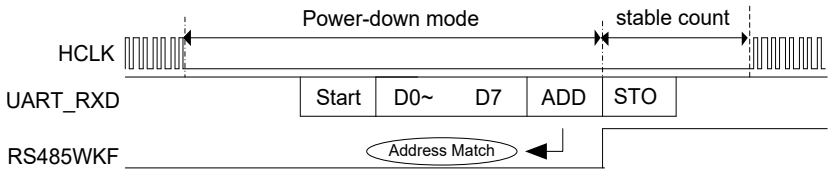
Note: Stable count means HCLK source recovery stable count.

Figure 6.24-8 UART Received Data FIFO Reached Threshold Wake-up

RS-485 Address Match (AAD Mode) Wake-up

The RS-485 address match wake-up function is enabled by setting WKRFRTEN (UART_WKCTL[2]) and WKRS485EN (UART_WKCTL[3]). This function is used for RS-485 Auto Address Detection (AAD) mode in RS-485 function mode and ADDRDEN (UART_ALTCTL[15]) is set to 1. In Power-down mode, when an address byte is detected and matches the ADDRMV (UART_ALTCTL[31:24]) or the number of received data in RX FIFO reaches the threshold value RFITL (UART_FIFO[7:4]), it can wake-up the system. If the WKRS485EN (UART_WKCTL[3]) is enabled, when an address byte is detected and matches the ADDRMV (UART_ALTCTL[31:24]) that cause the RS485 address match (AAD mode) wake-up flag RS485WKF (UART_WKSTS[3]) is generated. The RS-485 Address Match (AAD mode) wake-up is shown in Figure 6.24-9.

Note: The UART controller clock source should be selected as LXT in Power-down mode to receive data.



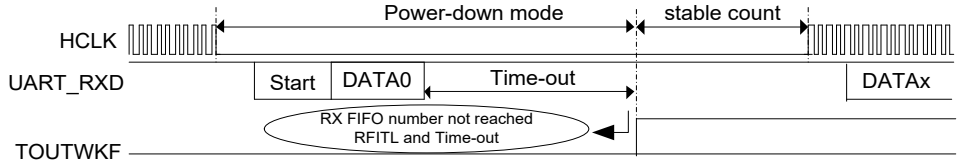
Note: Stable count means HCLK source recovery stable count.

Figure 6.24-9 UART RS-485 AAD Mode Address Match Wake-up

Received Data FIFO Threshold Time-out Wake-up

The received data FIFO threshold time-out wake-up function is enabled by setting WKRFRTEN (UART_WKCTL[2]) and WKTOUTEN (UART_WKCTL[4]). Setting TOCNTEN (UART_INTEN[11]) to enable receiver buffer time-out counter. In Power-down mode, when the number of received data in RX FIFO does not reach the threshold value RFITL (UART_FIFO[7:4]) and the time-out counter equals to the time-out value TOIC (UART_TOUT[7:0]), it can wake-up the system. If the WKTOUTEN (UART_WKCTL[4]) is enabled, when the time-out counter equals to the time-out value TOIC (UART_TOUT[7:0]) that cause the Received Data FIFO threshold time-out wake-up flag TOUTWKF (UART_WKSTS[4]) is generated. The Received Data FIFO threshold time-out wake-up is shown in Figure 6.24-10.

Note: The UART controller clock source should be selected as LXT in Power-down mode to receive data.



Note: Stable count means HCLK source recovery stable count.

Figure 6.24-10 UART Received Data FIFO Threshold Time-out Wake-up

6.24.5.7 UART Controller Interrupt and Status

Each UART controller supports ten types of interrupts including:

- Receive Data Available Interrupt (RDAINT)
- Transmit Holding Register Empty Interrupt (THREINT)
- Transmitter Empty Interrupt (TXENDIF)
- Receive Line Status Interrupt (RLSINT)
  - Break Interrupt Flag (BIF)
  - Framing Error Flag (FEF)
  - Parity Error Flag (PEF)
  - RS-485 Address Byte Detect Flag (ADDRDETF)
- MODEM Status Interrupt (MODEMINT)
  - Detect nCTS State Change Flag (CTSDETF)

- Receiver Buffer Time-out Interrupt (RXTOINT)
- Buffer Error Interrupt (BUFERRINT)
  - TX Overflow Error Interrupt Flag (TXOVIF)
  - RX Overflow Error Interrupt Flag (RXOVIF)
- Wake-up Interrupt (WKINT)
  - nCTS Wake-up Flag (CTSWKF)
  - Incoming Data Wake-up Flag (DATWKF)
  - Received Data FIFO Reached Threshold Wake-up Flag (RFRTWKF)
  - RS-485 Address Match (AAD mode) Wake-up Flag (RS485WKF)
  - Received Data FIFO Threshold Time-out Wake-up Flag (TOUTWKF)
- Auto-Baud Rate Interrupt (ABRINT)
  - Auto-baud Rate Detect Interrupt Flag (ABRDIF)
  - Auto-baud Rate Detect Time-out Interrupt Flag (ABRDTOIF)
- Single-wire Bit Error Detect Interrupt (SWBEINT)

Table 6.24-9 describes the interrupt sources and flags. The interrupt is generated when the interrupt flag is generated and the interrupt enable bit is set. User must clear the interrupt flag after the interrupt is generated.

Interrupt Source	Interrupt Indicator	Interrupt Enable Bit	Interrupt Flag	Flag Caused By	Flag Cleared By
Receive Data Available Interrupt	RDAINT	RDAIEN	RDAIF	N/A	Read UART_DAT
Transmit Holding Register Empty Interrupt	THREINT	THREIEN	THREIF	N/A	Write UART_DAT
Transmitter Empty Interrupt	TXENDINT	TXENDIEN	TXENDIF	N/A	Write UART_DAT
Receive Line Status Interrupt	RLSINT	RLSIEN	RLSIF	RLSIF = BIF	Write ‘1’ to BIF
				RLSIF = FEF	Write ‘1’ to FEF
				RLSIF = PEF	Write ‘1’ to PEF
				RLSIF = ADDRDETF	Write ‘1’ to ADDRDETF
Modem Status Interrupt	MODEMINT	MODEMIEN	MODEMIF	MODEMIF = CTSDETF	Write ‘1’ to CTSDETF
Receiver Buffer Time-out Interrupt	RXTOINT	RXTOIEN	RXTOIF	N/A	Read UART_DAT
Buffer Error Interrupt	BUFERRINT	BUFERRIEN	BUFERRIF	BUFERRIF = TXOVIF	Write ‘1’ to TXOVIF
				BUFERRIF = RXOVIF	Write ‘1’ to RXOVIF
Wake-up Interrupt	WKINT	WKIEN	WKIF	WKIF = CTSWKF	Write ‘1’ to CTSWKF
				WKIF = DATWKF	Write ‘1’ to DATWKF
				WKIF = RFRTWKF	Write ‘1’ to RFRTWKF

				WKIF = RS485WKF	Write '1' to RS485WKF
				WKIF = TOUTWKF	Write '1' to TOUTWKF
Auto-Baud Rate Interrupt	ABRINT	ABRIEN	ABRIF	ABRIF = ABRDIF	Write '1' to ABRDIF
				ABRIF = ABRDTOIF	Write '1' to ABRDTOIF
Single-wire Bit Error Detect Interrupt	SWBEINT	SWBEIEN	SWBEIF	N/A	Writing '1' to SWBEIF

Table 6.24-9 UART controller Interrupt Source and Flag List

6.24.5.8 UART Function Mode

The UART controller provides UART function (Setting FUNCSEL (UART_FUNCSEL [2:0]) to '000' to enable UART function mode). The UART baud rate is up to 9.5 Mbps.

The UART provides full-duplex and asynchronous communications. The transmitter and receiver contain 32 bytes FIFO for payloads. User can program receiver buffer trigger level and receiver buffer time-out detection for receiver. The transmitting data delay time between the last stop and the next start bit can be programed by setting DLY (UART_TOUT [15:8]) register. The UART supports hardware auto-flow control that provides programmable nRTS flow control trigger level. The number of data bytes in RX FIFO is equal to or greater than RTSTRGLV (UART_FIFO[19:16]), the nRTS is de-asserted.

UART Line Control Function

The UART controller supports fully programmable serial-interface characteristics by setting the UART_LINE register. User can program UART_LINE register for the word length, stop bit and parity bit setting. Table 6.24-10 and Table 6.24-11 list the UART word, stop bit length and the parity bit settings.

NSB (UART_LINE[2])	WLS (UART_LINE[1:0])	Word Length (Bit)	Stop Length (Bit)
0	00	5	1
0	01	6	1
0	10	7	1
0	11	8	1
1	00	5	1.5
1	01	6	2
1	10	7	2
1	11	8	2

Table 6.24-10 UART Line Control of Word and Stop Length Setting

Parity Type	SPE (UART_LINE[5])	EPE (UART_LINE[4])	PSS (UART_LINE[7])	PBE (UART_LINE[3])	Description
No Parity	x	x	x	0	No parity bit output.
Parity source from UART_DAT	x	x	1	1	Parity bit is generated and checked by software.

Odd Parity	0	0	0	1	Odd Parity is calculated by adding all the “1’s” in a data stream and adding a parity bit to the total bits, to make the total count an odd number.
Even Parity	0	1	0	1	Even Parity is calculated by adding all the “1’s” in a data stream and adding a parity bit to the total bits, to make the count an even number.
Forced Mask Parity	1	0	0	1	Parity bit always logic 1. Parity bit on the serial byte is set to “1” regardless of total number of “1’s” (even or odd counts).
Forced Space Parity	1	1	0	1	Parity bit always logic 0. Parity bit on the serial byte is set to “0” regardless of total number of “1’s” (even or odd counts).

Table 6.24-11 UART Line Control of Parity Bit Setting

UART Auto-Flow Control Function

The UART supports auto-flow control function that uses two signals, nCTS (clear-to-send) and nRTS (request-to-send), to control the flow of data transfer between the UART and external devices (e.g. Modem). When auto-flow is enabled, the UART is not allowed to receive data until the UART asserts nRTS to external device. When the number of bytes stored in the RX FIFO equals the value of RTSTRGLV (UART_FIFO [19:16]), the nRTS is de-asserted. The UART sends data out when UART detects nCTS is asserted from external device. If the valid asserted nCTS is not detected, the UART will not send data out. The auto flow control block diagram is shown in Figure 6.24-11.

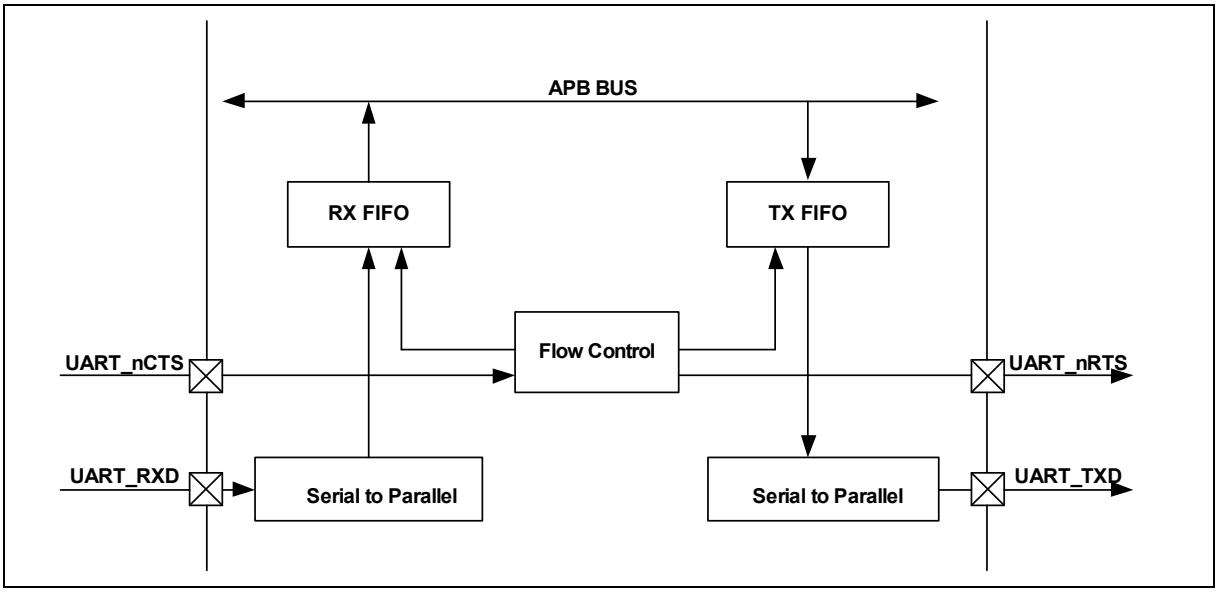


Figure 6.24-11 Auto-Flow Control Block Diagram

Figure 6.24-12 demonstrates the nCTS auto-flow control of UART function mode. User must set ATOCTSEN (UART_INTEN [13]) to enable nCTS auto-flow control function. The CTSACTLV (UART_MODEMSTS [8]) can set nCTS pin input active state. The CTSDETF (UART_MODEMSTS[0]) is set when any state change of nCTS pin input has occurred, and then TX data will be automatically transmitted from TX FIFO.

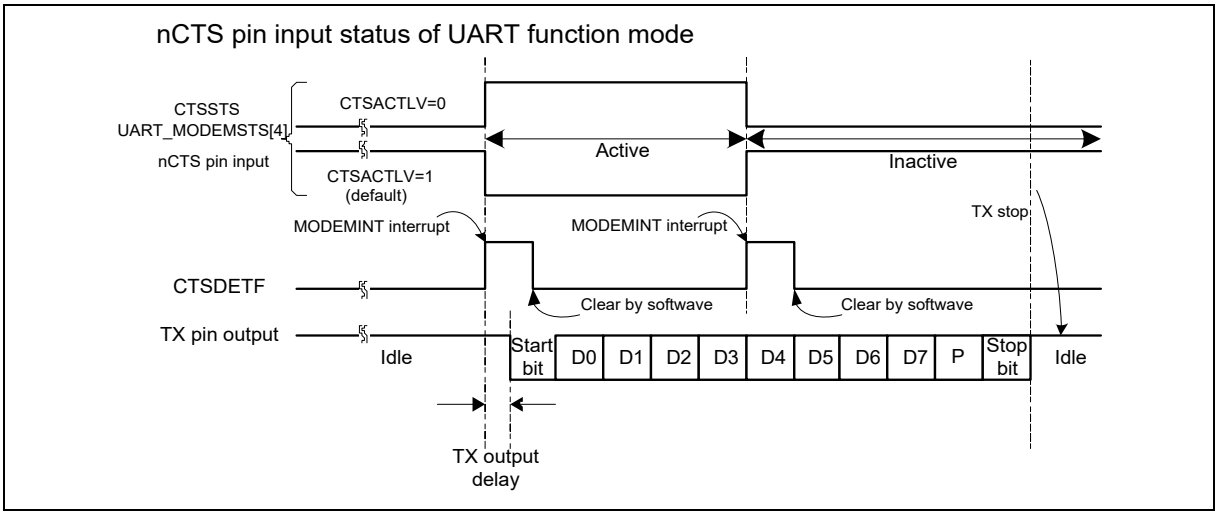


Figure 6.24-12 UART nCTS Auto-Flow Control Enabled

As shown in Figure 6.24-13, in UART nRTS auto-flow control mode (ATORTSEN(UART_INTEN[12])=1), the nRTS internal signal is controlled by UART FIFO controller with RTSTRGLV(UART_FIFO[19:16]) trigger level.

Setting RTSACTLV(UART_MODEM[9]) can control the nRTS pin output is inverse or non-inverse from nRTS signal. User can read the RTSSTS (UART_MODEM[13]) bit to get real nRTS pin output voltage logic status.

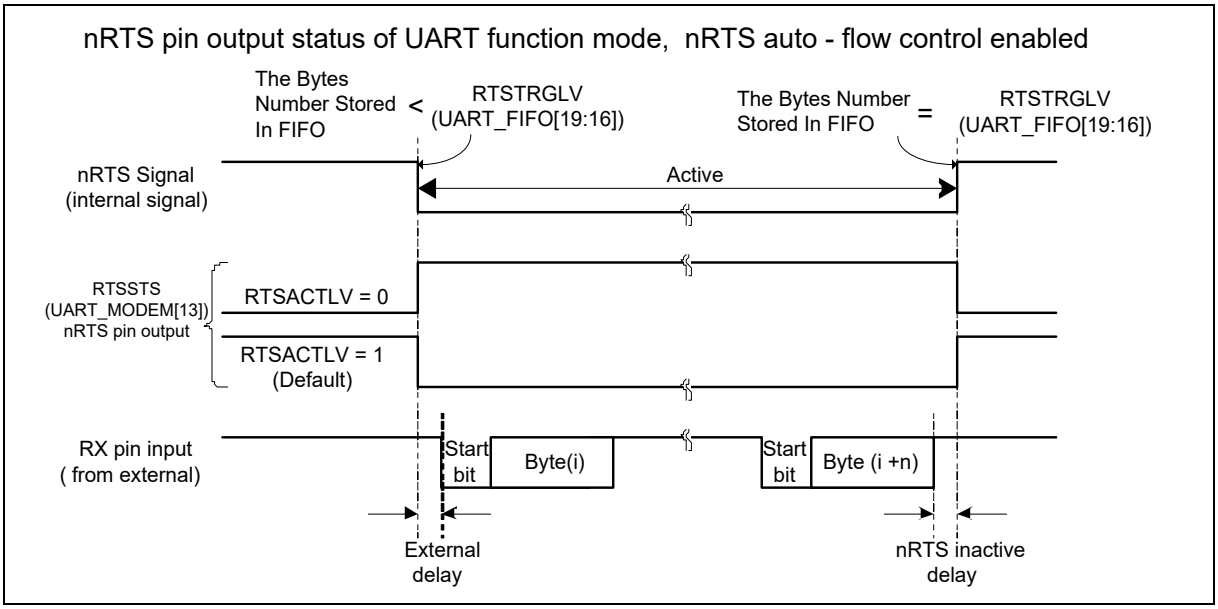


Figure 6.24-13 UART nRTS Auto-Flow Control Enabled

As shown in Figure 6.24-14, in software mode (ATORTSEN(UART_INTEN[12])=0), the nRTS flow is directly controlled by software programming of RTS(UART_MODEM[1]) control bit.

Setting RTSACTLV(UART_MODEM[9]) can control the nRTS pin output is inverse or non-inverse from RTS(UART_MODEM[1]) control bit. User can read the RTSSTS(UART_MODEM[13]) bit to get real nRTS pin output voltage logic status.

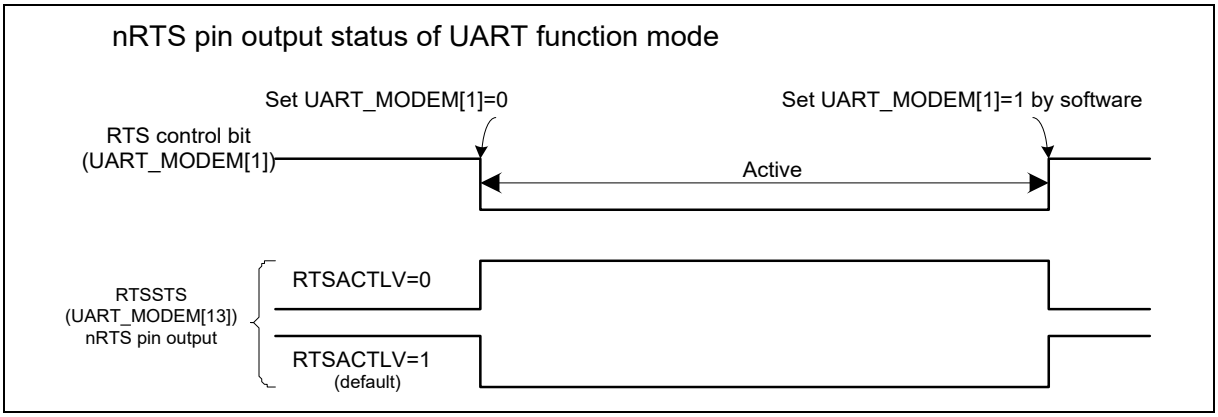


Figure 6.24-14 UART nRTS Auto-Flow with Software Control

6.24.5.9 IrDA Function Mode

The UART controller also provides Serial IrDA (SIR, Serial Infrared) function (Setting UART_FUNCSEL [2:0] to '010' to enable the IrDA function). The SIR specification defines a short-range infrared asynchronous serial transmission mode with one start bit, 8 data bits, and 1 stop bit. The maximum data rate is 115.2 kbps. The IrDA SIR block contains an IrDA SIR protocol encoder/decoder. The IrDA SIR protocol is half-duplex only. So, it cannot transmit and receive data at the same time. The IrDA SIR physical layer specifies a minimum 10 ms transfer delay between transmission and reception, and this delay feature must be implemented by software.

In IrDA mode, the BAUDM1 (UART_BAUD [29]) must be cleared.

Baud Rate = Clock / (16 * (BRD +2)), where BRD (UART_BAUD[15:0]) is Baud Rate Divider in UART_BAUD register.

Note: The tolerance of baud-rate is ±5% between IrDA master and IrDA slave.

The IrDA control block diagram is shown in Figure 6.24-15.

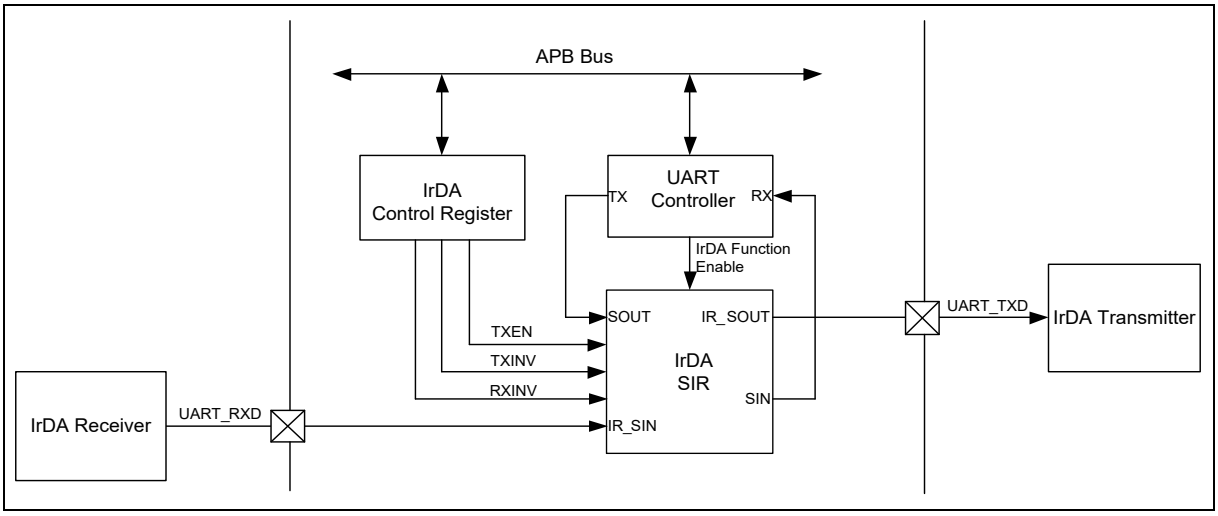


Figure 6.24-15 IrDA Control Block Diagram

IrDA SIR Transmit Encoder

The IrDA SIR Transmit Encoder modulates Non-Return-to-Zero (NRZ) transmit bit stream output from UART. The IrDA SIR physical layer specifies the use of Return-to-Zero, Inverted (RZI) modulation scheme which represents logic 0 as an infra light pulse. The modulated output pulse stream is transmitted to an external output driver and infrared light emitting diode.

The transmitted pulse width is specified as 3/16 period of baud rate.

IrDA SIR Receive Decoder

The IrDA SIR Receive Decoder demodulates the Return-to-Zero bit stream from the input detector and outputs the NRZ serial bits stream to the UART received data input.

In idle state, the decoder input is high. A start bit is detected when the decoder input is LOW. In normal operation, the RXINV (UART_IRDA[6]) is set to '1' and TXINV (UART_IRDA[5]) is set to '0'.

IrDA SIR Operation

The IrDA SIR encoder/decoder provides functionality which converts between UART data stream and half-duplex serial SIR interface. Figure 6.24-16 is IrDA encoder/decoder waveform.

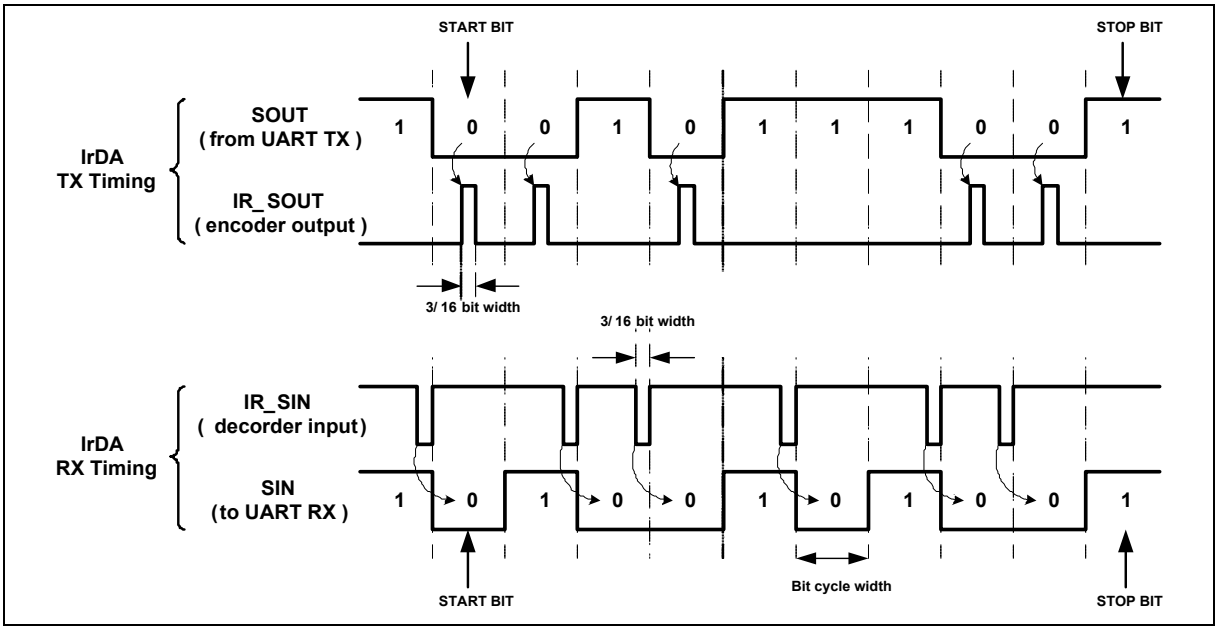


Figure 6.24-16 IrDA TX/RX Timing Diagram

6.24.5.10 RS-485 Function Mode

Another alternate function of UART controller is RS-485 function (user must set UART_FUNCSEL [2:0] to '011' to enable RS-485 function), and direction control provided by nRTS pin from an asynchronous serial port. The RS-485 transceiver control is implemented by using the nRTS control signal to enable the RS-485 driver. Many characteristics of the RX and TX are same as UART in RS-485 mode.

The UART controller can be configured as an RS-485 addressable slave and the RS-485 master transmitter will identify an address character by setting the parity (9-th bit) to 1. For data characters, the parity is set to 0. Software can use UART_LINE register to control the 9-th bit (When the PBE, EPE and SPE are set, the 9-th bit is transmitted 0 and when PBE and SPE are set and EPE is cleared, the 9-th bit is transmitted 1).

The controller supports three operation modes: RS-485 Normal Multidrop Operation Mode (NMM), RS-485 Auto Address Detection Operation Mode (AAD) and RS-485 Auto Direction Control Operation Mode (AUD). Software can choose any operation mode by programming the UART_ALTCTL register, and drive the transfer delay time between the last stop bit leaving the TX FIFO and the de-assertion of by setting DLY (UART_TOUT [15:8]) register.

RS-485 Normal Multidrop Operation Mode (NMM)

In RS-485 Normal Multidrop Operation Mode (RS485NMM (UART_ALTCTL[8]) = 1), in first, software must decide the data which before the address byte be detected will be stored in RX FIFO or not. If

software wants to ignore any data before address byte detected, the flow is set RXOFF (UART_FIFO [8]) then enable RS485NMM (UART_ALTCTL [8]) and the receiver will ignore any data until an address byte is detected (bit 9 = 1) and the address byte data will be stored in the RX FIFO. If software wants to receive any data before address byte detected, the flow is disables RXOFF (UART_FIFO [8]) then enable RS485NMM (UART_ALTCTL [8]) and the receiver will received any data.

If an address byte is detected (bit 9 = 1), it will generate an interrupt to CPU and RXOFF (UART_FIFO [8]) can decide whether accepting the following data bytes are stored in the RX FIFO. If software disables receiver by setting RXOFF (UART_FIFO [8]) register, when a next address byte is detected, the controller will clear the RXOFF (UART_FIFO [8]) bit and the address byte data will be stored in the RX FIFO.

RS-485 Auto Address Detection Operation Mode (AAD)

In RS-485 Auto Address Detection Operation Mode (RS485AAD (UART_ALTCTL[9]) = 1), the receiver will ignore any data until an address byte is detected (bit 9 = 1) and the address byte data matches the ADDRMV (UART_ALTCTL[31:24]) value. The address byte data will be stored in the RX FIFO. The all received byte data will be accepted and stored in the RX FIFO until an address byte data not match the ADDRMV (UART_ALTCTL[31:24]) value.

RS-485 Auto Direction Function (AUD)

Another option function of RS-485 controllers is RS-485 auto direction control function (RS485AUD (UART_ALTCTL[10) = 1). The RS-485 transceiver control is implemented by using the nRTS control signal from an asynchronous serial port. The nRTS line is connected to the RS-485 transceiver enable pin such that setting the nRTS line to high (logic 1) enables the RS-485 transceiver. Setting the nRTS line to low (logic 0) puts the transceiver into the tri-state condition. User can set RTSACTLV in UART_MODEM register to change the nRTS driving level.

Figure 6.24-17 demonstrates the RS-485 nRTS driving level in AUD mode. The nRTS pin will be automatically driven during TX data transmission.

Setting RTSACTLV(UART_MODEM[9]) can control nRTS pin output driving level. User can read the RTSSTS(UART_MODEM[13]) bit to get real nRTS pin output voltage logic status.

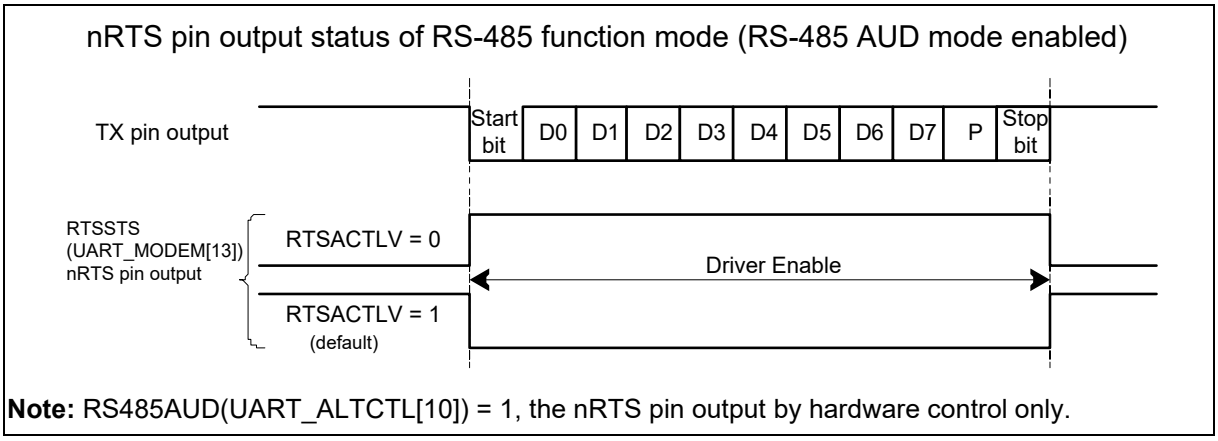
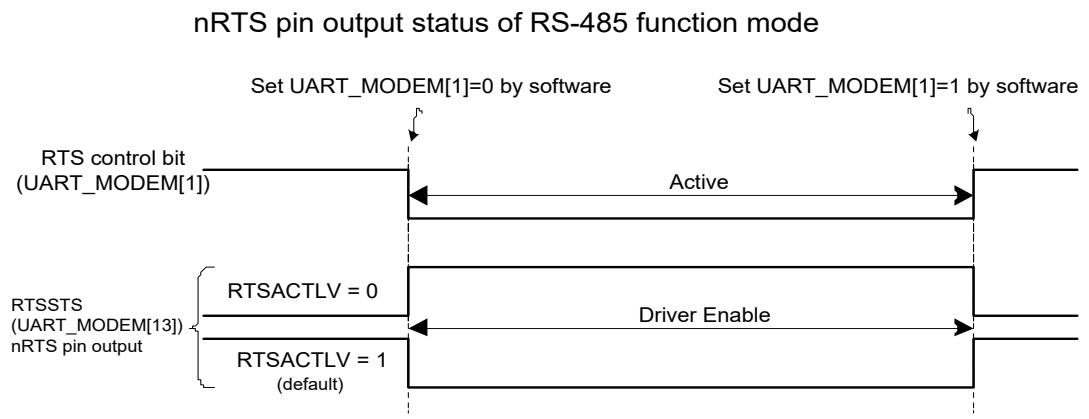


Figure 6.24-17 RS-485 nRTS Driving Level in Auto Direction Mode

Figure 6.24-18 demonstrates the RS-485 nRTS driving level in software control (RS485AUD (UART_ALTCTL[10])=0). The nRTS driving level is controlled by programing the RTS(UART_MODEM[1]) control bit.

Setting RTSACTLV (UART_MODEM[9]) can control the nRTS pin output is inverse or non-inverse from RTS(UART_MODEM[1]) control bit. User can read the RTSSTS (UART_MODEM[13]) bit to get real nRTS pin output voltage logic status. The structure of RS-485 frame is shown in Figure 6.24-19.

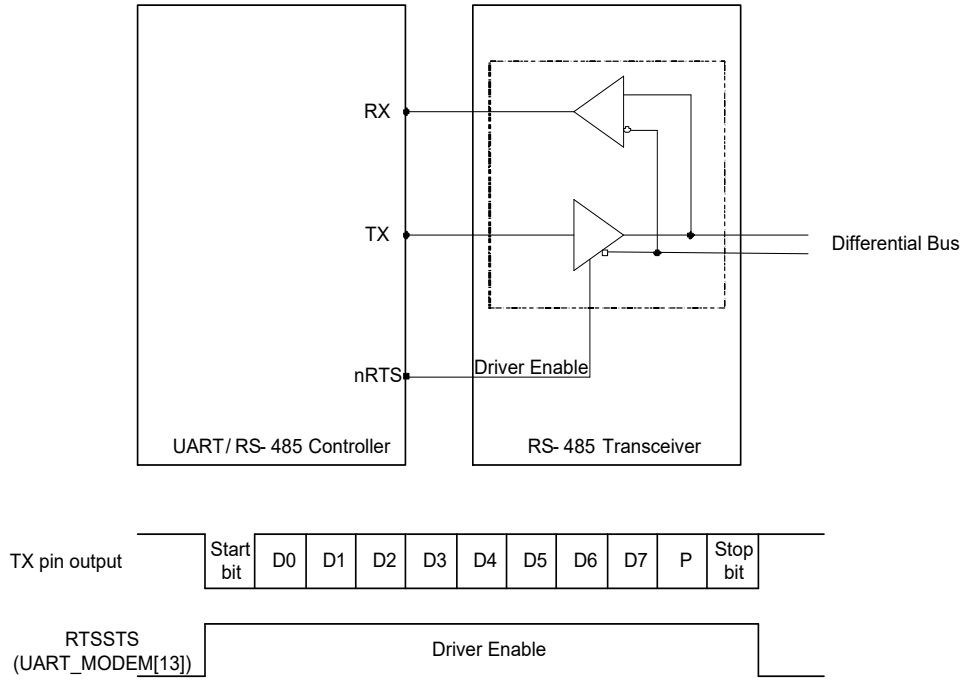


Note: RS485AUD (UART_ALTCTL[10]) = 0, support nRTS control bit by software control only.

Figure 6.24-18 RS-485 nRTS Driving Level with Software Control

Programming Sequence Example:

1. Program FUNCSEL in UART_FUNCSEL to select RS-485 function.
2. Program the RXOFF (UART_FIFO[8]) to determine enable or disable the receiver RS-485 receiver.
3. Program the RS485NMM (UART_ALTCTL[8]) or RS485AAD (UART_ALTCTL[9]) mode.
4. If the RS485AAD (UART_ALTCTL[9]) mode is selected, the ADDRMV (UART_ALTCTL[31:24]) is programmed for auto address match value.
5. Determine auto direction control by programming RS485AUD (UART_ALTCTL[10]).



Note: RS485AUD(UART_ALTCTL[10]) must be set to 1, and RTSACTLV(UART_MODEM[9]) must be set to 0.

Figure 6.24-19 Structure of RS-485 Frame

6.24.5.11 UART Single-wire Half Duplex

The UART controller provides single-wire half duplex function in UART function mode. Setting UART_FUNCSEL [2:0] to '100' to enable the UART Single-wire function. The single-wire bus is idle (RXIDLE(UART_FIFOSTS[29]) = 1) in RX state. Before writing data to transmit buffer (UART_DAT[7:0]), the bus state should be checked in idle (RXIDLE(UART_FIFOSTS[29]). By writing data to transmit buffer, the bus state transfers to TX state immediately. After the transmission, the bus state transfers from TX state to RX state.

The UART will not allowed to receive data in single-wire half duplex function TX mode. If nRTS is asserted in TX mode, nRTS will make the docking UART device send out data and cause bus confliction. To reduce the bus confliction, the UART controller supports flow control function and bit error detection but does not support auto-flow control function. The nRTS is automatically inactivated in single-wire TX state. In TX state, the UART controller will monitor bus state. If the bus state is not equal to TX state, the SWBEIF(UART_INTSTS[16]) is set.

6.24.5.12 PDMA Transfer Function

The UART controller supports PDMA transfer function.

By configuring PDMA parameter and set UART_DAT as the PDMA destination address. When TXPDMAEN (UART_INTEN[14]) is set to 1, the controller will issue request to PDMA controller to start the PDMA transmission process automatically.

By configuring PDMA parameter and set UART_DAT as the PDMA source address. When RXPDMAEN (UART_INTEN[15]) is set to 1, the controller will start the PDMA reception process. UART controller will issue request to PDMA controller automatically when there is data in the RX FIFO buffer.

6.24.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
UART Base Address: **UARTx_BA = 0x4070_0000 + (0x10000 * x)** **x=0,1..15** **UART16_BA = 0x4088_0000**				
UART_DAT **x=0,1..16**	UARTx_BA+0x00	R/W	UART Receive/Transmit Buffer Register	Undefined
UART_INTEN **x=0,1..16**	UARTx_BA+0x04	R/W	UART Interrupt Enable Register	0x0000_0000
UART_FIFO **x=0,1..16**	UARTx_BA+0x08	R/W	UART FIFO Control Register	0x0000_0101
UART_LINE **x=0,1..16**	UARTx_BA+0x0C	R/W	UART Line Control Register	0x0000_0000
UART_MODEM **x=0,1..16**	UARTx_BA+0x10	R/W	UART Modem Control Register	0x0000_0200
UART_MODEMSTS **x=0,1..16**	UARTx_BA+0x14	R/W	UART Modem Status Register	0x0000_0110
UART_FIFOSTS **x=0,1..16**	UARTx_BA+0x18	R/W	UART FIFO Status Register	0xB040_4000
UART_INTSTS **x=0,1..16**	UARTx_BA+0x1C	R/W	UART Interrupt Status Register	0x0040_0002
UART_TOUT **x=0,1..16**	UARTx_BA+0x20	R/W	UART Time-out Register	0x0000_0000
UART_BAUD **x=0,1..16**	UARTx_BA+0x24	R/W	UART Baud Rate Divider Register	0x0F00_0000
UART_IRDA **x=0,1..16**	UARTx_BA+0x28	R/W	UART IrDA Control Register	0x0000_0040
UART_ALTCTL **x=0,1..16**	UARTx_BA+0x2C	R/W	UART Alternate Control/Status Register	0x0000_000C
UART_FUNCSEL **x=0,1..16**	UARTx_BA+0x30	R/W	UART Function Select Register	0x0000_0000
UART_BRCOMP **x=0,1..16**	UARTx_BA+0x3C	R/W	UART Baud Rate Compensation Register	0x0000_0000
UART_WKCTL **x=0,1..16**	UARTx_BA+0x40	R/W	UART Wake-up Control Register	0x0000_0000

UART_WKSTS x=0,1..16	UARTx_BA+0x44	R/W	UART Wake-up Status Register	0x0000_0000
UART_DWKCOMP x=0,1..16	UARTx_BA+0x48	R/W	UART Incoming Data Wake-up Compensation Register	0x0000_0000

6.24.7 Register Description

UART Receive/Transmit Buffer Register (UART_DAT)

Register	Offset	R/W	Description	Reset Value
UART_DAT x=0,1..16	UARTx_BA+0x00	R/W	UART Receive/Transmit Buffer Register	Undefined

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**PARITY**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DAT							

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8]	**PARITY**	**Parity Bit Receive/Transmit Buffer** Write Operation: By writing to this bit, the parity bit will be stored in transmitter FIFO. If PBE (UART_LINE[3]) and PSS (UART_LINE[7]) are set, the UART controller will send out this bit follow the DAT (UART_DAT[7:0]) through the UART_TXD. Read Operation: If PBE (UART_LINE[3]) and PSS (UART_LINE[7]) are enabled, the parity bit can be read by this bit. **Note:** This bit has effect only when PBE (UART_LINE[3]) and PSS (UART_LINE[7]) are set.
[7:0]	**DAT**	**Data Receive/Transmit Buffer** Write Operation: By writing one byte to this register, the data byte will be stored in transmitter FIFO. The UART controller will send out the data stored in transmitter FIFO top location through the UART_TXD. Read Operation: By reading this register, the UART controller will return an 8-bit data received from receiver FIFO.

UART Interrupt Enable Register (UART_INTEN)

Register	Offset	R/W	Description	Reset Value
UART_INTEN x=0,1..16	UARTx_BA+0x04	R/W	UART Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved	TXENDIEN	Reserved			ABRIEN	Reserved	SWBEIEN
15	14	13	12	11	10	9	8
RXPDMAEN	TXPDMAEN	ATOCTSEN	ATORTSEN	TOCNTEN	Reserved		
7	6	5	4	3	2	1	0
Reserved	WKIEN	BUFEIEN	RXTOIEN	MODEMIEN	RLSIEN	THREIEN	RDAIEN

Bits	Description	
[31:23]	**Reserved**	Reserved.
[22]	**TXENDIEN**	**Transmitter Empty Interrupt Enable Bit** If TXENDIEN (UART_INTEN[22]) is enabled, the Transmitter Empty interrupt TXENDINT (UART_INTSTS[30]) will be generated when TXENDIF (UART_INTSTS[22]) is set (TX FIFO (UART_DAT) is empty and the STOP bit of the last byte has been transmitted). 0 = Transmitter empty interrupt Disabled. 1 = Transmitter empty interrupt Enabled.
[21:19]	**Reserved**	Reserved.
[18]	**ABRIEN**	**Auto-baud Rate Interrupt Enable Bit** 0 = Auto-baud rate interrupt Disabled. 1 = Auto-baud rate interrupt Enabled.
[17]	**Reserved**	Reserved.
[16]	**SWBEIEN**	**Single-wire Bit Error Detection Interrupt Enable Bit** Set this bit, the Single-wire Half Duplex Bit Error Detection Interrupt SWBEINT(UART_INTSTS[24]) is generated when Single-wire Bit Error Detection SWBEIF(UART_INTSTS[16]) is set. 0 = Single-wire Bit Error Detect Interrupt Disabled. 1 = Single-wire Bit Error Detect Interrupt Enabled. **Note:** This bit is valid when FUNCSEL (UART_FUNCSEL[2:0]) is select UART Single-wire mode.
[15]	**RXPDMAEN**	**RX PDMA Enable Bit** This bit can enable or disable RX PDMA service. 0 = RX PDMA Disabled. 1 = RX PDMA Enabled. **Note:** If RLSIEN (UART_INTEN[2]) is enabled and HWRLSINT (UART_INTSTS[26]) is set to 1, the RLS (Receive Line Status) Interrupt is caused. If RLS interrupt is caused by Break Error Flag BIF(UART_FIFOSTS[6]), Frame Error Flag FEF(UART_FIFO[5]) or Parity Error Flag PEF(UART_FIFOSTS[4]), UART PDMA receive request operation is stopped. Clear Break Error Flag BIF or Frame Error Flag FEF or Parity Error Flag PEF by writing “1” to corresponding BIF, FEF and PEF to make UART PDMA receive request operation continue.

[14]	**TXPDMAEN**	**TX PDMA Enable Bit** 0 = TX PDMA Disabled. 1 = TX PDMA Enabled. **Note:** If RLSIEN (UART_INTEN[2]) is enabled and HWRLSINT (UART_INTSTS[26]) is set to 1, the RLS (Receive Line Status) Interrupt is caused. If RLS interrupt is caused by Break Error Flag BIF(UART_FIFOSTS[6]), Frame Error Flag FEF(UART_FIFO[5]) or Parity Error Flag PEF(UART_FIFOSTS[4]), UART PDMA transmit request operation is stopped. Clear Break Error Flag BIF or Frame Error Flag FEF or Parity Error Flag PEF by writing "1" to corresponding BIF, FEF and PEF to make UART PDMA transmit request operation continue.
[13]	**ATOCTSEN**	**nCTS Auto-flow Control Enable Bit** 0 = nCTS auto-flow control Disabled. 1 = nCTS auto-flow control Enabled. **Note:** When nCTS auto-flow is enabled, the UART will send data to external device if nCTS input assert (UART will not send data to device until nCTS is asserted).
[12]	**ATORTSEN**	**nRTS Auto-flow Control Enable Bit** 0 = nRTS auto-flow control Disabled. 1 = nRTS auto-flow control Enabled. **Note:** When nRTS auto-flow is enabled, if the number of bytes in the RX FIFO equals the RTSTRGLV (UART_FIFO[19:16]), the UART will de-assert nRTS signal.
[11]	**TOCNTEN**	**Receive Buffer Time-out Counter Enable Bit** 0 = Receive Buffer Time-out counter Disabled. 1 = Receive Buffer Time-out counter Enabled.
[10:7]	**Reserved**	Reserved.
[6]	**WKIEN**	**Wake-up Interrupt Enable Bit** 0 = Wake-up Interrupt Disabled. 1 = Wake-up Interrupt Enabled.
[5]	**BUFEIEN**	**Buffer Error Interrupt Enable Bit** 0 = Buffer error interrupt Disabled. 1 = Buffer error interrupt Enabled.
[4]	**RXTOIEN**	**RX Time-out Interrupt Enable Bit** 0 = RX time-out interrupt Disabled. 1 = RX time-out interrupt Enabled.
[3]	**MODEMIEN**	**Modem Status Interrupt Enable Bit** 0 = Modem status interrupt Disabled. 1 = Modem status interrupt Enabled.
[2]	**RLSIEN**	**Receive Line Status Interrupt Enable Bit** 0 = Receive Line Status interrupt Disabled. 1 = Receive Line Status interrupt Enabled.
[1]	**THREIEN**	**Transmit Holding Register Empty Interrupt Enable Bit** 0 = Transmit holding register empty interrupt Disabled. 1 = Transmit holding register empty interrupt Enabled.
[0]	**RDAIEN**	**Receive Data Available Interrupt Enable Bit** 0 = Receive data available interrupt Disabled. 1 = Receive data available interrupt Enabled.

UART FIFO Control Register (UART_FIFO)

Register	Offset	R/W	Description	Reset Value
UART_FIFO x=0,1..16	UARTx_BA+0x08	R/W	UART FIFO Control Register	0x0000_0101

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				RTSTRGLV			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							RXOFF
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RFITL				Reserved	TXRST	RXRST	Reserved

Bits	Description	
[31:20]	**Reserved**	Reserved.
[19:16]	**RTSTRGLV**	**nRTS Trigger Level for Auto-flow Control** 0000 = nRTS Trigger Level is 1 byte. 0001 = nRTS Trigger Level is 4 bytes. 0010 = nRTS Trigger Level is 8 bytes. 0011 = nRTS Trigger Level is 14 bytes. 0100 = nRTS Trigger Level is 30 bytes. Others = Reserved. **Note:** This field is used for auto nRTS flow control.
[15:9]	**Reserved**	Reserved.
[8]	**RXOFF**	**Receiver Disable Bit** The receiver is disabled or not (set 1 to disable receiver). 0 = Receiver Enabled. 1 = Receiver Disabled. **Note:** This bit is used for RS-485 Normal Multi-drop mode. It should be programmed before RS485NMM (UART_ALTCTL [8]) is programmed.
[7:4]	**RFITL**	**RX FIFO Interrupt Trigger Level** When the number of bytes in the receive FIFO equals the RFITL, the RDAIF (UART_INTSTS[0]) will be set (if RDAIEN (UART_INTEN [0]) enabled, and an interrupt will be generated). 0000 = RX FIFO Interrupt Trigger Level is 1 byte. 0001 = RX FIFO Interrupt Trigger Level is 4 bytes. 0010 = RX FIFO Interrupt Trigger Level is 8 bytes. 0011 = RX FIFO Interrupt Trigger Level is 14 bytes. 0100 = RX FIFO Interrupt Trigger Level is 30 bytes. Others = Reserved.
[3]	**Reserved**	Reserved.

[2]	**TXRST**	**TX Field Software Reset** When TXRST (UART_FIFO[2]) is set, all the byte in the transmit FIFO and TX internal state machine are cleared. 0 = No effect. 1 = Reset the TX internal state machine and pointers. **Note 1:** This bit will automatically clear at least 3 UART peripheral clock cycles. **Note 2:** Before setting this bit, it should wait for the TXEMPTYF (UART_FIFOSTS[28]) be set.
[1]	**RXRST**	**RX Field Software Reset** When RXRST (UART_FIFO[1]) is set, all the byte in the receiver FIFO and RX internal state machine are cleared. 0 = No effect. 1 = Reset the RX internal state machine and pointers. **Note 1:** This bit will automatically clear at least 3 UART peripheral clock cycles. **Note 2:** Before setting this bit, it should wait for the RXIDLE (UART_FIFOSTS[29]) be set.
[0]	**Reserved**	Reserved.

UART Line Control Register (UART_LINE)

Register	Offset	R/W	Description	Reset Value
UART_LINE **x=0,1..16**	UARTx_BA+0x0C	R/W	UART Line Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**RXDINV**	**TXDINV**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PSS	**BCB**	**SPE**	**EPE**	**PBE**	**NSB**	**WLS**	

Bits	Description	
[31:10]	**Reserved**	Reserved.
[9]	**RXDINV**	**RX Data Inverted** 0 = Received data signal inverted Disabled. 1 = Received data signal inverted Enabled. **Note 1:** Before setting this bit, TXRXDIS (UART_FUNCSEL[3]) should be set then waited for TXRXACT (UART_FIFOSTS[31]) is cleared. When the configuration is done, cleared TXRXDIS (UART_FUNCSEL[3]) to activate UART controller. **Note 2:** This bit is valid when FUNCSEL (UART_FUNCSEL[2:0]) is select UART, LIN or RS485 function.
[8]	**TXDINV**	**TX Data Inverted** 0 = Transmitted data signal inverted Disabled. 1 = Transmitted data signal inverted Enabled. **Note 1:** Before setting this bit, TXRXDIS (UART_FUNCSEL[3]) should be set then waited for TXRXACT (UART_FIFOSTS[31]) is cleared. When the configuration is done, cleared TXRXDIS (UART_FUNCSEL[3]) to activate UART controller. **Note 2:** This bit is valid when FUNCSEL (UART_FUNCSEL[2:0]) is select UART or RS485 function.
[7]	**PSS**	**Parity Bit Source Selection** The parity bit can be selected to be generated and checked automatically or by software. 0 = Parity bit is generated by EPE (UART_LINE[4]) and SPE (UART_LINE[5]) setting and checked automatically. 1 = Parity bit generated and checked by software. **Note 1:** This bit has effect only when PBE (UART_LINE[3]) is set. **Note 2:** If PSS is 0, the parity bit is transmitted and checked automatically. If PSS is 1, the transmitted parity bit value can be determined by writing PARITY (UART_DAT[8]) and the parity bit can be read by reading PARITY (UART_DAT[8]).
[6]	**BCB**	**Break Control Bit** 0 = Break Control Disabled. 1 = Break Control Enabled. **Note:** When this bit is set to logic 1, the transmitted serial data output (TX) is forced to the Spacing State (logic 0). This bit acts only on TX line and has no effect on the transmitter logic.

[5]	**SPE**	**Stick Parity Enable Bit** 0 = Stick parity Disabled. 1 = Stick parity Enabled. **Note:** If PBE (UART_LINE[3]) and EPE (UART_LINE[4]) are logic 1, the parity bit is transmitted and checked as logic 0. If PBE (UART_LINE[3]) is 1 and EPE (UART_LINE[4]) is 0 then the parity bit is transmitted and checked as 1.
[4]	**EPE**	**Even Parity Enable Bit** 0 = Odd number of logic 1's is transmitted and checked in each word. 1 = Even number of logic 1's is transmitted and checked in each word. **Note:** This bit has effect only when PBE (UART_LINE[3]) is set.
[3]	**PBE**	**Parity Bit Enable Bit** 0 = Parity bit generated Disabled. 1 = Parity bit generated Enabled. **Note:** Parity bit is generated on each outgoing character and is checked on each incoming data.
[2]	**NSB**	**Number of "STOP Bit"** 0 = One "STOP bit" is generated in the transmitted data. 1 = When select 5-bit word length, 1.5 "STOP bit" is generated in the transmitted data. When select 6-, 7- and 8-bit word length, 2 "STOP bit" is generated in the transmitted data.
[1:0]	**WLS**	**Word Length Selection** This field sets UART word length. 00 = 5 bits. 01 = 6 bits. 10 = 7 bits. 11 = 8 bits.

UART Modem Control Register (UART_MODEM)

Register	Offset	R/W	Description	Reset Value
UART_MODEM x=0,1..16	UARTx_BA+0x10	R/W	UART Modem Control Register	0x0000_0200

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		RTSSTS	Reserved			RTSACTLV	Reserved
7	6	5	4	3	2	1	0
Reserved						RTS	Reserved

Bits	Description	
[31:14]	**Reserved**	Reserved.
[13]	**RTSSTS**	**nRTS Pin Status (Read Only)** This bit mirror from nRTS pin output of voltage logic status. 0 = nRTS pin output is low level voltage logic state. 1 = nRTS pin output is high level voltage logic state.
[12:10]	**Reserved**	Reserved.
[9]	**RTSACTLV**	**nRTS Pin Active Level** This bit defines the active level state of nRTS pin output. 0 = nRTS pin output is high level active. 1 = nRTS pin output is low level active. (Default) **Note 1:** Refer to Figure 6.24-13 and Figure 6.24-14 for UART function mode. **Note 2:** Refer to Figure 6.24-17 and Figure 6.24-18 for RS-485 function mode. **Note 3:** Before setting this bit, TXRXDIS (UART_FUNCSEL[3]) should be set then waited for TXRXACT (UART_FIFOSTS[31]) is cleared. When the configuration is done, cleared TXRXDIS (UART_FUNCSEL[3]) to activate UART controller.
[8:2]	**Reserved**	Reserved.
[1]	**RTS**	**nRTS Signal Control** This bit is direct control internal nRTS (Request-to-send) signal active or not, and then drive the nRTS pin output with RTSACTLV bit configuration. 0 = nRTS signal is active. 1 = nRTS signal is inactive. **Note 1:** The nRTS signal control bit is not effective when nRTS auto-flow control is enabled in UART function mode. **Note 2:** The nRTS signal control bit is not effective when RS-485 auto direction mode (AUD) is enabled in RS-485 function mode. **Note 3:** Single-wire mode is support this feature.
[0]	**Reserved**	Reserved.

UART Modem Status Register (UART_MODEMSTS)

Register	Offset	R/W	Description	Reset Value
UART_MODEMSTS **x=0,1..16**	UARTx_BA+0x14	R/W	UART Modem Status Register	0x0000_0110

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							CTSACTLV
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			CTSSTS	Reserved			CTSDETF

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8]	**CTSACTLV**	**nCTS Pin Active Level** This bit defines the active level state of nCTS pin input. 0 = nCTS pin input is high level active. 1 = nCTS pin input is low level active. (Default) **Note:** Before setting this bit, TXRXDIS (UART_FUNCSEL[3]) should be set then waited for TXRXACT (UART_FIFOSTS[31]) is cleared. When the configuration is done, cleared TXRXDIS (UART_FUNCSEL[3]) to activate UART controller.
[7:5]	**Reserved**	Reserved.
[4]	**CTSSTS**	**nCTS Pin Status (Read Only)** This bit mirror from nCTS pin input of voltage logic status. 0 = nCTS pin input is low level voltage logic state. 1 = nCTS pin input is high level voltage logic state. **Note:** This bit echoes when UART controller peripheral clock is enabled, and nCTS multi-function port is selected.
[3:1]	**Reserved**	Reserved.
[0]	**CTSDETF**	**Detect nCTS State Change Flag** This bit is set whenever nCTS input has change state, and it will generate Modem interrupt to CPU when MODEMIEN (UART_INTEN [3]) is set to 1. 0 = nCTS input has not change state. 1 = nCTS input has change state. **Note:** This bit can be cleared by writing “1” to it.

UART FIFO Status Register (UART_FIFOSTS)

Register	Offset	R/W	Description	Reset Value
UART_FIFOSTS x=0,1..16	UARTx_BA+0x18	R/W	UART FIFO Status Register	0xB040_4000

31	30	29	28	27	26	25	24
TXRXACT	Reserved	RXIDLE	TXEMPTYF	Reserved			TXOVIF
23	22	21	20	19	18	17	16
TXFULL	TXEMPTY	TXPTR					
15	14	13	12	11	10	9	8
RXFULL	RXEMPTY	RXPTR					
7	6	5	4	3	2	1	0
Reserved	BIF	FEF	PEF	ADDRDETF	ABRDTOIF	ABRDIF	RXOVIF

Bits	Description	
[31]	**TXRXACT**	**TX and RX Active Status (Read Only)** This bit indicates TX and RX are active or inactive. 0 = TX and RX are inactive. 1 = TX and RX are active. (Default) **Note:** When TXRXDIS (UART_FUNCSEL[3]) is set and both TX and RX are in idle state, this bit is cleared. The UART controller cannot transmit or receive data at this moment. Otherwise this bit is set.
[30]	**Reserved**	Reserved.
[29]	**RXIDLE**	**RX Idle Status (Read Only)** This bit is set by hardware when RX is idle. 0 = RX is busy. 1 = RX is idle. (Default)
[28]	**TXEMPTYF**	**Transmitter Empty Flag (Read Only)** This bit is set by hardware when TX FIFO (UART_DAT) is empty and the STOP bit of the last byte has been transmitted. 0 = TX FIFO is not empty or the STOP bit of the last byte has been not transmitted. 1 = TX FIFO is empty and the STOP bit of the last byte has been transmitted. **Note:** This bit is cleared automatically when TX FIFO is not empty or the last byte transmission has not completed.
[27:25]	**Reserved**	Reserved.
[24]	**TXOVIF**	**TX Overflow Error Interrupt Flag** If TX FIFO (UART_DAT) is full, an additional write to UART_DAT will cause this bit to logic 1. 0 = TX FIFO is not overflow. 1 = TX FIFO is overflow. **Note:** This bit can be cleared by writing “1” to it.
[23]	**TXFULL**	**Transmitter FIFO Full (Read Only)** This bit indicates TX FIFO full or not. 0 = TX FIFO is not full.

		1 = TX FIFO is full. **Note:** This bit is set when the number of usage in TX FIFO Buffer is equal to 16, otherwise it is cleared by hardware.
[22]	**TXEMPTY**	**Transmitter FIFO Empty (Read Only)** This bit indicates TX FIFO empty or not. 0 = TX FIFO is not empty. 1 = TX FIFO is empty. **Note:** When the last byte of TX FIFO has been transferred to Transmitter Shift Register, hardware sets this bit high. It will be cleared when writing data into UART_DAT (TX FIFO not empty).
[21:16]	**TXPTR**	**TX FIFO Pointer (Read Only)** This field indicates the TX FIFO Buffer Pointer. When CPU writes one byte into UART_DAT, TXPTR increases one. When one byte of TX FIFO is transferred to Transmitter Shift Register, TXPTR decreases one. The Maximum value shown in TXPTR is 31. When the using level of TX FIFO Buffer equal to 32, the TXFULL bit is set to 1 and TXPTR will show 0. As one byte of TX FIFO is transferred to Transmitter Shift Register, the TXFULL bit is cleared to 0 and TXPTR will show 31.
[15]	**RXFULL**	**Receiver FIFO Full (Read Only)** This bit initiates RX FIFO full or not. 0 = RX FIFO is not full. 1 = RX FIFO is full. **Note:** This bit is set when the number of usage in RX FIFO Buffer is equal to 32, otherwise it is cleared by hardware.
[14]	**RXEMPTY**	**Receiver FIFO Empty (Read Only)** This bit initiate RX FIFO empty or not. 0 = RX FIFO is not empty. 1 = RX FIFO is empty. **Note:** When the last byte of RX FIFO has been read by CPU, hardware sets this bit high. It will be cleared when UART receives any new data.
[13:8]	**RXPTR**	**RX FIFO Pointer (Read Only)** This field indicates the RX FIFO Buffer Pointer. When UART receives one byte from external device, RXPTR increases one. When one byte of RX FIFO is read by CPU, RXPTR decreases one. The Maximum value shown in RXPTR is 31. When the using level of RX FIFO Buffer equal to 32, the RXFULL bit is set to 1 and RXPTR will show 0. As one byte of RX FIFO is read by CPU, the RXFULL bit is cleared to 0 and RXPTR will show 31.
[7]	**Reserved**	Reserved.
[6]	**BIF**	**Break Interrupt Flag** This bit is set to logic 1 whenever the received data input (RX) is held in the “spacing state” (logic 0) for longer than a full word transmission time (that is, the total time of “start bit” + data bits + parity + stop bits). 0 = No Break interrupt is generated. 1 = Break interrupt is generated. **Note:** This bit can be cleared by writing “1” to it.
[5]	**FEF**	**Framing Error Flag** This bit is set to logic 1 whenever the received character does not have a valid “stop bit” (that is, the stop bit following the last data bit or parity bit is detected as logic 0). 0 = No framing error is generated. 1 = Framing error is generated. **Note:** This bit can be cleared by writing “1” to it.
[4]	**PEF**	**Parity Error Flag** This bit is set to logic 1 whenever the received character does not have a valid “parity bit”.

		0 = No parity error is generated. 1 = Parity error is generated. **Note:** This bit can be cleared by writing “1” to it.
[3]	**ADDRDETF**	**RS-485 Address Byte Detect Flag** 0 = Receiver detects a data that is not an address bit (bit 9 =’0’). 1 = Receiver detects a data that is an address bit (bit 9 =’1’). **Note 1:** This field is used for RS-485 function mode and ADDRDEN (UART_ALTCTL[15]) is set to 1 to enable Address detection mode. **Note 2:** This bit can be cleared by writing “1” to it.
[2]	**ABRDTOIF**	**Auto-baud Rate Detect Time-out Interrupt Flag** This bit is set to logic “1” in Auto-baud Rate Detect mode when the baud rate counter is overflow. 0 = Auto-baud rate counter is underflow. 1 = Auto-baud rate counter is overflow. **Note:** This bit can be cleared by writing “1” to it.
[1]	**ABRDIF**	**Auto-baud Rate Detect Interrupt Flag** This bit is set to logic “1” when auto-baud rate detect function is finished. 0 = Auto-baud rate detect function is not finished. 1 = Auto-baud rate detect function is finished. **Note:** This bit can be cleared by writing “1” to it.
[0]	**RXOVIF**	**RX Overflow Error Interrupt Flag** This bit is set when RX FIFO overflow. If the number of bytes of received data is greater than RX_FIFO (UART_DAT) size 32 bytes, this bit will be set. 0 = RX FIFO is not overflow. 1 = RX FIFO is overflow. **Note:** This bit can be cleared by writing “1” to it.

UART Interrupt Status Register (UART_INTSTS)

Register	Offset	R/W	Description	Reset Value
UART_INTSTS x=0,1..16	UARTx_BA+0x1C	R/W	UART Interrupt Status Register	0x0040_0002

31	30	29	28	27	26	25	24
ABRINT	TXENDINT	PBUFEINT	PTOINT	PMODINT	PRLSINT	Reserved	SWBEINT
23	22	21	20	19	18	17	16
Reserved	TXENDIF	PBUFEIF	PTOIF	PMODIF	PRLSIF	Reserved	SWBEIF
15	14	13	12	11	10	9	8
Reserved	WKINT	BUFEINT	RXTOINT	MODEMINT	RLSINT	THREINT	RDAINT
7	6	5	4	3	2	1	0
Reserved	WKIF	BUFEIF	RXTOIF	MODEMIF	RLSIF	THREIF	RDAIF

Bits	Description	
[31]	**ABRINT**	**Auto-baud Rate Interrupt Indicator (Read Only)** This bit is set if ABRIEN (UART_INTEN[18]) and ABRIF (UART_ALTCTL[17]) are both set to 1. 0 = No Auto-baud Rate interrupt is generated. 1 = The Auto-baud Rate interrupt is generated.
[30]	**TXENDINT**	**Transmitter Empty Interrupt Indicator (Read Only)** This bit is set if TXENDIEN (UART_INTEN[22]) and TXENDIF(UART_INTSTS[22]) are both set to 1. 0 = No Transmitter Empty interrupt is generated. 1 = Transmitter Empty interrupt is generated.
[29]	**PBUFEINT**	**PDMA Mode Buffer Error Interrupt Indicator (Read Only)** This bit is set if BUFERRIEN (UART_INTEN[5]) and HWBUFEIF (UART_INTSTS[21]) are both set to 1. 0 = No buffer error interrupt is generated in PDMA mode. 1 = Buffer error interrupt is generated in PDMA mode.
[28]	**PTOINT**	**PDMA Mode RX Time-out Interrupt Indicator (Read Only)** This bit is set if RXTOIEN (UART_INTEN[4]) and HWTOIF(UART_INTSTS[20]) are both set to 1. 0 = No RX time-out interrupt is generated in PDMA mode. 1 = RX time-out interrupt is generated in PDMA mode.
[27]	**PMODINT**	**PDMA Mode MODEM Status Interrupt Indicator (Read Only)** This bit is set if MODEMIEN (UART_INTEN[3]) and HWMODIF(UART_INTSTS[19]) are both set to 1. 0 = No Modem interrupt is generated in PDMA mode. 1 = Modem interrupt is generated in PDMA mode.
[26]	**PRLSINT**	**PDMA Mode Receive Line Status Interrupt Indicator (Read Only)** This bit is set if RLSIEN (UART_INTEN[2]) and HWRLSIF(UART_INTSTS[18]) are both set to 1. 0 = No RLS interrupt is generated in PDMA mode. 1 = RLS interrupt is generated in PDMA mode.
[25]	**Reserved**	Reserved.

[24]	**SWBEINT**	**Single-wire Bit Error Detect Interrupt Indicator (Read Only)** This bit is set if SWBEIEN (UART_INTEN[16]) and SWBEIF (UART_INTSTS[16]) are both set to 1. 0 = No Single-wire Bit Error Detection Interrupt generated. 1 = Single-wire Bit Error Detection Interrupt generated.
[23]	**Reserved**	Reserved.
[22]	**TXENDIF**	**Transmitter Empty Interrupt Flag** This bit is set when TX FIFO (UART_DAT) is empty and the STOP bit of the last byte has been transmitted (TXEMPTYF (UART_FIFOSTS[28]) is set). If TXENDIEN (UART_INTEN[22]) is enabled, the Transmitter Empty interrupt will be generated. 0 = No transmitter empty interrupt flag is generated. 1 = Transmitter empty interrupt flag is generated. **Note:** This bit is cleared automatically when TX FIFO is not empty or the last byte transmission has not completed.
[21]	**PBUFEIF**	**PDMA Mode Buffer Error Interrupt Flag (Read Only)** This bit is set when the TX or RX FIFO overflows (TXOVIF (UART_FIFOSTS [24]) or RXOVIF (UART_FIFOSTS[0]) is set). When BUFERRIF (UART_INTSTS[5]) is set, the transfer maybe is not correct. If BUFERRIEN (UART_INTEN [5]) is enabled, the buffer error interrupt will be generated. 0 = No buffer error interrupt flag is generated in PDMA mode. 1 = Buffer error interrupt flag is generated in PDMA mode. **Note:** This bit is cleared when both TXOVIF (UART_FIFOSTS[24]) and RXOVIF (UART_FIFOSTS[0]) are cleared.
[20]	**PTOIF**	**PDMA Mode RX Time-out Interrupt Flag (Read Only)** This bit is set when the RX FIFO is not empty and no activities occurred in the RX FIFO and the time-out counter equal to TOIC (UART_TOUT[7:0]). If RXTOIEN (UART_INTEN [4]) is enabled, the RX time-out interrupt will be generated . 0 = No RX time-out interrupt flag is generated in PDMA mode. 1 = RX time-out interrupt flag is generated in PDMA mode. **Note:** This bit is read only and user can read UART_DAT (RX is in active) to clear it.
[19]	**PMODIF**	**PDMA Mode MODEM Interrupt Flag (Read Only)** This bit is set when the nCTS pin has state change (CTSDETF (UART_MODEMSTS [0] =1)). If MODEMIEN (UART_INTEN [3]) is enabled, the Modem interrupt will be generated. 0 = No Modem interrupt flag is generated in PDMA mode. 1 = Modem interrupt flag is generated in PDMA mode. **Note:** This bit is read only and reset to 0 when the bit CTSDETF (UART_MODEMSTS[0]) is cleared by writing 1 on CTSDETF (UART_MODEMSTS [0]).
[18]	**PRLSIF**	**PDMA Mode Receive Line Status Flag (Read Only)** This bit is set when the RX receive data have parity error, frame error or break error (at least one of 3 bits, BIF (UART_FIFOSTS[6]), FEF (UART_FIFOSTS[5]) and PEF (UART_FIFOSTS[4]) is set). If RLSIEN (UART_INTEN [2]) is enabled, the RLS interrupt will be generated. 0 = No RLS interrupt flag is generated in PDMA mode. 1 = RLS interrupt flag is generated in PDMA mode. **Note 1:** In RS-485 function mode, this field include “receiver detect any address byte received address byte character (bit9 = ‘1’) bit". **Note 2:** In UART function mode, this bit is read only and reset to 0 when all bits of BIF(UART_FIFOSTS[6]), FEF(UART_FIFOSTS[5]) and PEF(UART_FIFOSTS[4]) are cleared. **Note 3:** In RS-485 function mode, this bit is read only and reset to 0 when all bits of BIF(UART_FIFOSTS[6]), FEF(UART_FIFOSTS[5]), PEF(UART_FIFOSTS[4]) and ADDRDETF (UART_FIFOSTS[3]) are cleared.
[17]	**Reserved**	Reserved.
[16]	**SWBEIF**	**Single-wire Bit Error Detection Interrupt Flag** This bit is set when the single wire bus state not equals to UART controller TX state in Single-wire mode.

		0 = No single-wire bit error detection interrupt flag is generated. 1 = Single-wire bit error detection interrupt flag is generated. **Note 1:** This bit is active when FUNCSEL (UART_FUNCSEL[2:0]) is select UART Single-wire mode. **Note 2:** This bit can be cleared by writing "1" to it.
[15]	**Reserved**	Reserved.
[14]	**WKINT**	**UART Wake-up Interrupt Indicator (Read Only)** This bit is set if WKIEN (UART_INTEN[6]) and WKIF (UART_INTSTS[6]) are both set to 1. 0 = No UART wake-up interrupt is generated. 1 = UART wake-up interrupt is generated.
[13]	**BUFEINT**	**Buffer Error Interrupt Indicator (Read Only)** This bit is set if BUFERRIEN(UART_INTEN[5]) and BUFERRIF(UART_ INTSTS[5]) are both set to 1. 0 = No buffer error interrupt is generated. 1 = Buffer error interrupt is generated.
[12]	**RXTOINT**	**RX Time-out Interrupt Indicator (Read Only)** This bit is set if RXTOIEN (UART_INTEN[4]) and RXTOIF(UART_INTSTS[4]) are both set to 1. 0 = No RX time-out interrupt is generated. 1 = RX time-out interrupt is generated.
[11]	**MODEMINT**	**MODEM Status Interrupt Indicator (Read Only)** This bit is set if MODEMIEN(UART_INTEN[3]) and MODEMIF(UART_INTSTS[3]) are both set to 1 0 = No Modem interrupt is generated. 1 = Modem interrupt is generated..
[10]	**RLSINT**	**Receive Line Status Interrupt Indicator (Read Only)** This bit is set if RLSIEN (UART_INTEN[2]) and RLSIF(UART_INTSTS[2]) are both set to 1. 0 = No RLS interrupt is generated. 1 = RLS interrupt is generated.
[9]	**THREINT**	**Transmit Holding Register Empty Interrupt Indicator (Read Only)** This bit is set if THREIEN (UART_INTEN[1]) and THREIF(UART_INTSTS[1]) are both set to 1. 0 = No THRE interrupt is generated. 1 = THRE interrupt is generated.
[8]	**RDAINT**	**Receive Data Available Interrupt Indicator (Read Only)** This bit is set if RDAIEN (UART_INTEN[0]) and RDAIF (UART_INTSTS[0]) are both set to 1. 0 = No RDA interrupt is generated. 1 = RDA interrupt is generated.
[7]	**Reserved**	Reserved.
[6]	**WKIF**	**UART Wake-up Interrupt Flag (Read Only)** This bit is set when TOUTWKF (UART_WKSTS[4]), RS485WKF (UART_WKSTS[3]), RFRTWKF (UART_WKSTS[2]), DATWKF (UART_WKSTS[1]) or CTSWKF(UART_WKSTS[0]) is set to 1. 0 = No UART wake-up interrupt flag is generated. 1 = UART wake-up interrupt flag is generated. **Note:** This bit is cleared if all of TOUTWKF, RS485WKF, RFRTWKF, DATWKF and CTSWKF are cleared to 0 by writing 1 to the corresponding interrupt flag.
[5]	**BUFEIF**	**Buffer Error Interrupt Flag (Read Only)** This bit is set when the TX FIFO or RX FIFO overflows (TXOVIF (UART_FIFOSTS[24]) or RXOVIF (UART_FIFOSTS[0]) is set). When BUFERRIF (UART_INTSTS[5]) is set, the transfer is not correct. If BUFERRIEN (UART_INTEN [5]) is enabled, the buffer error interrupt will be generated. 0 = No buffer error interrupt flag is generated.

		1 = Buffer error interrupt flag is generated. **Note:** This bit is cleared if both of RXOVIF(UART_FIFOSTS[0]) and TXOVIF(UART_FIFOSTS[24]) are cleared to 0 by writing 1 to RXOVIF(UART_FIFOSTS[0]) and TXOVIF(UART_FIFOSTS[24]).
[4]	**RXTOIF**	**RX Time-out Interrupt Flag (Read Only)** This bit is set when the RX FIFO is not empty and no activities occurred in the RX FIFO and the time-out counter equal to TOIC (UART_TOUT[7:0]). If RXTOIEN (UART_INTEN [4]) is enabled, the RX time-out interrupt will be generated. 0 = No RX time-out interrupt flag is generated. 1 = RX time-out interrupt flag is generated. **Note:** This bit is read only and user can read UART_DAT (RX is in active) to clear it.
[3]	**MODEMIF**	**MODEM Interrupt Flag (Read Only)** This bit is set when the nCTS pin has state change (CTSDETF (UART_MODEMSTS[0]) = 1). If MODEMIEN (UART_INTEN [3]) is enabled, the Modem interrupt will be generated. 0 = No Modem interrupt flag is generated. 1 = Modem interrupt flag is generated. **Note:** This bit is read only and reset to 0 when bit CTSDETF is cleared by a write 1 on CTSDETF(UART_MODEMSTS[0]).
[2]	**RLSIF**	**Receive Line Interrupt Flag (Read Only)** This bit is set when the RX receive data have parity error, frame error or break error (at least one of 3 bits, BIF(UART_FIFOSTS[6]), FEF(UART_FIFOSTS[5]) and PEF(UART_FIFOSTS[4]), is set). If RLSIEN (UART_INTEN [2]) is enabled, the RLS interrupt will be generated. 0 = No RLS interrupt flag is generated. 1 = RLS interrupt flag is generated. **Note 1:** In RS-485 function mode, this field is set include “receiver detect and received address byte character (bit9 = ‘1’) bit". At the same time, the bit of ADDRDETF (UART_FIFOSTS[3]) is also set. **Note 2:** This bit is read only and reset to 0 when all bits of BIF (UART_FIFOSTS[6]), FEF(UART_FIFOSTS[5]) and PEF(UART_FIFOSTS[4]) are cleared. **Note 3:** In RS-485 function mode, this bit is read only and reset to 0 when all bits of BIF (UART_FIFOSTS[6]), FEF(UART_FIFOSTS[5]), PEF(UART_FIFOSTS[4]) and ADDRDETF (UART_FIFOSTS[3]) are cleared.
[1]	**THREIF**	**Transmit Holding Register Empty Interrupt Flag** This bit is set when the last data of TX FIFO is transferred to Transmitter Shift Register. If THREIEN (UART_INTEN[1]) is enabled, the THRE interrupt will be generated. 0 = No THRE interrupt flag is generated. 1 = THRE interrupt flag is generated. **Note:** This bit is read only and it will be cleared when writing data into UART_DAT (TX FIFO not empty).
[0]	**RDAIF**	**Receive Data Available Interrupt Flag** When the number of bytes in the RX FIFO equals the RFITL then the RDAIF(UART_INTSTS[0]) will be set. If RDAIEN (UART_INTEN [0]) is enabled, the RDA interrupt will be generated. 0 = No RDA interrupt flag is generated. 1 = RDA interrupt flag is generated. **Note:** This bit is read only and it will be cleared when the number of unread bytes of RX FIFO drops below the threshold level (RFITL(UART_FIFO[7:4]).

UART Time-out Register (UART_TOUT)

Register	Offset	R/W	Description	Reset Value
UART_TOUT **x=0,1..16**	UARTx_BA+0x20	R/W	UART Time-out Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DLY							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TOIC							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:8]	**DLY**	**TX Delay Time Value** This field is used to programming the transfer delay time between the last stop bit and next start bit. The unit is bit time.
[7:0]	**TOIC**	**Time-out Interrupt Comparator** The time-out counter resets and starts counting (the counting clock = baud rate) whenever the RX FIFO receives a new data word if time out counter is enabled by setting TOCNTEN (UART_INTEN[11]). Once the content of time-out counter is equal to that of time-out interrupt comparator (TOIC (UART_TOUT[7:0])), a receiver time-out interrupt (RXTOINT(UART_INTSTS[12])) is generated if RXTOIEN (UART_INTEN [4]) enabled. A new incoming data word or RX FIFO empty will clear RXTOIF (UART_INTSTS[4]). In order to avoid receiver time-out interrupt generation immediately during one character is being received, TOIC value should be set between 40 and 255. So, for example, if TOIC is set with 40, the time-out interrupt is generated after four characters are not received when 1 stop bit and no parity check is set for UART transfer.

UART Baud Rate Divider Register (UART_BAUD)

Register	Offset	R/W	Description	Reset Value
UART_BAUD **x=0,1..16**	UARTx_BA+0x24	R/W	UART Baud Rate Divider Register	0x0F00_0000

31	30	29	28	27	26	25	24
Reserved		BAUDM1	BAUDM0	EDIVM1			
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
BRD							
7	6	5	4	3	2	1	0
BRD							

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29]	**BAUDM1**	**BAUD Rate Mode Selection Bit 1** This bit is baud rate mode selection bit 1. UART provides three baud rate calculation modes. This bit combines with BAUDM0 (UART_BAUD[28]) to select baud rate calculation mode. The detail description is shown in Table 6.24-4. **Note:** In IrDA mode must be operated in mode 0.
[28]	**BAUDM0**	**BAUD Rate Mode Selection Bit 0** This bit is baud rate mode selection bit 0. UART provides three baud rate calculation modes. This bit combines with BAUDM1 (UART_BAUD[29]) to select baud rate calculation mode. The detail description is shown in Table 6.24-4.
[27:24]	**EDIVM1**	**Extra Divider for BAUD Rate Mode 1** This field is used for baud rate calculation in mode 1 and has no effect for baud rate calculation in mode 0 and mode 2. The detail description is shown in Table 6.24-4.
[23:16]	**Reserved**	Reserved.
[15:0]	**BRD**	**Baud Rate Divider** The field indicates the baud rate divider. This filed is used in baud rate calculation. The detail description is shown in Table 6.24-4.

UART IrDA Control Register (UART_IRDA)

Register	Offset	R/W	Description	Reset Value
UART_IRDA **x=0,1..16**	UARTx_BA+0x28	R/W	UART IrDA Control Register	0x0000_0040

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**RXINV**	**TXINV**	**Reserved**			**TXEN**	**Reserved**

Bits	Description	
[31:7]	**Reserved**	Reserved.
[6]	**RXINV**	**IrDA Inverse Receive Input Signal** 0 = None inverse receiving input signal. 1 = Inverse receiving input signal. (Default) **Note 1:** Before setting this bit, TXRXDIS (UART_FUNCSEL[3]) should be set then waited for TXRXACT (UART_FIFOSTS[31]) is cleared. When the configuration is done, cleared TXRXDIS (UART_FUNCSEL[3]) to activate UART controller. **Note 2:** This bit is valid when FUNCSEL (UART_FUNCSEL[2:0]) is select IrDA function.
[5]	**TXINV**	**IrDA Inverse Transmitting Output Signal** 0 = None inverse transmitting signal. (Default). 1 = Inverse transmitting output signal. **Note 1:** Before setting this bit, TXRXDIS (UART_FUNCSEL[3]) should be set then waited for TXRXACT (UART_FIFOSTS[31]) is cleared. When the configuration is done, cleared TXRXDIS (UART_FUNCSEL[3]) to activate UART controller. **Note 2:** This bit is valid when FUNCSEL (UART_FUNCSEL[2:0]) is select IrDA function.
[4:2]	**Reserved**	Reserved.
[1]	**TXEN**	**IrDA Receiver/Transmitter Selection Enable Bit** 0 = IrDA Transmitter Disabled and Receiver Enabled. (Default) 1 = IrDA Transmitter Enabled and Receiver Disabled.
[0]	**Reserved**	Reserved.

UART Alternate Control/Status Register (UART_ALTCTL)

Register	Offset	R/W	Description	Reset Value
UART_ALTCTL x=0,1..16	UARTx_BA+0x2C	R/W	UART Alternate Control/Status Register	0x0000_000C

31	30	29	28	27	26	25	24
ADDRMV							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			**ABRDBITS**		**ABRDEN**	**ABRIF**	**Reserved**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ADDRDEN	**Reserved**				**RS485AUD**	**RS485AAD**	**RS485NMM**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:24]	**ADDRMV**	**Address Match Value** This field contains the RS-485 address match values. **Note:** This field is used for RS-485 auto address detection mode.
[23:21]	**Reserved**	Reserved.
[20:19]	**ABRDBITS**	**Auto-baud Rate Detect Bit Length** 00 = 1-bit time from Start bit to the 1st rising edge. The input pattern shall be 0x01. 01 = 2-bit time from Start bit to the 1st rising edge. The input pattern shall be 0x02. 10 = 4-bit time from Start bit to the 1st rising edge. The input pattern shall be 0x08. 11 = 8-bit time from Start bit to the 1st rising edge. The input pattern shall be 0x80. **Note :** The calculation of bit number includes the START bit.
[18]	**ABRDEN**	**Auto-baud Rate Detect Enable Bit** 0 = Auto-baud rate detect function Disabled. 1 = Auto-baud rate detect function Enabled. **Note :** This bit is cleared automatically after auto-baud detection is finished.
[17]	**ABRIF**	**Auto-baud Rate Interrupt Flag (Read Only)** This bit is set when auto-baud rate detection function finished or the auto-baud rate counter was overflow and if ABRIEN(UART_INTEN [18]) is set then the auto-baud rate interrupt will be generated. 0 = No auto-baud rate interrupt flag is generated. 1 = Auto-baud rate interrupt flag is generated. **Note:** This bit is read only, but it can be cleared by writing “1” to ABRDTOIF (UART_FIFOSTS[2]) and ABRDIF(UART_FIFOSTS[1]).
[16]	**Reserved**	Reserved.
[15]	**ADDRDEN**	**RS-485 Address Detection Enable Bit** This bit is used to enable RS-485 Address Detection mode. 0 = Address detection mode Disabled. 1 = Address detection mode Enabled.

		Note: This bit is used for RS-485 any operation mode.
[14:11]	**Reserved**	Reserved.
[10]	**RS485AUD**	**RS-485 Auto Direction Function** 0 = RS-485 Auto Direction Operation function (AUD) Disabled. 1 = RS-485 Auto Direction Operation function (AUD) Enabled. **Note:** It can be active with RS-485_AAD or RS-485_NMM operation mode.
[9]	**RS485AAD**	**RS-485 Auto Address Detection Operation Mode** 0 = RS-485 Auto Address Detection Operation mode (AAD) Disabled. 1 = RS-485 Auto Address Detection Operation mode (AAD) Enabled. **Note:** It cannot be active with RS-485_NMM operation mode.
[8]	**RS485NMM**	**RS-485 Normal Multi-drop Operation Mode** 0 = RS-485 Normal Multi-drop Operation mode (NMM) Disabled. 1 = RS-485 Normal Multi-drop Operation mode (NMM) Enabled. **Note:** It cannot be active with RS-485_AAD operation mode.
[7:0]	**Reserved**	Reserved.

UART Function Select Register (UART_FUNCSEL)

Register	Offset	R/W	Description	Reset Value
UART_FUNCSEL **x=0,1..16**	UARTx_BA+0x30	R/W	UART Function Select Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**TXRXDIS**	**FUNCSEL**		

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3]	**TXRXDIS**	**TX and RX Disable Bit** Setting this bit can disable TX and RX. 0 = TX and RX Enabled. 1 = TX and RX Disabled. **Note:** The TX and RX will not be disabled immediately when this bit is set. The TX and RX complete current task before TX and RX are disabled. When TX and RX are disabled, the TXRXACT (UART_FIFOSTS[31]) is cleared.
[2:0]	**FUNCSEL**	**Function Select** 000 = UART function. 010 = IrDA function. 011 = RS-485 function. 100 = UART Single-wire function. Others = Reserved.

UART Baud Rate Compensation Register (UART_BRCOMP)

Register	Offset	R/W	Description	Reset Value
UART_BRCOMP x=0,1..16	UARTx_BA+0x3C	R/W	UART Baud Rate Compensation Register	0x0000_0000

31	30	29	28	27	26	25	24
BRCOMPD	**Reserved**						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**BRCOMP**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BRCOMP							

Bits	Description	
[31]	**BRCOMPD**	**Baud Rate Compensation Decrease** 0 = Positive (increase one module clock) compensation for each compensated bit. 1 = Negative (decrease one module clock) compensation for each compensated bit.
[30:9]	**Reserved**	Reserved.
[8:0]	**BRCOMP**	**Baud Rate Compensation Patten** These 9-bits are used to define the relative bit is compensated or not. BRCOMP[7:0] is used to define the compensation of DAT (UART_DAT[7:0]) and BRCOM[8] is used to define PARITY (UART_DAT[8]).

UART Wake-up Control Register (UART_WKCTL)

Register	Offset	R/W	Description	Reset Value
UART_WKCTL **x=0,1..16**	UARTx_BA+0x40	R/W	UART Wake-up Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**WKTOUTEN**	**WKAADEN**	**WKRFRTEN**	**WKDATEN**	**WKCTSEN**

Bits	Description	
[31:5]	**Reserved**	Reserved.
[4]	**WKTOUTEN**	**Received Data FIFO Reached Threshold Time-out Wake-up Enable Bit** 0 = Received Data FIFO reached threshold time-out wake-up system function Disabled. 1 = Received Data FIFO reached threshold time-out wake-up system function Enabled. **Note 1:** When the system is in Power-down mode, Received Data FIFO reached threshold time-out will wake up system from Power-down mode. **Note 2:** It is suggested the function is enabled when the WKRFRTEN (UART_WKCTL[2]) is set to 1.
[3]	**WKAADEN**	**RS-485 Address Match Wake-up Enable Bit** 0 = RS-485 Address Match (AAD mode) wake-up system function Disabled. 1 = RS-485 Address Match (AAD mode) wake-up system function Enabled. **Note 1:** When the system is in .Power-down mode, RS-485 Address Match will wake -up system from Power-down mode. **Note 2:** This bit is used for RS-485 Auto Address Detection (AAD) mode in RS-485 function mode and ADDRDEN (UART_ALTCTL[15]) is set to 1.
[2]	**WKRFRTEN**	**Received Data FIFO Reached Threshold Wake-up Enable Bit** 0 = Received Data FIFO reached threshold wake-up system function Disabled. 1 = Received Data FIFO reached threshold wake-up system function Enabled. **Note:** When the system is in Power-down mode, Received Data FIFO reached threshold will wake-up system from Power-down mode.
[1]	**WKDATEN**	**Incoming Data Wake-up Enable Bit** 0 = Incoming data wake-up system function Disabled. 1 = Incoming data wake-up system function Enabled. **Note:** When the system is in Power-down mode, incoming data will wake-up system from Power-down mode.
[0]	**WKCTSEN**	**nCTS Wake-up Enable Bit** 0 = nCTS Wake-up system function Disabled. 1 = nCTS Wake-up system function Enabled. **Note:** When the system is in Power-down mode, an external nCTS change will wake up system from

		Power-down mode.

UART Wake-up Status Register (UART_WKSTS)

Register	Offset	R/W	Description	Reset Value
UART_WKSTS **x=0,1..16**	UARTx_BA+0x44	R/W	UART Wake-up Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**TOUTWKF**	**RS485WKF**	**RFRTWKF**	**DATWKF**	**CTSWKF**

Bits	Description	
[31:5]	**Reserved**	Reserved.
[4]	**TOUTWKF**	**Received Data FIFO Threshold Time-out Wake-up Flag** This bit is set if chip wake-up from power-down state by Received Data FIFO Threshold Time-out wake-up. 0 = Chip stays in power-down state. 1 = Chip wake-up from power-down state by Received Data FIFO reached threshold time-out. **Note 1:** If WKTOUTEN (UART_WKCTL[4]) is enabled, the Received Data FIFO reached threshold time-out wake-up cause this bit is set to '1'. **Note 2:** This bit can be cleared by writing '1' to it.
[3]	**RS485WKF**	**RS-485 Address Match Wake-up Flag** This bit is set if chip wake-up from power-down state by RS-485 Address Match (AAD mode). 0 = Chip stays in power-down state. 1 = Chip wake-up from power-down state by RS-485 Address Match (AAD mode) wake-up. **Note 1:** If WKRS485EN (UART_WKCTL[3]) is enabled, the RS-485 Address Match (AAD mode) wake-up cause this bit is set to '1'. **Note 2:** This bit can be cleared by writing '1' to it.
[2]	**RFRTWKF**	**Received Data FIFO Reached Threshold Wake-up Flag** This bit is set if chip wake-up from power-down state by Received Data FIFO reached threshold wake-up . 0 = Chip stays in power-down state. 1 = Chip wake-up from power-down state by Received Data FIFO Reached Threshold wake-up. **Note 1:** If WKRFRTEN (UART_WKCTL[2]) is enabled, the Received Data FIFO Reached Threshold wake-up cause this bit is set to '1'. **Note 2:** This bit can be cleared by writing '1' to it.
[1]	**DATWKF**	**Incoming Data Wake-up Flag** This bit is set if chip wake-up from power-down state by data wake-up. 0 = Chip stays in power-down state. 1 = Chip wake-up from power-down state by Incoming Data wake-up.

		Note 1: If WKDATEN (UART_WKCTL[1]) is enabled, the Incoming Data wake-up cause this bit is set to '1'. **Note 2:** This bit can be cleared by writing '1' to it.
[0]	**CTSWKF**	**nCTS Wake-up Flag** This bit is set if chip wake-up from power-down state by nCTS wake-up. 0 = Chip stays in power-down state. 1 = Chip wake-up from power-down state by nCTS wake-up. **Note 1:** If WKCTSEN (UART_WKCTL[0]) is enabled, the nCTS wake-up cause this bit is set to '1'. **Note 2:** This bit can be cleared by writing '1' to it.

UART Incoming Data Wake-up Compensation Register (UART_DWKCOMP)

Register	Offset	R/W	Description	Reset Value
UART_DWKCOMP x=0,1..16	UARTx_BA+0x48	R/W	UART Incoming Data Wake-up Compensation Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
STCOMP							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
STCOMP							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**STCOMP**	**Start Bit Compensation Value** These bits field indicate how many clock cycle selected by UART_CLK do the UART controller can get the 1st bit (start bit) when the device is woken up from Power-down mode. **Note:** It is valid only when WKDATEN (UART_WKCTL[1]) is set.

6.25 Smart Card Host Interface (SC)

6.25.1 Overview

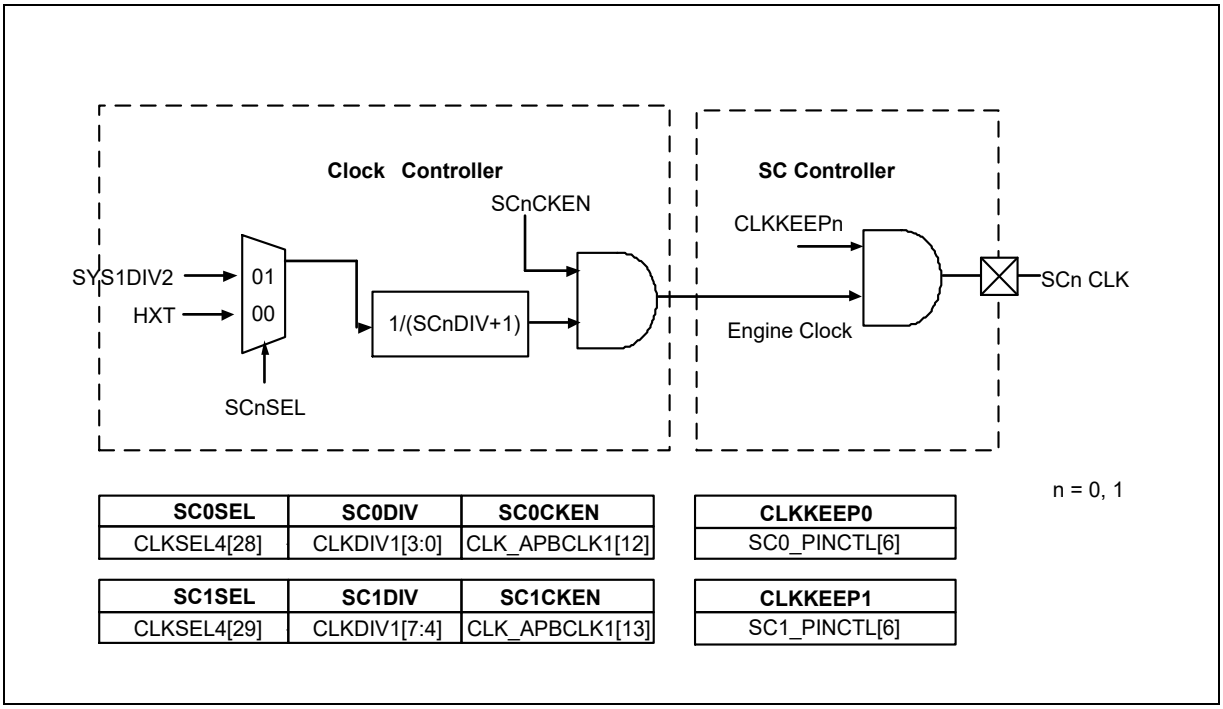
The Smart Card Interface controller (SC controller) is based on ISO/IEC 7816-3 standard and fully compliant with PC/SC Specifications. It also provides status of card insertion/removal. It can also be set as UART mode to communicate with other devices.

6.25.2 Features

- ISO 7816-3 T = 0, T = 1 compliant
- EMV2000 compliant
- Two ISO 7816-3 ports
- Separates receive/transmit 4-byte entry FIFO for data payloads
- Programmable transmission clock frequency
- Programmable receiver buffer trigger level
- Programmable guard time selection (11 ETU ~ 267 ETU)
- One 24-bit timer and two 8-bit timers for Answer to Request (ATR) and waiting times processing
- Supports auto direct / inverse convention function
- Supports transmitter and receiver error retry and error number limiting function
- Supports hardware activation sequence process, and the time between PWR on and CLK start is configurable
- Supports hardware warm reset sequence process
- Supports hardware deactivation sequence process
- Supports hardware auto deactivation sequence when detected the card removal
- Supports UART mode
  - Full duplex, asynchronous communications
  - Separates receiving / transmitting 4-byte entry FIFO for data payloads
  - Supports programmable baud rate generator
  - Supports programmable receiver buffer trigger level
  - Programmable transmitting data delay time between the last stop bit leaving the TX-FIFO and the de-assertion by setting EGT (SCn_EGT[7:0])
  - Programmable even, odd or no parity bit generation and detection
  - Programmable stop bit, 1- or 2- stop bit generation

6.25.3 Block Diagram

The SC clock control and block diagram are shown in Figure 6.25-1 and Figure 6.25-2. The SC controller is completely asynchronous design with two clock domains, PCLK and SC engine clock. Note that the PCLK should be higher than or equal to the frequency of SC engine clock.



SC0SEL	SC0DIV	SC0CKEN
CLKSEL4[28]	CLKDIV1[3:0]	CLK_APBCLK1[12]

CLKKEEP0
SC0_PINCTL[6]

SC1SEL	SC1DIV	SC1CKEN
CLKSEL4[29]	CLKDIV1[7:4]	CLK_APBCLK1[13]

CLKKEEP1
SC1_PINCTL[6]

Figure 6.25-1 SC Clock Control Diagram (7-bit Pre-scale Counter in Clock Controller)

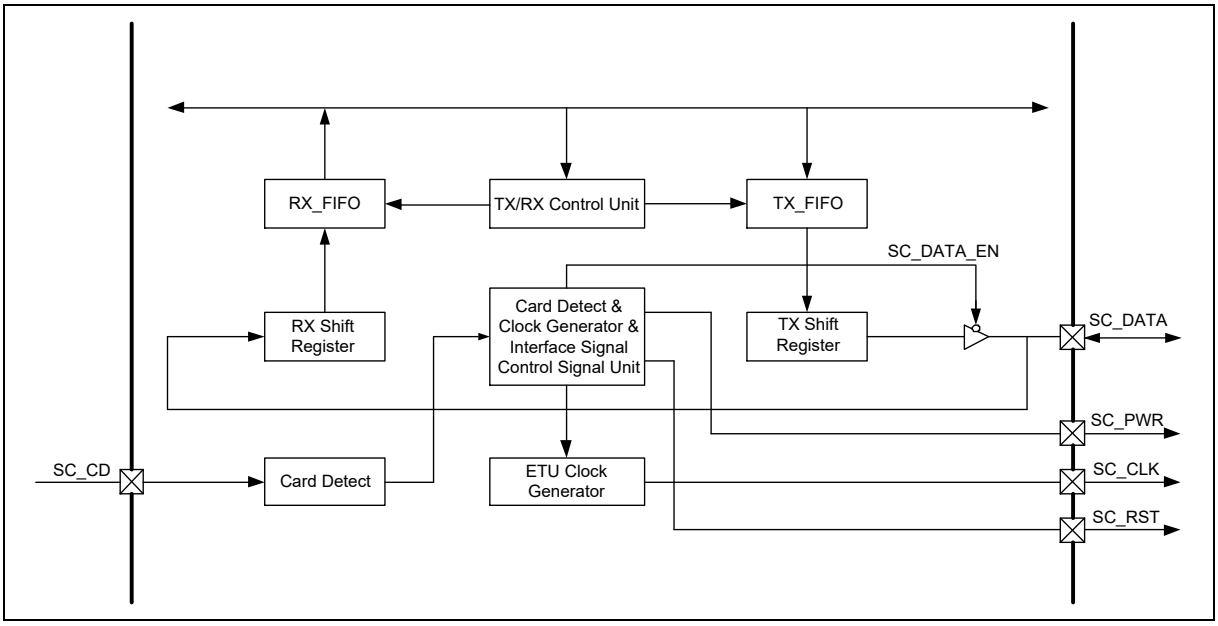


Figure 6.25-2 SC Controller Block Diagram

6.25.4 Basic Configuration

The SC Host Controller Pin description is shown in Table 6.25-1.

Pin	Type	Description
SCn_DATA	Bi-direction	SC Host Controller DATA
SCn_CD	Input	SC Host Controller Card Detect
SCn_PWR	Output	SC Host Controller Power ON/OFF
SCn_CLK	Output	SC Host Controller Clock
SCn_RST	Output	SC Host Controller Reset

Table 6.25-1 SC Host Controller Pin Description

The UART Mode Pin description is shown in Table 6.25-2:

Pin	Type	Description
SCn_DATA	Input	UART Receive Data
SCn_CLK	Output	UART Transmit Data

Table 6.25-2 UART Pin Description

6.25.4.1 SC0 Basic Configuration

- Clock Source Configuration
  - Select the source of SC0 peripheral clock on SC0SEL (CLK_CLKSEL4[28]).
  - Select the clock divider number of SC0 peripheral clock on SC0DIV(CLK_CLKDIV1[3:0]).
  - Enable SC0 peripheral clock in SC0CKEN (CLK_APBCLK1[12]).
- Reset Configuration
  - Reset SC0 controller in SC0RST (SYS_IPRST2[0]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
SC0	SC0_CLK	PI.1	MFP7
		PL.8	MFP9
		PJ.0	MFP8
		PK.12	MFP8
		PF.6	MFP13
		PF.10	MFP9
	SC0_DAT	PI.2	MFP7
		PL.9	MFP9
		PJ.1	MFP8
		PK.5	MFP8
		PK.13	MFP8

		PF.7	MFP13
		PF.11	MFP9
	SC0_PWR	PK.2	MFP7
		PL.11	MFP9
		PJ.3	MFP8
		PK.7	MFP8
		PK.15	MFP8
		PF.9	MFP13
		PF.13	MFP9
	SC0_RST	PI.3	MFP7
		PL.10	MFP9
		PJ.2	MFP8
		PK.6	MFP8
		PK.14	MFP8
		PF.8	MFP13
		PF.12	MFP9
	SC0_nCD	PI.0	MFP7
		PL.5	MFP9
		PN.15	MFP8
		PB.9	MFP13

6.25.4.2 SC1 Basic Configuration

- Clock Source Configuration
  - Select the source of SC1 peripheral clock on SC1SEL (CLK_CLKSEL4 [29]).
  - Select the clock divider number of SC1 peripheral clock on SC1DIV (CLK_CLKDIV1 [7:4]).
  - Enable SC1 peripheral clock in SC1CKEN (CLK_APBCLK1 [13]).
- Reset Configuration
  - Reset SC1 controller in SC1RST (SYS_IPRST2 [1]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
SC1	SC1_CLK	PL.12	MFP7
		PG.4	MFP10
		PJ.12	MFP7
		PD.8	MFP10
		PL.0	MFP9

	SC1_DAT	PL.13	MFP7
		PG.5	MFP10
		PJ.13	MFP7
		PD.9	MFP10
		PL.1	MFP9
	SC1_PWR	PL.15	MFP7
		PG.7	MFP10
		PJ.15	MFP7
		PD.11	MFP10
		PL.3	MFP9
	SC1_RST	PL.14	MFP7
		PG.6	MFP10
		PJ.14	MFP7
		PD.10	MFP10
		PL.2	MFP9
	SC1_nCD	PM.12	MFP7
		PK.3	MFP7
		PD.7	MFP10
		PL.4	MFP9

6.25.5 Functional Description

Basically, the smart card interface acts as a half-duplex asynchronous communication port and its data format is composed of ten consecutive bits which is shown in SC Data Character.

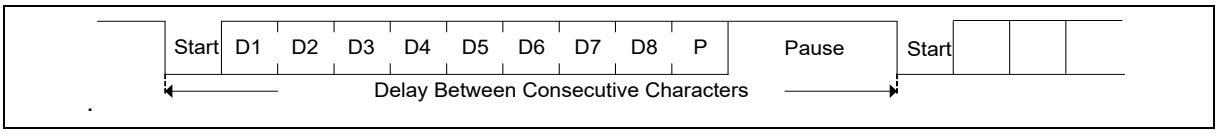


Figure 6.25-3 SC Data Character

The Smart Card Interface controller supports hardware activation, warm reset and deactivation sequence. The activation, warm reset and deactivation sequence is described as follows.

6.25.5.1 Smart Card Pin Configuration

The Smart Card Interface pin status can be observed by polling following registers.

- SCn_RST is an output pin. Its status can be observed by RSTSTS (SCn_PINCTL[18]). Programming RSTEN (SCn_PINCTL[1]) ‘0’ or ‘1’ can drive this output pin low or high.
- SCn_PWR is an output pin. Its status can be observed by PWRSTS (SCn_PINCTL[17]). Programming PWREN (SCn_PINCTL[0]) to ‘0’ or ‘1’ can drive this output pin low or high. PWRINV (SCn_PINCTL[11]) can inverse the SCn_PWR output. User must select PWRINV (SCn_PINCTL[11]) before smart card is enabled by SCEN (SCn_CTL[0]).
- SCn_DATA is a bidirectional pin, DATASTS(SCn_PINCTL[16]) shows the pin status when

SC is receiving data. Programming SCDATA (SCn_PINCTL[9]) to '0' or '1' can drive this pin output low or high.

- SCn_CLK is an output pin. It outputs Smart card clock SCn CLK. Programming CLKKEEP (SCn_PINCTL[6]) '0' or '1' to disable or enable this pin. Programming CSTOPLV (SCn_PINCTL[5]) can determine the SCn_CLK is stopped at high or low when this pin is disable.
- SCn_CD(Card Detect Pin) state represent the status of the card is inserted or not. SCn_CD pin status can be observed by CDPINSTS(SC_CDPINSTS[13]). SCn_CD related function can be set by CDLV(SC_CTL[26]), CDDBSEL(SC_CTL[25:24]), CDIF(SC_INTSTS[7]), CDIEN(SC_INTEN[7]).
- CDLV(SC_CTL[26]) determines what kind of pin level change represents the card insertion. CDDBSEL(SC_CTL[25:24]) determines the de-bounce cycles. When the card status CINSERT (SCn_STATUS[12]) or CREMOVE (SCn_STATUS[11]) is detected, CDIF will set to 1. If CDIEN is enable, SC will deliver an interrupt to CPU when CDIF is set to 1. Card Detect Pin is recommended setting before enabling SC.

6.25.5.2 Activation, Warm Reset and Deactivation Sequence

Activation

The activation sequence is shown in Figure 6.25-4:

1. Set SCn_RST to low by programming RSTEN (SCn_PINCTL[1]) to '0', and wait SYNC (SCn_PINCTL[31]) is cleared to 0.
2. Set SCn_PWR at high level by programming PWREN (SCn_PINCTL[0]) to '1' and SCn_DATA at high level (reception mode) by programming SCDATA (SCn_PINCTL[9]) to '1', and wait SYNC(SCn_PINCTL[31]) is cleared to 0.
3. Enable SCn_CLK clock by programming CLKKEEP (SCn_PINCTL[6]) to '1', and wait SYNC(SCn_PINCTL[31]) is cleared to 0.
4. De-assert SCn_RST to high by programming RSTEN (SCn_PINCTL[1]) to '1', and wait SYNC(SCn_PINCTL[31]) is cleared to 0.

The activation sequence can be controlled in two ways. The procedure is shown as follows:

- Software Timing Control:

  Set SCn_PINCTL and SCn_TMRCTLx (x = 0, 1, 2) to process the activation sequence. SCn_PWR, SCn_CLK, SCn_RST and SCn_DATA pin state can be programmed by SCn_PINCTL. The programming method is shown in activation sequence. The activation sequence timing can be controlled by setting SCn_TMRCTLx (x = 0, 1, 2). This programming procedure provides user with a flexible timing setting for activation sequence.

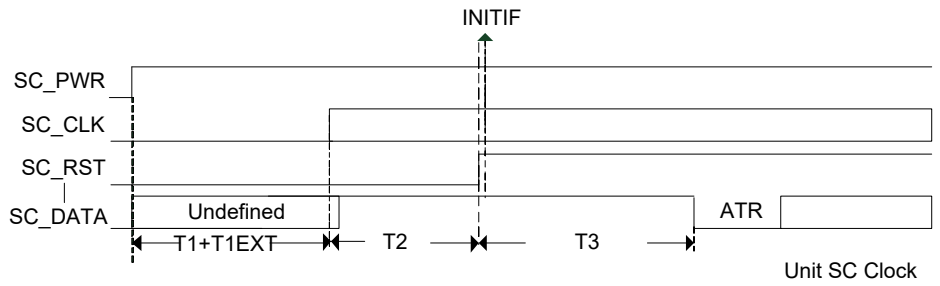
- Hardware Timing Control:

  Set ACTEN (SCn_ALTCTL[3]) to '1' and the interface will perform the activation sequence by hardware. The SCn_PWR to SCn_CLK start (T1) and SCn_CLK_start to SCn_RST assert (T2) can be selected by programming INITSEL (SCn_ALTCTL[9:8]). The SCn_PWR to SCn_CLK length can be configure by setting T1EXT(SCn_ACTCTL[4:0]). This programming procedure provides user with a simple setting for activation sequence. During the hardware activation, RX receive is disabled and cannot receive data.

The activation control sequence in hardware activation mode:

1. Set activation timing by setting INITSEL (SCn_ALTCTL[9:8]).
2. Timer0 can be selected by setting TMRSEL (SCn_CTL[14:13]) is 11.
3. Set operation mode OPMODE (SCn_TMRCTL0[27:24]) to 0011 and give an Answer to Request (ATR) value by setting CNT (SCn_TMRCTL0[23:0]) register.

4. When hardware de-asserts SCn_RST to high, hardware will generate an interrupt INITIF (SCn_INTSTS[8]) to CPU at the same time if INITIEN (SCn_INTEN[8]) is 1.
5. If the Timer0 decreases the counter to “0” (started from SCn_RST de-assert) and the card does not respond ATR before that time, hardware will generate an interrupt flag TMR0IF (SCn_INTSTS[3]).



Time	Comment
T1	SC_PWR to SC_CLK Start
T1EXT	Configurable cycles for T1
T2	SC_CLK Start to SC_RST Assert
T3	SC_CLK Start to ART Appear

INITSEL (SC_ALTCTL[9:8])	T1	T2
00	83.5	491
01	133	537
10	165	569
11	165	42060

Note : The values are measured by chip I / O pin and the real value will depend on system design

Figure 6.25-4 SC Activation Sequence

Warm Reset

The warm reset sequence is shown in Figure 6.25-5.

1. Set SCn_RST to low by programming RSTEN (SCn_PINCTL[1]) to ‘0’, and wait SYNC(SCn_PINCTL[31]) is cleared to 0.
2. Set SCn_DATA to high by programming SCDATA (SCn_PINCTL[9]) to ‘1’, and wait SYNC(SCn_PINCTL[31]) is cleared to 0.
3. Set SCn_RST to high by programming RSTEN (SCn_PINCTL[1]) to ‘1’, and wait SYNC(SCn_PINCTL[31]) is cleared to 0.

The warm reset sequence can be controlled in two ways. The procedure is shown as follows.

- Software Timing Control:

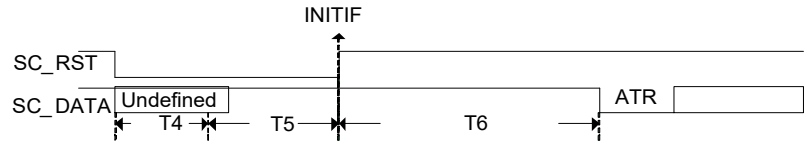
  Set SCn_PINCTL and SCn_TMRCTLx (x = 0, 1, 2) to process the warm reset sequence. The SCn_RST and SCn_DATA pin state can be programmed by SCn_PINCTL. The warm reset sequence timing can be controlled by setting SCn_TMRCTLx (x = 0, 1, 2). This programming procedure provides user with a flexible timing setting for warm reset sequence.

- Hardware Timing Control:

  Set WARSTEN (SCn_ALTCTL[4]) to ‘1’ and the interface will perform the warm reset sequence by hardware. The SCn_RST to SCn_DATA reception mode (T4) and SCn_DATA reception mode to SCn_RST assert (T5) can be selected by programming INITSEL (SCn_ALTCTL[9:8]). This programming procedure provides user with a simple setting for warm reset sequence. During the hardware warm reset, RX receive is disabled and cannot receive data.

The following is the warm reset control sequence by hardware:

1. Set warm reset timing by setting INITSEL (SCn_ALTCTL[9:8]).
2. Select Timer0 by setting TMRSEL (SCn_CTL[14:13]) to 11.
3. Set operation mode OPMODE (SCn_TMRCTL0[27:24]) to 0011 and give an Answer to Request (ATR) value by setting CNT (SCn_TMRCTL0[23:0]) register.
4. Set CNTEN0 (SCn_ALTCTL[5]) and WARSTEN (SCn_ALTCTL[4]) to start counting.
5. When hardware de-asserts SCn_RST to high, hardware will generate an interrupt INITIF (SCn_INTSTS[8]) to CPU at the same time if INITIEN (SCn_INTEN[8]) is 1.
6. If the Timer0 decreases the counter to '0' (start from SCn_RST) and the card does not respond ATR before that time, hardware will generate an interrupt flag TMR0IF (SCn_INTSTS[3]).



Time	Comment
T4	SC_RST to SC_DATA Reception Mode
T5	SC_DATA Reception Mode to SC_RST Assert
T6	SC_CLK Start to ATR Appear

INITSEL (SC_ALTCTL[9:8])	T4	T5
00	81	483
01	129	531
10	161	563
11	161	42106

Unit: SC Clock

Note: This value is measured by chip IO pin and the real value will depend on system design

Figure 6.25-5 SC Warm Reset Sequence

Deactivation

The deactivation sequence is shown in Figure 6.25-6.

1. Set SCn_RST to low by programming RSTEN (SCn_PINCTL[1]) to '0', and wait SYNC(SCn_PINCTL[31]) is cleared to 0.
2. Stop SCn_CLK by programming CLKKEEP (SCn_PINCTL[6]) to '0', and wait SYNC(SCn_PINCTL[31]) is cleared to 0.
3. Set SCn_DATA to low by programming SCDATA (SCn_PINCTL[9]) to '0', and wait SYNC(SCn_PINCTL[31]) is cleared to 0.
4. Deactivate SCn_PWR by programming PWREN (SCn_PINCTL[0]) to '0', and wait SYNC(SCn_PINCTL[31]) is cleared to 0.

The deactivation sequence can be controlled in two ways. The procedure is shown as follows.

- Software Timing Control:

  Set SCn_PINCTL and SCn_TMRCTL0 to process the deactivation sequence. SCn_PWR, SCn_CLK, SCn_RST and SCn_DATA pin state can be programmed by SCn_PINCTL. The deactivation sequence timing can be controlled by setting SCn_TMRCTL0. This programming procedure provides user with a flexible timing setting for deactivation sequence.

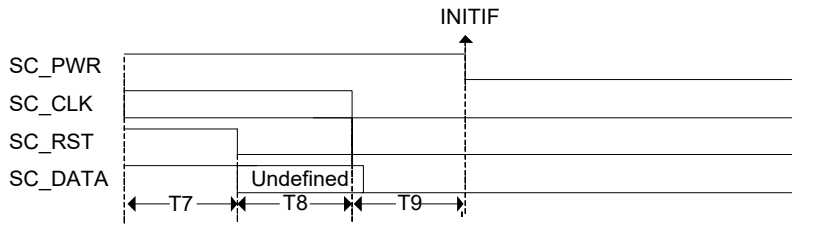
- Hardware Timing Control:

  DACTEN (SCn_ALTCTL[2]) to '1' and the interface will perform the deactivation sequence by hardware. The Deactivation Trigger to SCn_RST low (T7), SCn_RST low to SCn_CLK (T8) and stop SCn_CLK to stop SCn_PWR (T9) time can be selected by programming

INITSEL (SCn_ALTCTL[9:8]). This programming procedure provides user with a simple setting for deactivation sequence.

When hardware de-asserts SCn_PWR to low, the SC controller will generate an interrupt INITIF (SCn_INTSTS[8]) to CPU at the same time if INITIEN (SCn_INTEN[8]) is 1.

The SC controller also supports auto deactivation sequence when the card removal detection is enabled by setting ADACEN (SCn_ALTCTL[11]).



Time	Comment
T7	Deactivation Trigger to SC_RST Low
T8	SMC_RST Low to Stop SC_CLK
T9	Stop SC_CLK to Stop SC_PWR

INITSEL (SC_ALTCTL[9:8])	T7	T8	T9
00	97	83	87
01	145	131	135
10	177	163	167
11	177	163	167

Unit: SC Clock

Figure 6.25-6 SC Deactivation Sequence

6.25.5.3 Basic Operation Flow

Basic operation flow of smartcard can be referenced from ISO 7816-3 and EMV.

The programming sequence flow is shown as follows:

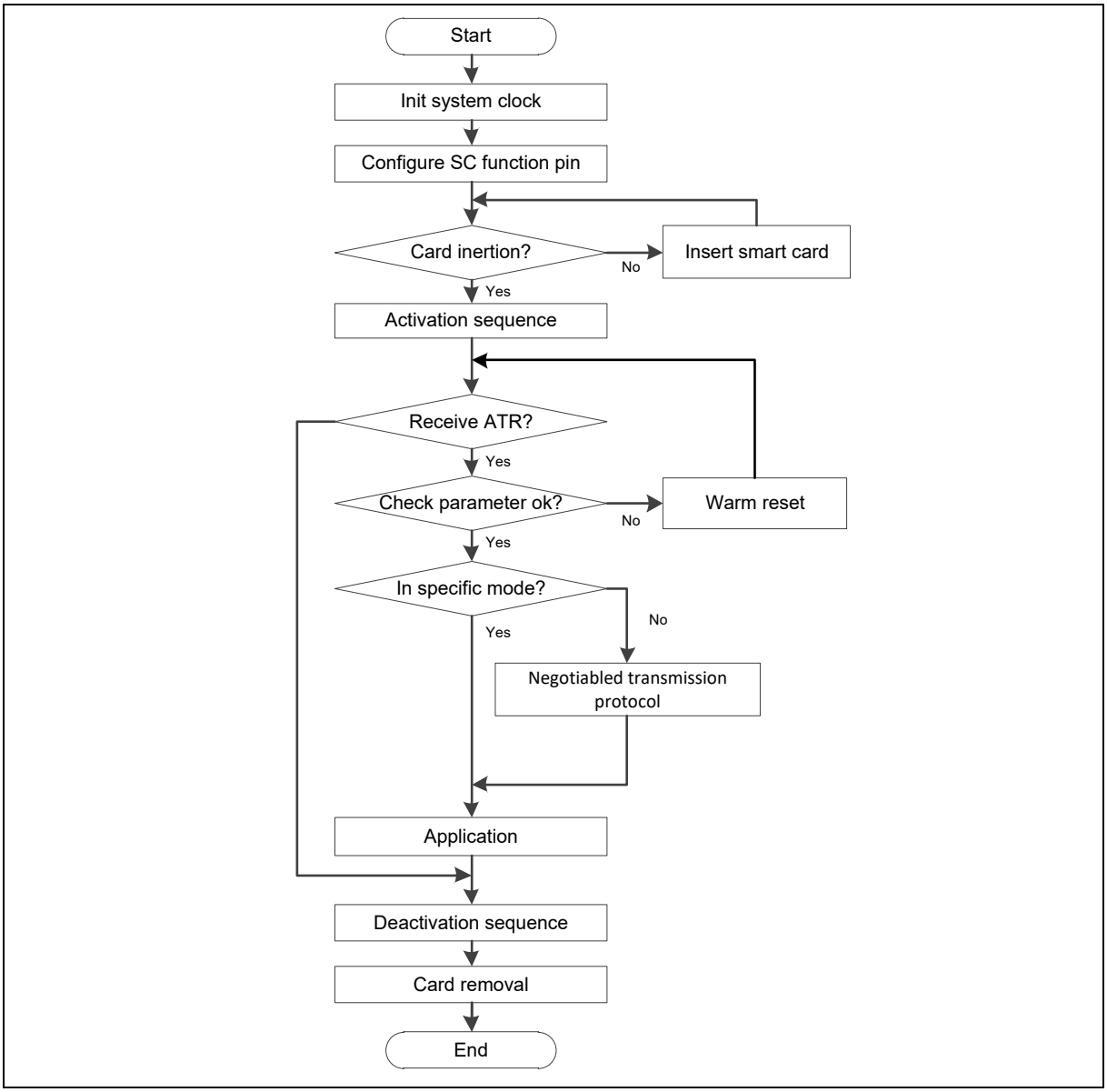
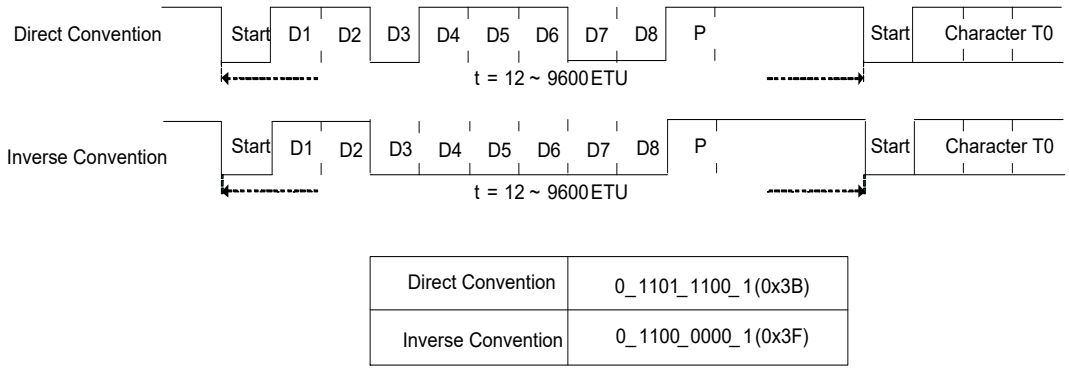


Figure 6.25-7 Basic Operation Flow

6.25.5.4 Initial Character TS

According to ISO 7816-3, the initial character TS has two possible patterns shown in Figure 6.25-8. If the TS pattern is 1100_0000, it is inverse convention. When decoded by inverse convention, the conveyed byte is equal to 0x3F. If the TS pattern is 1101_1100, it is direct convention. When decoded by direct convention, the conveyed byte is equal to 0x3B. User can set AUTOCEN (SCn_CTL[3]) and then the operating convention will be decided by hardware. User can also set the CONSEL (SCn_CTL[5:4]) register (set to '00' or '11') to change the operating convention after SC received TS of answer to request (ATR).

If auto convention function is enabled by setting AUTOCEN (SCn_CTL[3]) register, the setting step must be done before Answer to Request (ATR) state and the received first data must be 0x3B or 0x3F. After hardware received first data and stored it at SCn_DAT, the hardware will decide the convention and change the CONSEL (SCn_CTL[5:4]) register automatically. If the received first data is neither 0x3B nor 0x3F, ACERRIF (SCn_INTSTS[10] Auto Convention Error Interrupt Status Flag) will be set and the hardware will generate an interrupt to CPU if ACERRIEN (SCn_INTEN[10]) is 1.



Direct Convention	0_1101_1100_1(0x3B)
Inverse Convention	0_1100_0000_1(0x3F)

Figure 6.25-8 Initial Character TS

6.25.5.5 Transfer Data Flow and Data Buffer Status

After setting initial sequence, SC can start transferring data which format is corresponding to ISO 7816-3. Set ETURDIV(SC_ETUCTL[11:0]) to 273 to make ETU(Element Timing Unit) meet ISO 7816-3. Writing data to SC_DAT, SC will send out an 8-bit data to smart card. Reading data from SC_DAT, SC will return an 8-bit received data from smart card.

Data buffer status show in SC_STATUS. TXPOINT(SC_STATUS[26:24]) and RXPOINT(SC_STATUS[18:16]) represent how many data in transmit buffer and received buffer. TXEMPTY(SC_STATUS[9]), TXFULL(SC_STATUS[10]), TXOV(SC_STATUS[8]), RXEMPTY(SC_STATUS[1]), RXFULL(SC_STATUS[2]), RXOV(SC_STATUS[0]), represent the transmitted/received buffer status is full, empty, or overflow. SC even can generate interrupt for the transmit buffer empty situation by setting TBEIEN(SCn_INTEN[1]) flag. After interrupt is generated, it can be clear by writing 1 to TBEIF(SCn_INTSTS[1]). SC can decide to transfer data or not by polling these statuses.

TX and RX can be disabled separately by setting TXOFF(SC_CTL[2]) and RXOFF(SC_CTL[1]). TXACT(SC_STATUS[31]) and RXACT(SC_STATUS[23]) represent the TX transfer/RX transfer is active or not.

6.25.5.6 Receiver Buffer Time-out

The time-out down counter resets and starts counting whenever the RX buffer received a new data. Once the counter decrease to 1 and no new data is received or CPU does not read data by reading SCn_DAT, a receiver time-out flag RXTOIF (SCn_INTSTS[9]) will be set, and hardware will generate an interrupt to CPU when RXTOIEN (SCn_INTEN[9]) is enabled.

6.25.5.7 Error Signal and Character Repetition

According to ISO 7816-3 T=0 mode description, as shown in Figure 6.25-9, if the receiver receives a wrong parity bit, it will pull the SCn_DAT to low by 1.5 bit period to inform the transmitter parity error. Then the transmitter will retransmit the character. The SC interface controller supports hardware error detection function in receiver and supports hardware re-transmit function in transmitter.

User can enable re-transmit function by setting TXRTYEN (SCn_CTL[23]). User can also define the retry (re-transmit) number limitation in TXRTY (SCn_CTL[22:20]). If the re-transmit number is between 1 and TXRTY, TXRERR (SCn_STATUS[29]) flag will be set by hardware. The re-transmit number is up to TXRTY +1 and if the re-transmit number is equal to TXRTY +1, TXOVERR (SCn_STATUS[30]) flag will be set by hardware and if TERRIEN (SCn_INTEN[2]) is enabled, SC controller will generate a transfer error interrupt to CPU, and TERRIF (SC_INTSTS[2]) flag will also be set.

User can also enable re-received function by setting RXRTYEN (SCn_CTL[19]).The received retry number limitation is defined in RXRTY (SCn_CTL[18:16]). If the re-received number is between 1 and

RXRTY, RXRERR (SCn_STATUS[21]) flag will be set by hardware. The receiver retry number is up to RXRTY +1, if the number of received errors by receiver is equal to RXRTY +1, receiver will receive this error data to buffer and RXOVERR (SCn_STATUS[22]) flag will be set by hardware and if TERRIEN (SCn_INTEN[2]) is enabled, SC controller will generate a transfer error interrupt to CPU, and TERRIF (SC_INTSTS[2]) flag will also be set.

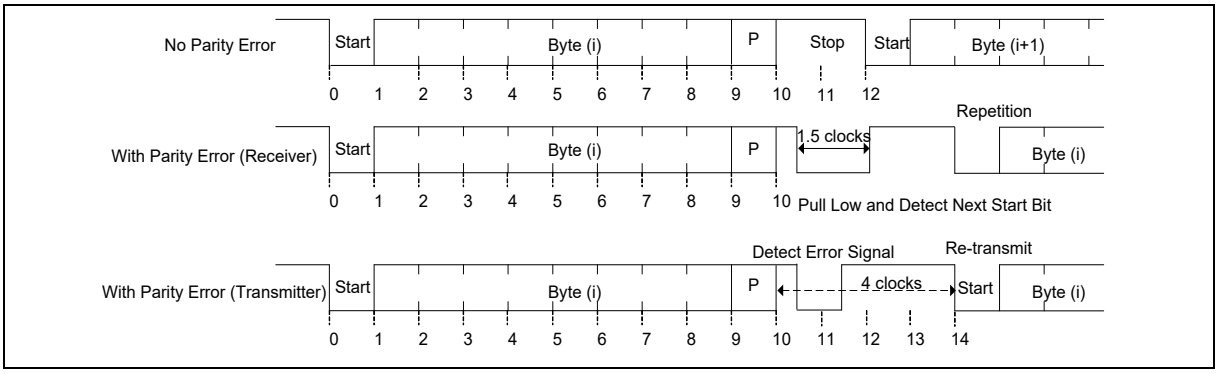


Figure 6.25-9 SC Error Signal

6.25.5.8 Internal Timer Operation Mode

The smart card interface includes a 24-bit time-out counter and two 8-bit time-out counters. These counters help the controller in processing different real-time interval. Each counter can be set to start counting once the trigger enable bit (CNTENx in SCn_ALTCTL[7:5], x = 0, 1, 2) has been written or a START bit has been detected.

The following is the programming flow:

1. Enable counter by setting TMRSEL (SCn_CTL[14:13]) to 11.
2. Select operation mode OPMODE (SCn_TMRCTLx[27:24], x = 0, 1, 2).
3. Give a count value CNT for Timer0, Timer1 and Timer2 by setting CNT0(SCn_TMRCTL0[23:0]), CNT1(SCn_TMRCTL1[7:0]) and CNT2(SCn_TMRCTL2[7:0] register).
4. Set CNTEN0 (SCn_ALTCTL [5]), CNTEN1 (SCn_ALTCTL [6]) or CNTEN2 (SCn_ALTCTL [7]) to enable timer. ACTSTS0(SCn_ALTCTL[13]), ACTSTS1(SCn_ALTCTL[14]) and ACTSTS2(SCn_ALTCTL[15]) represent the status that timer is enable or not.
5. Wait until the counting condition is satisfied, the timer starts counting.
6. When internal timer counter satisfied interrupt conditions in different modes and TMR0IEN(SCn_INTEN[3]), TMR1IEN(SCn_INTEN[4]), TMR2IEN(SCn_INTEN[5]) are enabled, SC will generate an interrupt to CPU.

The SCn_TMRCTL0, SCn_TMRCTL1 and SCn_TMRCTL2 timer operation mode are listed in Table 6.25-3.

Note 1: Only Timer0 (SCn_TMRCTL0 register) supports mode 0011.

Note 2: START bit can only be detected when TX or RX is idle or finish the last transmission.

OPMODE (SCn_TMRCTLx[27:24]), X = 0, 1, 2)	Operation Mode Description

0000	The down counter is started when CNTENx (SCn_ALTCTL[7:5]) enabled and ended when counter time-out. The time-out counter value will be CNT (SCn_TMRCTL0[23:0], SCn_TMRCTL1[7:0], SCn_TMRCTL2[7:0]) +1.	
	Start	Start counting when CNTENx (SCn_ALTCTL[7:5]) enabled.
	End	When the down counter equals 0, hardware will set TMR0IF, TMR1IF, TMR2IF (SCn_INTSTS[5:3]) and clear CNTENx (SCn_ALTCTL[7:5]) automatically.
0001	The down counter is started when the first START bit (reception or transmission) detected and ended when counter time-out. It takes 2 ETU to detect first START bit after writing data to Tx or receiving data from RX. The time-out counter value will be CNT (SCn_TMRCTL0[23:0], SCn_TMRCTL1[7:0], SCn_TMRCTL2[7:0]) +1.	
	Start	Start counting when the first START bit (reception or transmission) detected after CNTENx (SCn_ALTCTL[7:5]) set to 1.
	End	When the down counter equals 0, hardware will set TMR0IF, TMR1IF, TMR2IF (SCn_INTSTS[5:3]) and clear CNTENx (SCn_ALTCTL[7:5]) automatically.
0010	The down counter is started when the first START bit (reception) detected and ended when counter time-out. It takes 2 ETU to detect first START bit after receiving data from RX. The time-out value will be CNT (SCn_TMRCTL0[23:0], SCn_TMRCTL1[7:0], SCn_TMRCTL2[7:0]) +1.	
	Start	Start counting when the first START bit (reception) detected bit after CNTENx (SCn_ALTCTL[7:5]) set to 1.
	End	When the down counter equals 0, hardware will set TMR0IF, TMR1IF, TMR2IF (SCn_INTSTS[5:3]) and clear CNTENx (SCn_ALTCTL[7:5]) automatically.
0011	The down counter is only used for hardware activation, warm reset sequence to measure ATR timing. The timing starts when SCn_RST de-assertion and ends when ATR response received or time-out. If the counter decreases to 0 before ATR response received, hardware will set TMR0IF (SCn_INTSTS[3]) and generate an interrupt to CPU if TMR0IEN (SCn_INTEN[3]) enabled. The time-out value will be CNT (SCn_TMRCTL0[23:0]) +1.	
	Start	Start counting when SCn_RST de-assertion after CNTEN0 (SCn_ALTCTL[5]) set to 1. It is only used for hardware activation, warm reset mode.
	End	When the down counter equals 0 before ATR response received, hardware will set TMR0IF and clear CNTEN0 (SCn_ALTCTL[5]) automatically. When ATR received and down counter does not equal to 0, hardware will clear CNTEN0 (SCn_ALTCTL[5]) automatically.
0100	Start	Start down counter counting when CNTENx (SCn_ALTCTL[7:5]) enabled.
	Recount and reload	When ACTSTSx (SCn_ALTCTL[15:13]) is 1, user can change CNT (SCn_TMRCTL0[23:0], SCn_TMRCTL0[7:0], SCn_TMRCTL0[7:0]) value at any time. It will reload the last value which is filled into the CNT(SCn_TMRCTL0[23:0], SCn_TMRCTL1[7:0], SCn_TMRCTL2[7:0]) before the counter counts to 0. Only when the down counter equals 0, counter reload the CNT (SCn_TMRCTL0[23:0], SCn_TMRCTL1[7:0], SCn_TMRCTL2[7:0]) value and start to recount.
	Interrupt	If the counter decreases to 0, hardware will set TMR0IF, TMR1IF, TMR2IF (SCn_INTSTS[5:3]) and generate an interrupt to CPU if TMRxIEN (SCn_INTEN[5:3]) enabled. The time-out value will be CNT (SCn_TMRCTL0[23:0], SCn_TMRCTL1[7:0], SCn_TMRCTL2[7:0]) +1.
	End	The down counter stopped when use clears CNTENx (SCn_ALTCTL[7:5]) bit.
0101	Start	The down counter is started when the first START bit (reception or transmission) detected after CNTENx (SCn_ALTCTL[7:5]) set to 1. It takes 2 ETU to detect START bit after writing data to TX or receiving data from RX.

	Reload	When ACTSTSx (SCn_ALTCTL[15:13]) is 1, user can change CNT (SCn_TMRCTL0[23:0], SCn_TMRCTL0[7:0], SCn_TMRCTL0[7:0]) value at any time. It will reload the last value which is filled into the CNT(SCn_TMRCTL0[23:0], SCn_TMRCTL1[7:0], SCn_TMRCTL2[7:0]) before the counter counts to 0. Only when the down counter equals 0, counter will reload the CNT (SCn_TMRCTL0[23:0], SCn_TMRCTL1[7:0], SCn_TMRCTL2[7:0]) value.
	Recount	After down counter reloads the CNT value, timer counter starts to recount only when the next START bit is detected.
	Interrupt	If the counter decreases to 0, hardware will set TMR0IF, TMR1IF, TMR2IF (SCn_INTSTS[5:3]) and generate an interrupt to CPU if TMRxIEN (SCn_INTEN[5:3]) enabled. The time-out value will be CNT (SCn_TMRCTL0[23:0], SCn_TMRCTL1[7:0], SCn_TMRCTL2[7:0]) +1.
	End	The down counter stopped when user clears CNTENx (SCn_ALTCTL[7:5]) bit.
0110	Start	The down counter is started when the first START bit (reception) detected after CNTENx (SCn_ALTCTL[7:5]) set to 1. It takes 2 ETU to detect START bit after writing data to Tx or receiving data from RX.
	Reload	When ACTSTSx (SCn_ALTCTL[15:13]) is 1, user can change CNT (SCn_TMRCTL0[23:0], SCn_TMRCTL0[7:0], SCn_TMRCTL0[7:0]) value at any time. It will reload the last value which is filled into the CNT(SCn_TMRCTL0[23:0], SCn_TMRCTL1[7:0], SCn_TMRCTL2[7:0]) before the counter counts to 0. Only when the down counter equals 0, counter reload the CNT (SCn_TMRCTL0[23:0], SCn_TMRCTL1[7:0], SCn_TMRCTL2[7:0]) value.
	Recount	After the down counter reloads the CNT value, timer counter starts to recount only when the next START bit is detected.
	Interrupt	If the counter decreases to 0, hardware will set TMR0IF, TMR1IF, TMR2IF (SCn_INTSTS[5:3]) and generate an interrupt to CPU if TMRxIEN (SCn_INTEN[5:3]) enabled. The time-out value will be CNT (SCn_TMRCTL0[23:0], SCn_TMRCTL1[7:0], SCn_TMRCTL2[7:0])+1.
	End	The down counter stopped when user clears CNTENx (SCn_ALTCTL[7:5]) bit.
0111	Start	The down counter is started when the first START bit (reception or transmission) detected after CNTENx (SCn_ALTCTL[7:5]) set to 1. It takes 2 ETU to detect START bit after writing data to TX or receiving data from RX.
	Reload &recount	Only when the next START bit is detected, counter will reload the new value of CNT (SCn_TMRCTL0[23:0], SCn_TMRCTL1[7:0], SCn_TMRCTL2[7:0]) and recount.
	Interrupt	If the counter decreases to 0 before the next START bit detected, hardware will set TMR0IF, TMR1IF, TMR2IF (SCn_INTSTS[5:3]) and generate an interrupt to CPU if TMRxIEN (SCn_INTEN[5:3]) enabled. The time-out value will be CNT (SCn_TMRCTL0[23:0], SCn_TMRCTL1[7:0], SCn_TMRCTL2[7:0]) +1.
	End	The down counter stopped when user clears CNTENx (SCn_ALTCTL[7:5]) bit.
1111	Start	The down counter starts counting when user sets CNTENx (SCn_ALTCTL[7:5]) bit and it will count to time-out.
	Reload &recount	Only when the next START bit is detected, counter will reload the new value of CNT (SCn_TMRCTL0[23:0], SCn_TMRCTL1[7:0], SCn_TMRCTL2[7:0]) and recount.
	Interrupt	If the counter decreases to 0 before the next START bit detected, hardware will generate time-out interrupt flag TMR0IF, TMR1IF, TMR2IF (SCn_INTSTS[5:3]). The time-out value will be CNTx (SCn_TMRCTL0[23:0], SCn_TMRCTL1[7:0], SCn_TMRCTL2[7:0]) +1.
	End	The down counter stopped when user clears CNTENx (SCn_ALTCTL[7:5]) bit.

Table 6.25-3 Timer0/Timer1/Timer2 Operation Mode

6.25.5.9 Block Guard Time and Extra Guard Time

Block guard time means the minimum interval between the leading edges of two consecutive characters between different transfer directions. This field indicates the counter for the bit length of block guard time. According to ISO7816-3, in T = 0 mode, user must fill 15 (real block guard time = 16.5) to this field; in T = 1 mode, user must fill 21 (real block guard time = 22.5) to it.

In transmit direction, the smart card sends data to smart card host controller, first. After the period is greater than BGT (SCn_CTL[12:8]), the smart card host controller begin to send the data.

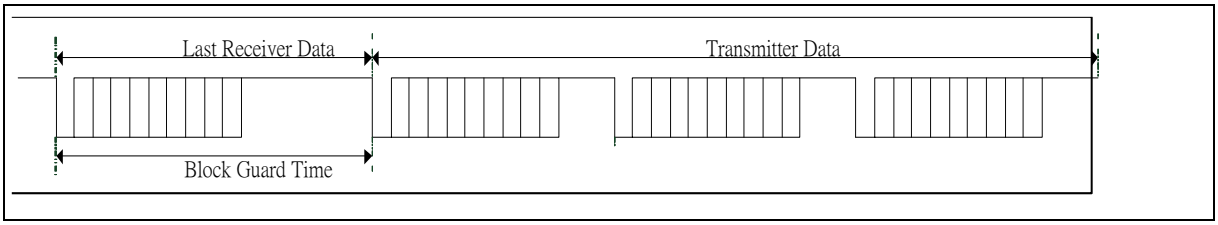


Figure 6.25-10 Transmit Direction Block Guard Time Operation

In receive direction, the smart card host controller sends data to smart card, first. If the smart card responds data to smart card host controller at the time which is less than BGT (SCn_CTL[12:8]), the block guard time interrupt BGTIF (SC_INTSTS[6]) is generated when RXBGTEN (SCn_ALTCTL[12]) and BGTIEN(SCn_INTEN[6]) is enabled.

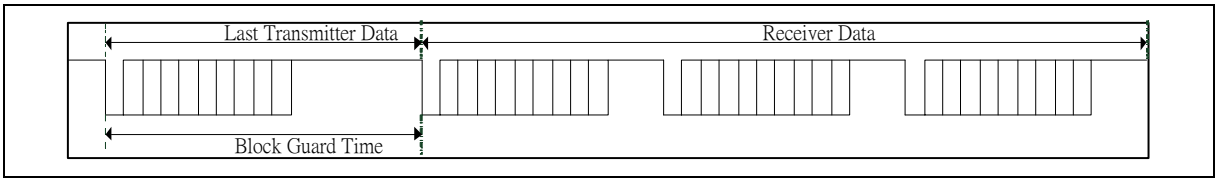


Figure 6.25-11 Receive Direction Block Guard Time Operation

Extra Guard Time is EGT (SCn_EGT[7:0]), and it only affects the data transmitted by smart card interface. The format is shown as Figure 6.25-12.

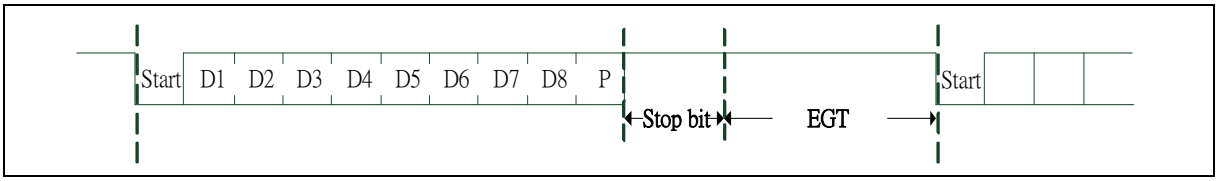


Figure 6.25-12 Extra Guard Time Operation

6.25.5.10 UART Mode

When the UARTEN (SCn_UARTCTL[0]) bit is set, the Smart Card Interface controller can also be used as basic UART function. The following is the programming example for UART mode.

1. Set UARTEN (SCn_UARTCTL[0]) bit to enter UART mode.
2. Do user reset by setting RXRST (SCn_ALTCTL[1]) and TXRST(SCn_ALTCTL[0]) bit to ensure that all state machine return idle state.
3. Fill “0” to CONSEL (SCn_CTL[5:4]) and AUTOCEN (SCn_CTL[3]) field. In UART mode, those

fields must be “0”.

4. Select the UART baud rate by setting ETURDIV (SCn_ETUCTL[11:0]) fields. For example, if smartcard module clock is 24 MHz and target baud rate is 115200 bps, ETURDIV should fill with ((24000000 / 115200) - 1).
5. Select the data format include data length (by setting WLS (SCn_UARTCTL[5:4]), parity format (by setting OPE (SCn_UARTCTL[7]) and PBOFF (SCn_UARTCTL[6])) and stop bit length (by setting NSB (SCn_CTL[15] or EGT (SCn_EGT[7:0])).
6. Select the receiver buffer number trigger level by setting RXTRGLV (SCn_CTL[7:6]) field and select the receiver buffer time-out interval by setting RFTM (SCn_RXTOUT[8:0]) field.
7. Write the SCn_DAT (SCn_DAT[7:0]) (TX) register or read the SCn_DAT (SCn_DAT[7:0]) (RX) register can perform UART function.

6.25.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
SC Base Address: **SCn_BA = 0x4090_0000 + (0x1_0000 * n)** **n=0,1**				
SC_DAT	SCn_BA+0x00	R/W	SC Receive/Transmit Holding Buffer Register	0xXXXX_XXXX
SC_CTL	SCn_BA+0x04	R/W	SC Control Register	0x0000_0000
SC_ALTCTL	SCn_BA+0x08	R/W	SC Alternate Control Register	0x0000_0000
SC_EGT	SCn_BA+0x0C	R/W	SC Extra Guard Time Register	0x0000_0000
SC_RXTOUT	SCn_BA+0x10	R/W	SC Receive Buffer Time-out Counter Register	0x0000_0000
SC_ETUCTL	SCn_BA+0x14	R/W	SC Element Time Unit Control Register	0x0000_0173
SC_INTEN	SCn_BA+0x18	R/W	SC Interrupt Enable Control Register	0x0000_0000
SC_INTSTS	SCn_BA+0x1C	R/W	SC Interrupt Status Register	0x0000_0002
SC_STATUS	SCn_BA+0x20	R/W	SC Transfer Status Register	0x0000_X202
SC_PINCTL	SCn_BA+0x24	R/W	SC Pin Control State Register	0x0000_0000
SC_TMRCTL0	SCn_BA+0x28	R/W	SC Internal Timer0 Control Register	0x0000_0000
SC_TMRCTL1	SCn_BA+0x2C	R/W	SC Internal Timer1 Control Register	0x0000_0000
SC_TMRCTL2	SCn_BA+0x30	R/W	SC Internal Timer2 Control Register	0x0000_0000
SC_UARTCTL	SCn_BA+0x34	R/W	SC UART Mode Control Register	0x0000_0000
SC_ACTCTL	SCn_BA+0x4C	R/W	SC Activation Control Register	0x0000_0000

6.25.7 Register Description

SC Receive/Transmit Holding Buffer Register (SC_DAT)

Register	Offset	R/W	Description	Reset Value
SC_DAT	SCn_BA+0x00	R/W	SC Receive/Transmit Holding Buffer Register	0xXXXX_XXXX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DAT							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**DAT**	**Receive/Transmit Holding Buffer** Write Operation: By writing data to DAT, the SC will send out an 8-bit data. **Note:** If SCEN (SCn_CTL[0]) is not enabled, DAT cannot be programmed. Read Operation: By reading DAT, the SC will return an 8-bit received data.

SC Control Register (SC_CTL)

Register	Offset	R/W	Description	Reset Value
SC_CTL	SCn_BA+0x04	R/W	SC Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	SYNC	Reserved			CDLV	CDDBSEL	
23	22	21	20	19	18	17	16
TXRTYEN	TXRTY			RXRTYEN	RXRTY		
15	14	13	12	11	10	9	8
NSB	TMRSEL		BGT				
7	6	5	4	3	2	1	0
RXTRGLV		CONSEL		AUTOCEN	TXOFF	RXOFF	SCEN

Bits	Description	
[31]	**Reserved**	Reserved.
[30]	**SYNC**	**SYNC Flag Indicator (Read Only)** Due to synchronization, user should check this bit before writing a new value to RXRTY and TXRTY fields. 0 = Synchronizing is completed, and user can write new data to RXRTY and TXRTY. 1 = Last value is synchronizing.
[29:27]	**Reserved**	Reserved.
[26]	**CDLV**	**Card Detect Level Selection** 0 = When hardware detects the card detect pin (SCn_CD) from high to low, it indicates a card is detected. 1 = When hardware detects the card detect pin (SCn_CD) from low to high, it indicates a card is detected. **Note:** User must select card detect level before Smart Card controller enabled.
[25:24]	**CDDBSEL**	**Card Detect De-bounce Selection** This field indicates the card detect de-bounce selection. 00 = De-bounce sample card insert once per 384 (128 * 3) SC module clocks and de-bounce sample card removal once per 128 SC module clocks. Other configurations are reserved.
[23]	**TXRTYEN**	**TX Error Retry Enable Bit** This bit enables transmitter retry function when parity error has occurred. 0 = TX error retry function Disabled. 1 = TX error retry function Enabled.
[22:20]	**TXRTY**	**TX Error Retry Count Number** This field indicates the maximum number of transmitter retries that are allowed when parity error has occurred. **Note 1:** The real retry number is TXRTY + 1, so 8 is the maximum retry number. **Note 2:** This field cannot be changed when TXRTYEN enabled. The change flow is to disable TXRTYEN first and then fill in new retry value.
[19]	**RXRTYEN**	**RX Error Retry Enable Bit** This bit enables receiver retry function when parity error has occurred.

		0 = RX error retry function Disabled. 1 = RX error retry function Enabled. **Note:** User must fill in the RXRTY value before enabling this bit.
[18:16]	**RXRTY**	**RX Error Retry Count Number** This field indicates the maximum number of receiver retries that are allowed when parity error has occurred **Note 1:** The real retry number is RXRTY + 1, so 8 is the maximum retry number. **Note 2:** This field cannot be changed when RXRTYEN enabled. The change flow is to disable RXRTYEN first and then fill in new retry value.
[15]	**NSB**	**Stop Bit Length** This field indicates the length of stop bit. 0 = The stop bit length is 2 ETU. 1= The stop bit length is 1 ETU. **Note 1:** The default stop bit length is 2. SC and UART adopts NSB to program the stop bit length. **Note 2:** In UART mode, RX can receive the data sequence in 1 stop bit or 2 stop bits with NSB is set to 0.
[14:13]	**TMRSEL**	**Timer Channel Selection** 00 = All internal timer function Disabled. 11 = Internal 24 bit timer and two 8 bit timers Enabled. User can configure them by setting SCn_TMRCTL0[23:0], SCn_TMRCTL1[7:0] and SCn_TMRCTL2[7:0]. Other configurations are reserved
[12:8]	**BGT**	**Block Guard Time** Block guard time means the minimum interval between the leading edges of two consecutive characters between different transfer directions. This field indicates the counter for the bit length of block guard time. According to ISO 7816-3, in T = 0 mode, user must fill 15 (real block guard time = 16.5) to this field; in T = 1 mode, user must fill 21 (real block guard time = 22.5) to it. **Note:** The real block guard time is BGT + 1.
[7:6]	**RXTRGLV**	**RX Buffer Trigger Level** When the number of bytes in the receiving buffer equals the RXTRGLV, the RDAIF will be set. If RDAIEN (SCn_INTEN[0]) is enabled, an interrupt will be generated to CPU. 00 = RX Buffer Trigger Level with 01 bytes. 01 = RX Buffer Trigger Level with 02 bytes. 10 = RX Buffer Trigger Level with 03 bytes. 11 = Reserved.
[5:4]	**CONSEL**	**Convention Selection** 00 = Direct convention. 01 = Reserved. 10 = Reserved. 11 = Inverse convention. **Note:** If AUTOCEN (SCn_CTL[3]) is enabled, this field is ignored.
[3]	**AUTOCEN**	**Auto Convention Enable Bit** This bit is used for enable auto convention function. 0 = Auto-convention Disabled. 1 = Auto-convention Enabled. If user enables auto convention function, the setting step must be done before Answer to Reset (ATR) state and the first data must be 0x3B or 0x3F. After hardware received first data and stored it at buffer, hardware will decided the convention and change the CONSEL (SCn_CTL[5:4]) bits automatically when received first data is 0x3B or 0x3F. If received first byte is 0x3B, TS is direct convention, CONSEL (SCn_CTL[5:4]) will be set to 00 automatically, otherwise the TS is inverse convention, and CONSEL (SCn_CTL[5:4]) will be set to 11. If the first data is not 0x3B or 0x3F, hardware will set ACERRIF (SCn_INTSTS[10]) and generate an

		interrupt to CPU when ACERRIEN (SCn_INTEN[10]) is enabled.
[2]	**TXOFF**	**TX Transition Disable Control Bit** This bit is used for disable TX transition function. 0 = The transceiver Enabled. 1 = The transceiver Disabled.
[1]	**RXOFF**	**RX Transition Disable Control Bit** This bit is used for disable RX transition function. 0 = The receiver Enabled. 1 = The receiver Disabled. **Note:** If AUTOCEN (SCn_CTL[3]) is enabled, this field is ignored.
[0]	**SCEN**	**SC Controller Enable Bit** Set this bit to 1 to enable SC operation. If this bit is cleared, 0 = SC will force all transition to IDLE state. 1 = SC controller is enabled and all function can work correctly. **Note:** SCEN must be set to 1 before filling in other SC registers, or smart card will not work properly.

SC Alternate Control Register (SC_ALTCTL)

Register	Offset	R/W	Description	Reset Value
SC_ALTCTL	SCn_BA+0x08	R/W	SC Alternate Control Register	0x0000_0000

31	30	29	28	27	26	25	24
SYNC	Reserved						
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
ACTSTS2	ACTSTS1	ACTSTS0	RXBGTEN	ADACEN	Reserved	INITSEL	
7	6	5	4	3	2	1	0
CNTEN2	CNTEN1	CNTEN0	WARSTEN	ACTEN	DACTEN	RXRST	TXRST

Bits	Description	
[31]	**SYNC**	**SYNC Flag Indicator (Read Only)** Due to synchronization, user should check this bit when writing a new value to SCn_ALTCTL register. 0 = Synchronizing is completed; user can write new data to SCn_ALTCTL register. 1 = Last value is synchronizing.
[30:16]	**Reserved**	Reserved.
[15]	**ACTSTS2**	**Internal Timer2 Active Status (Read Only)** This bit indicates the timer counter status of timer2. 0 = Timer2 is not active. 1 = Timer2 is active. **Note:** Timer2 is active does not always mean timer2 is counting the CNT (SCn_TMRCTL2[7:0]).
[14]	**ACTSTS1**	**Internal Timer1 Active Status (Read Only)** This bit indicates the timer counter status of timer1. 0 = Timer1 is not active. 1 = Timer1 is active. **Note:** Timer1 is active does not always mean timer1 is counting the CNT (SCn_TMRCTL1[7:0]).
[13]	**ACTSTS0**	**Internal Timer0 Active Status (Read Only)** This bit indicates the timer counter status of timer0. 0 = Timer0 is not active. 1 = Timer0 is active. **Note:** Timer0 is active does not always mean timer0 is counting the CNT (SCn_TMRCTL0[23:0]).
[12]	**RXBGTEN**	**Receiver Block Guard Time Function Enable Bit** This bit enables the receiver block guard time function. 0 = Receiver block guard time function Disabled. 1 = Receiver block guard time function Enabled.
[11]	**ADACEN**	**Auto Deactivation When Card Removal** This bit is used for enable hardware auto deactivation when smart card is removed.

		0 = Auto deactivation Disabled. 1 = Auto deactivation Enabled. **Note:** When the card is removed, hardware will stop any process and then do deactivation sequence if this bit is set. If auto deactivation process completes, hardware will set INITIF (SCn_INTSTS[8]) also.
[10]	**Reserved**	Reserved.
[9:8]	**INITSEL**	**Initial Timing Selection** This field indicates the initial timing of hardware activation, warm-reset or deactivation. The unit of initial timing is SC module clock. Activation: refer to SC Activation Sequence in Figure 6.25-4. Warm-reset: refer to Warm-Reset Sequence in Figure 6.25-5. Deactivation: refer to Deactivation Sequence in Figure 6.25-6. **Note:** When setting activation and warm reset in Timer0 operation mode 0011, it may have deviation at most 128 SC module clock cycles.
[7]	**CNTEN2**	**Internal Timer2 Start Enable Bit** This bit enables Timer 2 to start counting. User can fill 0 to stop it and set 1 to reload and count. The counter unit is ETU base. 0 = Stop counting. 1 = Start counting. **Note 1:** This field is used for internal 8 bit timer when TMRSEL (SCn_CTL[14:13]) is 11 only. Do not fill in CNTEN2 when TMRSEL (SCn_CTL[14:13]) is not equal to 11. **Note 2:** If the operation mode is not in auto-reload mode (SCn_TMRCTL2[26] = 0), this bit will be auto-cleared by hardware. **Note 3:** If SCEN (SCn_CTL[0]) is not enabled, this filed cannot be programmed.
[6]	**CNTEN1**	**Internal Timer1 Start Enable Bit** This bit enables Timer 1 to start counting. User can fill 0 to stop it and set 1 to reload and count. The counter unit is ETU base. 0 = Stop counting. 1 = Start counting. **Note 1:** This field is used for internal 8 bit timer when TMRSEL(SCn_CTL[14:13]) is 11 only. Do not fill CNTEN1 when TMRSEL (SCn_CTL[14:13]) is not equal to 11. **Note 2:** If the operation mode is not in auto-reload mode (SCn_TMRCTL1[26] = 0), this bit will be auto-cleared by hardware. **Note 3:** If SCEN (SCn_CTL[0]) is not enabled, this filed cannot be programmed.
[5]	**CNTEN0**	**Internal Timer0 Start Enable Bit** This bit enables Timer 0 to start counting. User can fill 0 to stop it and set 1 to reload and count. The counter unit is ETU base. 0 = Stop counting. 1 = Start counting. **Note 1:** This field is used for internal 24 bit timer when TMRSEL (SCn_CTL[14:13]) is 11 only. **Note 2:** If the operation mode is not in auto-reload mode (SCn_TMRCTL0[26] = 0), this bit will be auto-cleared by hardware. **Note 3:** If SCEN (SCn_CTL[0]) is not enabled, this filed cannot be programmed.
[4]	**WARSTEN**	**Warm Reset Sequence Generator Enable Bit** This bit enables SC controller to initiate the card by warm reset sequence. 0 = No effect. 1 = Warm reset sequence generator Enabled. **Note 1:** When the warm reset sequence completed, this bit will be cleared automatically and the INITIF (SCn_INTSTS[8]) will be set to 1. **Note 2:** This field will be cleared by TXRST (SCn_ALTCTL[0]) and RXRST (SCn_ALTCTL[1]). Thus, do not fill in this bit WARSTEN, TXRST and RXRST at the same time.

		Note 3: If SCEN (SCn_CTL[0]) is not enabled, this filed cannot be programmed. **Note 4**: During the warm reset sequence, RX is disabled automatically and cannot receive data. After the warm reset sequence completion, RXOFF (SCn_CTL[1]) keeps the state before perform warm reset sequence.
[3]	**ACTEN**	**Activation Sequence Generator Enable Bit** This bit enables SC controller to initiate the card by activation sequence. 0 = No effect. 1 = Activation sequence generator Enabled. **Note 1:** When the activation sequence completed, this bit will be cleared automatically and the INITIF (SCn_INTSTS[8]) will be set to 1. **Note 2:** This field will be cleared by TXRST (SCn_ALTCTL[0]) and RXRST (SCn_ALTCTL[1]). Thus, do not fill in this bit ACTEN, TXRST and RXRST at the same time. **Note 3:** If SCEN (SCn_CTL[0]) is not enabled, this filed cannot be programmed. **Note 4**: During the activation sequence, RX is disabled automatically and cannot receive data. After the activation sequence completion, RXOFF (SCn_CTL[1]) keeps the state before hardware activation.
[2]	**DACTEN**	**Deactivation Sequence Generator Enable Bit** This bit enables SC controller to initiate the card by deactivation sequence. 0 = No effect. 1 = Deactivation sequence generator Enabled. **Note 1:** When the deactivation sequence completed, this bit will be cleared automatically and the INITIF (SCn_INTSTS[8]) will be set to 1. **Note 2:** This field will be cleared by TXRST (SCn_ALTCTL[0]) and RXRST (SCn_ALTCTL[1]). Thus, do not fill in this bit DACTEN, TXRST and RXRST at the same time. **Note 3:** If SCEN (SCn_CTL[0]) is not enabled, this filed cannot be programmed.
[1]	**RXRST**	**RX Software Reset** When RXRST is set, all the bytes in the receive buffer and RX internal state machine will be cleared. 0 = No effect. 1 = Reset the RX internal state machine and pointers. **Note:** This bit will be auto cleared after reset is complete.
[0]	**TXRST**	**TX Software Reset** When TXRST is set, all the bytes in the transmit buffer and TX internal state machine will be cleared. 0 = No effect. 1 = Reset the TX internal state machine and pointers. **Note:** This bit will be auto cleared after reset is complete.

SC Extra Guard Time Register (SC_EGT)

Register	Offset	R/W	Description	Reset Value
SC_EGT	SCn_BA+0x0C	R/W	SC Extra Guard Time Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EGT							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**EGT**	**Extra Guard Time** This field indicates the extra guard time value. **Note:** The extra guard time unit is ETU base.

SC Receiver Buffer Time-out Register (SC_RXTOUT)

Register	Offset	R/W	Description	Reset Value
SC_RXTOUT	SCn_BA+0x10	R/W	SC Receive Buffer Time-out Counter Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							RFTM
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RFTM							

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8:0]	**RFTM**	**SC Receiver FIFO Time-out Counter** The time-out down counter resets and starts counting whenever the RX buffer received a new data. Once the counter decrease to 1 and no new data is received or CPU does not read data by reading SCn_DAT, a receiver time-out flag RXTOIF (SCn_INTSTS[9]) will be set, and hardware will generate an interrupt to CPU when RXTOIEN (SCn_INTEN[9]) is enabled. **Note 1:** The counter unit is ETU based and the interval of time-out is RFTM + 0.5. **Note 2:** Filling in all 0 to this field indicates to disable this function.

SC Element Time Unit Control Register (SC_ETUCTL)

Register	Offset	R/W	Description	Reset Value
SC_ETUCTL	SCn_BA+0x14	R/W	SC Element Time Unit Control Register	0x0000_0173

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				ETURDIV			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ETURDIV							

Bits	Description	
[31:12]	**Reserved**	Reserved.
[11:0]	**ETURDIV**	**ETU Rate Divider** The field is used for ETU clock rate divider. The real ETU is ETURDIV + 1. **Note:** User can configure this field, but this field must be greater than 0x04.

SC Interrupt Enable Control Register (SC_INTEN)

Register	Offset	R/W	Description	Reset Value
SC_INTEN	SCn_BA+0x18	R/W	SC Interrupt Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					ACERRIEN	RXTOIEN	INITIEN
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CDIEN	BGTIEN	TMR2IEN	TMR1IEN	TMR0IEN	TERRIEN	TBEIEN	RDAIEN

Bits	Description	
[31:11]	**Reserved**	Reserved.
[10]	**ACERRIEN**	**Auto Convention Error Interrupt Enable Bit** This field is used to enable auto-convention error interrupt. 0 = Auto-convention error interrupt Disabled. 1 = Auto-convention error interrupt Enabled.
[9]	**RXTOIEN**	**Receiver Buffer Time-out Interrupt Enable Bit** This field is used to enable receiver buffer time-out interrupt. 0 = Receiver buffer time-out interrupt Disabled. 1 = Receiver buffer time-out interrupt Enabled.
[8]	**INITIEN**	**Initial End Interrupt Enable Bit** This field is used to enable activation (ACTEN (SCn_ALTCTL[3] = 1)), deactivation (DACTEN (SCn_ALTCTL[2] = 1)) and warm reset (WARSTEN (SCn_ALTCTL [4])) sequence complete interrupt. 0 = Initial end interrupt Disabled. 1 = Initial end interrupt Enabled.
[7]	**CDIEN**	**Card Detect Interrupt Enable Bit** This field is used to enable card detect interrupt. The card detect status is CDPINSTS (SCn_STATUS[13]). 0 = Card detect interrupt Disabled. 1 = Card detect interrupt Enabled.
[6]	**BGTIEN**	**Block Guard Time Interrupt Enable Bit** This field is used to enable block guard time interrupt in receive direction. 0 = Block guard time interrupt Disabled. 1 = Block guard time interrupt Enabled. **Note:** This bit is valid only for receive direction block guard time.
[5]	**TMR2IEN**	**Timer2 Interrupt Enable Bit** This field is used to enable Timer2 interrupt function. 0 = Timer2 interrupt Disabled. 1 = Timer2 interrupt Enabled.

[4]	**TMR1IEN**	**Timer1 Interrupt Enable Bit** This field is used to enable the Timer1 interrupt function. 0 = Timer1 interrupt Disabled. 1 = Timer1 interrupt Enabled.
[3]	**TMR0IEN**	**Timer0 Interrupt Enable Bit** This field is used to enable Timer0 interrupt function. 0 = Timer0 interrupt Disabled. 1 = Timer0 interrupt Enabled.
[2]	**TERRIEN**	**Transfer Error Interrupt Enable Bit** This field is used to enable transfer error interrupt. The transfer error states is at SCn_STATUS register which includes receiver break error BEF (SCn_STATUS[6]), frame error FEF (SCn_STATUS[5]), parity error PEF (SCn_STATUS[4]), receive buffer overflow error RXOV (SCn_STATUS[0]), transmit buffer overflow error TXOV (SCn_STATUS[8]), receiver retry over limit error RXOVERR (SCn_STATUS[22]) and transmitter retry over limit error TXOVERR (SCn_STATUS[30]). 0 = Transfer error interrupt Disabled. 1 = Transfer error interrupt Enabled.
[1]	**TBEIEN**	**Transmit Buffer Empty Interrupt Enable Bit** This field is used to enable transmit buffer empty interrupt. 0 = Transmit buffer empty interrupt Disabled. 1 = Transmit buffer empty interrupt Enabled.
[0]	**RDAIEN**	**Receive Data Reach Interrupt Enable Bit** This field is used to enable received data reaching trigger level RXTRGLV (SCn_CTL[7:6]) interrupt. 0 = Receive data reach trigger level interrupt Disabled. 1 = Receive data reach trigger level interrupt Enabled.

SC Interrupt Status Register (SC_INTSTS)

Register	Offset	R/W	Description	Reset Value
SC_INTSTS	SCn_BA+0x1C	R/W	SC Interrupt Status Register	0x0000_0002

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved					ACERRIF	RXTOIF	INITIF
7	6	5	4	3	2	1	0
CDIF	BGTIF	TMR2IF	TMR1IF	TMR0IF	TERRIF	TBEIF	RDAIF

Bits	Description	
[31:11]	**Reserved**	Reserved.
[10]	**ACERRIF**	**Auto Convention Error Interrupt Status Flag** This field indicates auto convention sequence error. 0 = Received TS at ATR state is 0x3B or 0x3F. 1 = Received TS at ATR state is neither 0x3B nor 0x3F. **Note:** This bit can be cleared by writing 1 to it.
[9]	**RXTOIF**	**Receive Buffer Time-out Interrupt Status Flag (Read Only)** This field is used for indicate receive buffer time-out interrupt status flag. 0 = Receive buffer time-out interrupt did not occur. 1 = Receive buffer time-out interrupt occurred. **Note:** This bit is read only; user must read all receive buffer remaining data by reading SCn_DAT register to clear it.
[8]	**INITIF**	**Initial End Interrupt Status Flag** This field is used for activation (ACTEN (SCn_ALTCTL[3])), deactivation (DACTEN (SCn_ALTCTL[2])) and warm reset (WARSTEN (SCn_ALTCTL[4])) sequence interrupt status flag. 0 = Initial sequence is not complete. 1 = Initial sequence is completed. **Note:** This bit can be cleared by writing 1 to it.
[7]	**CDIF**	**Card Detect Interrupt Status Flag (Read Only)** This field is used for card detect interrupt status flag. The card detect status is CINSERT (SCn_STATUS[12]) and CREMOVE (SCn_STATUS[11]). 0 = Card detect event did not occur. 1 = Card detect event occurred. **Note:** This bit is read only; user must to clear CINSERT or CREMOVE status to clear it.
[6]	**BGTIF**	**Block Guard Time Interrupt Status Flag** This field is used for indicate block guard time interrupt status flag in receive direction. 0 = Block guard time interrupt did not occur. 1 = Block guard time interrupt occurred.

		Note 1: This bit is valid only when RXBGTEN (SCn_ALTCTL[12]) is enabled. **Note 2:** This bit can be cleared by writing 1 to it.
[5]	**TMR2IF**	**Timer2 Interrupt Status Flag** This field is used for Timer2 interrupt status flag. 0 = Timer2 interrupt did not occur. 1 = Timer2 interrupt occurred. **Note:** This bit can be cleared by writing 1 to it.
[4]	**TMR1IF**	**Timer1 Interrupt Status Flag** This field is used for Timer1 interrupt status flag. 0 = Timer1 interrupt did not occur. 1 = Timer1 interrupt occurred. **Note:** This bit can be cleared by writing 1 to it.
[3]	**TMR0IF**	**Timer0 Interrupt Status Flag** This field is used for Timer0 interrupt status flag. 0 = Timer0 interrupt did not occur. 1 = Timer0 interrupt occurred. **Note:** This bit can be cleared by writing 1 to it.
[2]	**TERRIF**	**Transfer Error Interrupt Status Flag** This field is used for transfer error interrupt status flag. The transfer error states is at SCn_STATUS register which includes receiver break error BEF (SCn_STATUS[6]), frame error FEF (SCn_STATUS[5], parity error PEF (SCn_STATUS[4] and receive buffer overflow error RXOV (SCn_STATUS[0]), transmit buffer overflow error TXOV (SCn_STATUS[8]), receiver retry over limit error RXOVERR (SCn_STATUS[22] or transmitter retry over limit error TXOVERR (SCn_STATUS[30]). 0 = Transfer error interrupt did not occur. 1 = Transfer error interrupt occurred. **Note 1:** This field is the status flag of BEF, FEF, PEF, RXOV, TXOV, RXOVERR or TXOVERR. **Note 2:** This bit can be cleared by writing 1 to it.
[1]	**TBEIF**	**Transmit Buffer Empty Interrupt Status Flag (Read Only)** This field is used for transmit buffer empty interrupt status flag. 0 = Transmit buffer is not empty. 1 = Transmit buffer is empty. **Note:** This bit is read only. If user wants to clear this bit, user must write data to DAT (SCn_DAT[7:0]) and then this bit will be cleared automatically.
[0]	**RDAIF**	**Receive Data Reach Interrupt Status Flag (Read Only)** This field is used for received data reaching trigger level RXTRGLV (SCn_CTL[7:6]) interrupt status flag. 0 = Number of receive buffer is less than RXTRGLV setting. 1 = Number of receive buffer data equals the RXTRGLV setting. **Note:** This bit is read only. If user reads data from SCn_DAT and receiver buffer data byte number is less than RXTRGLV, this bit will be cleared automatically.

SC Transfer Status Register (SC_STATUS)

Register	Offset	R/W	Description	Reset Value
SC_STATUS	SCn_BA+0x20	R/W	SC Transfer Status Register	0x0000_X202

31	30	29	28	27	26	25	24
TXACT	TXOVERR	TXRERR	Reserved		TXPOINT		
23	22	21	20	19	18	17	16
RXACT	RXOVERR	RXRERR	Reserved		RXPOINT		
15	14	13	12	11	10	9	8
Reserved		CDPINSTS	CINSERT	CREMOVE	TXFULL	TXEMPTY	TXOV
7	6	5	4	3	2	1	0
Reserved	BEF	FEF	PEF	Reserved	RXFULL	RXEMPTY	RXOV

Bits	Description	
[31]	**TXACT**	**Transmit in Active Status Flag (Read Only)** This bit indicates TX transmit status. 0 = This bit is cleared automatically when TX transfer is finished or the last byte transmission has completed. 1 = Transmit is active and this bit is set by hardware when TX transfer is in active and the STOP bit of the last byte has not been transmitted.
[30]	**TXOVERR**	**Transmitter over Retry Error** This bit is used for transmitter retry counts over than retry number limitation. 0 = Transmitter retries counts is less than TXRTY (SCn_CTL[22:20]) + 1. 1 = Transmitter retries counts is equal or over to TXRTY (SCn_CTL[22:20]) + 1. **Note:** This bit can be cleared by writing 1 to it.
[29]	**TXRERR**	**Transmitter Retry Error** This bit is used for indicate transmitter error retry and set by hardware. 0 = No TX retry transfer. 1 = TX has any error and retries transfer. **Note 1:** This bit can be cleared by writing 1 to it. **Note 2:** This bit is a flag and cannot generate any interrupt to CPU.
[28:27]	**Reserved**	Reserved.
[26:24]	**TXPOINT**	**Transmit Buffer Pointer Status (Read Only)** This field indicates the TX buffer pointer status. When CPU writes data into SCn_DAT, TXPOINT increases one. When one byte of TX buffer is transferred to transmitter shift register, TXPOINT decreases one.
[23]	**RXACT**	**Receiver in Active Status Flag (Read Only)** This bit indicates RX transfer status. 0 = This bit is cleared automatically when RX transfer is finished. 1 = This bit is set by hardware when RX transfer is in active.
[22]	**RXOVERR**	**Receiver over Retry Error** This bit is used for receiver retry counts over than retry number limitation.

		0 = Receiver retries counts is less than RXRTY (SCn_CTL[18:16]) + 1. 1 = Receiver retries counts is equal or over than RXRTY (SCn_CTL[18:16]) + 1. **Note 1:** This bit can be cleared by writing 1 to it. **Note 2:** If CPU enables receiver retries function by setting RXRTYEN (SCn_CTL[19]), hardware will not set this flag.
[21]	**RXRERR**	**Receiver Retry Error** This bit is used for receiver error retry and set by hardware. 0 = No RX retry transfer. 1 = RX has any error and retries transfer. **Note 1:** This bit can be cleared by writing 1 to it. **Note 2** This bit is a flag and cannot generate any interrupt to CPU. **Note 3:** If CPU enables receiver retries function by setting RXRTYEN (SCn_CTL[19]), hardware will not set this flag.
[20:19]	**Reserved**	Reserved.
[18:16]	**RXPOINT**	**Receive Buffer Pointer Status (Read Only)** This field indicates the RX buffer pointer status. When SC controller receives one byte from external device, RXPOINT increases one. When one byte of RX buffer is read by CPU, RXPOINT decreases one.
[15:14]	**Reserved**	Reserved.
[13]	**CDPINSTS**	**Card Detect Pin Status (Read Only)** This bit is the pin status of SCn_CD. 0 = The SCn_CD pin state at low. 1 = The SCn_CD pin state at high.
[12]	**CINSERT**	**Card Insert Status of SCn_CD Pin** This bit is set whenever card has been inserted. 0 = No effect. 1 = Card insert. **Note 1:** This bit can be cleared by writing “1” to it. **Note 2:** The card detect function will start after SCEN (SCn_CTL[0]) set.
[11]	**CREMOVE**	**Card Removal Status of SCn_CD Pin** This bit is set whenever card has been removal. 0 = No effect. 1 = Card removed. **Note 1:** This bit can be cleared by writing “1” to it. **Note 2:** Card detect function will start after SCEN (SCn_CTL[0]) set.
[10]	**TXFULL**	**Transmit Buffer Full Status Flag (Read Only)** This bit indicates TX buffer full or not. 0 = TX buffer count is less than 4. 1 = TX buffer count equals to 4.
[9]	**TXEMPTY**	**Transmit Buffer Empty Status Flag (Read Only)** This bit indicates TX buffer empty or not. 0 = TX buffer is not empty. 1 = TX buffer is empty; it means the last byte of TX buffer has been transferred to Transmitter Shift Register. **Note:** This bit will be cleared when writing data into DAT (SCn_DAT[7:0]).
[8]	**TXOV**	**Transmit Overflow Error Interrupt Status Flag** This bit is set when TX buffer overflow.

		0 = TX buffer is not overflow. 1 = TX buffer is overflow when TX buffer is full and an additional write operation to DAT (SCn_DAT[7:0]). **Note:** This bit can be cleared by writing 1 to it.
[7]	**Reserved**	Reserved.
[6]	**BEF**	**Receiver Break Error Status Flag** This bit is set to logic 1 whenever the received data input (RX) held in the "spacing state" (logic 0) is longer than a full word transmission time (that is, the total time of “start bit” + “data bits” + “parity bit” + “stop bits”). 0 = Receiver break error flag did not occur. 1 = Receiver break error flag occurred. **Note 1:** This bit can be cleared by writing 1 to it. **Note 2:** If CPU sets receiver retries function by setting RXRTYEN (SCn_CTL[19]), hardware will not set this flag.
[5]	**FEF**	**Receiver Frame Error Status Flag** This bit is set to logic 1 whenever the received character does not have a valid "stop bit" (that is, the stop bit following the last data bit or parity bit is detected as logic 0). 0 = Receiver frame error flag did not occur. 1 = Receiver frame error flag occurred. **Note 1:** This bit can be cleared by writing 1 to it. **Note 2:** If CPU sets receiver retries function by setting RXRTYEN (SCn_CTL[19]), hardware will not set this flag.
[4]	**PEF**	**Receiver Parity Error Status Flag** This bit is set to logic 1 whenever the received character does not have a valid "parity bit". 0 = Receiver parity error flag did not occur. 1 = Receiver parity error flag occurred. **Note 1:** This bit can be cleared by writing 1 to it. **Note 2:** If CPU sets receiver retries function by setting RXRTYEN (SCn_CTL[19]), hardware will not set this flag.
[3]	**Reserved**	Reserved.
[2]	**RXFULL**	**Receive Buffer Full Status Flag (Read Only)** This bit indicates Rx buffer full or not. 0 = RX buffer count is less than 4. 1 = RX buffer count equals to 4.
[1]	**RXEMPTY**	**Receive Buffer Empty Status Flag (Read Only)** This bit indicates RX buffer empty or not. 0 = RX buffer is not empty. 1 = RX buffer is empty, it means the last byte of RX buffer has read from DAT (SCn_DAT[7:0]) by CPU.
[0]	**RXOV**	**Receive Overflow Error Status Flag** This bit is set when RX buffer overflow. 0 = RX buffer is not overflow. 1 = RX buffer is overflow when the number of received bytes is greater than RX buffer size (4 bytes). **Note:** This bit can be cleared by writing 1 to it.

SC Pin Control State Register (SC_PINCTL)

Register	Offset	R/W	Description	Reset Value
SC_PINCTL	SCn_BA+0x24	R/W	SC Pin Control State Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	SYNC	Reserved					
23	22	21	20	19	18	17	16
Reserved					RSTSTS	PWRSTS	DATASTS
15	14	13	12	11	10	9	8
Reserved				PWRINV	Reserved	SCDATA	Reserved
7	6	5	4	3	2	1	0
Reserved	CLKKEEP	Reserved				RSTEN	PWREN

Bits	Description	
[31]	**Reserved**	Reserved.
[30]	**SYNC**	**SYNC Flag Indicator (Read Only)** Due to synchronization, user should check this bit when writing a new value to SCn_PINCTL register. 0 = Synchronizing is completed; user can write new data to SCn_PINCTL register. 1 = Last value is synchronizing.
[29:19]	**Reserved**	Reserved.
[18]	**RSTSTS**	**SCn_RST Pin Status (Read Only)** This bit is the pin status of SCn_RST. 0 = SCn_RST pin is low. 1 = SCn_RST pin is high.
[17]	**PWRSTS**	**SCn_PWR Pin Status (Read Only)** This bit is the pin status of SCn_PWR. 0 = SCn_PWR pin to low. 1 = SCn_PWR pin to high.
[16]	**DATASTS**	**SCn_DATA Pin Status (Read Only)** This bit is the pin status of SCn_DATA. 0 = The SCn_DATA pin status is low. 1 = The SCn_DATA pin status is high.
[15:12]	**Reserved**	Reserved.
[11]	**PWRINV**	**SCn_PWR Pin Inverse** This bit is used for inverse the SCn_PWR pin. There are four kinds of combination for SCn_PWR pin setting by PWRINV (SCn_PINCTL[11]) and PWREN (SCn_PINCTL[0]). PWRINV (SCn_PINCTL[11]) is bit 1 and PWREN (SCn_PINCTL[0]) is bit 0 and all conditions as below list, 00 = SCn_PWR pin is 0. 01 = SCn_PWR pin is 1.

		10 = SCn_PWR pin is 1. 11 = SCn_PWR pin is 0. **Note:** User must select PWRINV (SCn_PINCTL[11]) before smart card is enabled by SCEN (SCn_CTL[0]).
[10]	**Reserved**	Reserved.
[9]	**SCDATA**	**SCn_DATA Pin Signal** This bit is the signal status of SCn_DATA but user can drive SCn_DATA pin to high or low by setting this bit. 0 = Drive SCn_DATA pin to low. 1 = Drive SCn_DATA pin to high. Read this field to get SCn_DATA signal status. 0 = SCn_DATA signal status is low. 1 = SCn_DATA signal status is high. **Note:** When SC is at activation, warm reset or deactivation mode, this bit will be changed automatically. Thus, do not fill in this field when SC is in these modes.
[8:7]	**Reserved**	Reserved.
[6]	**CLKKEEP**	**SC Clock Enable Bit** 0 = SC clock generation Disabled. 1 = SC clock always keeps free running. **Note:** When operating in activation, warm reset or deactivation mode, this bit will be changed automatically. Thus, do not fill in this field when operating in these modes.
[5:2]	**Reserved**	Reserved.
[1]	**RSTEN**	**SCn_RST Pin Signal** User can set RSTEN (SCn_PINCTL[1]) to decide SCn_RST pin is in high or low level. Write this field to drive SCn_RST pin. 0 = Drive SCn_RST pin to low. 1 = Drive SCn_RST pin to high. Read this field to get SCn_RST signal status. 0 = SCn_RST signal status is low. 1 = SCn_RST signal status is high. **Note:** When operating at activation, warm reset or deactivation mode, this bit will be changed automatically. Thus, do not fill in this field when operating in these modes.
[0]	**PWREN**	**SCn_PWR Pin Signal** User can set PWRINV (SCn_PINCTL[11]) and PWREN (SCn_PINCTL[0]) to decide SCn_PWR pin is in high or low level. Write this field to drive SCn_PWR pin Refer PWRINV (SCn_PINCTL[11]) description for programming SCn_PWR pin voltage level. Read this field to get SCn_PWR signal status. 0 = SCn_PWR signal status is low. 1 = SCn_PWR signal status is high. **Note:** When operating at activation, warm reset or deactivation mode, this bit will be changed automatically. Thus, do not fill in this field when operating in these modes.

SC Timer0 Control Register (SC_TMRCTL0)

Register	Offset	R/W	Description	Reset Value
SC_TMRCTL0	SCn_BA+0x28	R/W	SC Internal Timer0 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
SYNC	Reserved			OPMODE			
23	22	21	20	19	18	17	16
CNT							
15	14	13	12	11	10	9	8
CNT							
7	6	5	4	3	2	1	0
CNT							

Bits	Description	
[31]	**SYNC**	**SYNC Flag Indicator (Read Only)** Due to synchronization, user should check this bit when writing a new value to the SCn_TMRCTL0 register. 0 = Synchronizing is completed; user can write new data to SCn_TMRCTL0 register. 1 = Last value is synchronizing.
[30:28]	**Reserved**	Reserved.
[27:24]	**OPMODE**	**Timer0 Operation Mode Selection** This field indicates the internal 24-bit Timer0 operation selection. Refer to Table 6.25-3 for programming Timer0.
[23:0]	**CNT**	**Timer0 Counter Value** This field indicates the internal Timer0 counter values. **Note:** Unit of Timer0 counter is ETU base.

SC Timer1 Control Register (SC_TMRCTL1)

Register	Offset	R/W	Description	Reset Value
SC_TMRCTL1	SCn_BA+0x2C	R/W	SC Internal Timer1 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
SYNC	Reserved			OPMODE			
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
CNT							

Bits	Description	
[31]	**SYNC**	**SYNC Flag Indicator (Read Only)** Due to synchronization, software should check this bit when writing a new value to SCn_TMRCTL1 register. 0 = Synchronizing is completed; user can write new data to SCn_TMRCTL1 register. 1 = Last value is synchronizing.
[30:28]	**Reserved**	Reserved.
[27:24]	**OPMODE**	**Timer 1 Operation Mode Selection** This field indicates the internal 8-bit Timer1 operation selection. Refer to Table 6.25-3 for programming Timer1.
[23:8]	**Reserved**	Reserved.
[7:0]	**CNT**	**Timer 1 Counter Value** This field indicates the internal Timer1 counter values. **Note:** Unit of Timer1 counter is ETU base.

SC Timer2 Control Register (SC_TMRCTL2)

Register	Offset	R/W	Description	Reset Value
SC_TMRCTL2	SCn_BA+0x30	R/W	SC Internal Timer2 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
SYNC	Reserved			OPMODE			
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
CNT							

Bits	Description	
[31]	**SYNC**	**SYNC Flag Indicator (Read Only)** Due to synchronization, user should check this bit when writing a new value to SCn_TMRCTL2 register. 0 = Synchronizing is completed; user can write new data to SCn_TMRCTL2 register. 1 = Last value is synchronizing.
[30:28]	**Reserved**	Reserved.
[27:24]	**OPMODE**	**Timer 2 Operation Mode Selection** This field indicates the internal 8-bit Timer2 operation selection Refer to Table 6.25-3 for programming Timer2.
[23:8]	**Reserved**	Reserved.
[7:0]	**CNT**	**Timer 2 Counter Value** This field indicates the internal Timer2 counter values. **Note:** Unit of Timer2 counter is ETU base.

SC UART Mode Control Register (SC_UARTCTL)

Register	Offset	R/W	Description	Reset Value
SC_UARTCTL	SCn_BA+0x34	R/W	SC UART Mode Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
OPE	PBOFF	WLS		Reserved			UARTEN

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**OPE**	**Odd Parity Enable Bit** This is used for odd/even parity selection. 0 = Even number of logic 1 are transmitted or check the data word and parity bits in receiving mode. 1 = Odd number of logic 1 are transmitted or check the data word and parity bits in receiving mode. **Note:** This bit has effect only when PBOFF bit is 0.
[6]	**PBOFF**	**Parity Bit Disable Bit** Sets this bit is used for disable parity check function. 0 = Parity bit is generated or checked between the “last data word bit” and “stop bit” of the serial data. 1 = Parity bit is not generated (transmitting data) or checked (receiving data) during transfer. **Note:** In smart card mode, this field must be 0 (default setting is with parity bit).
[5:4]	**WLS**	**Word Length Selection** This field is used for select UART data length. 00 = Word length is 8 bits. 01 = Word length is 7 bits. 10 = Word length is 6 bits. 11 = Word length is 5 bits. **Note:** In smart card mode, this WLS must be 00.
[3:1]	**Reserved**	Reserved.
[0]	**UARTEN**	**UART Mode Enable Bit** Sets this bit to enable UART mode function. 0 = Smart Card mode. 1 = UART mode. **Note 1:** When operating in UART mode, user must set CONSEL (SCn_CTL[5:4]) = 00 and AUTOCEN (SCn_CTL[3]) = 0. **Note 2:** When operating in Smart Card mode, user must set UARTEN (SCn_UARTCTL[0]) = 0. **Note 3:** When UART mode is enabled, hardware will generate a reset to reset FIFO and internal state

Bits	Description	
		machine.

SC Activation Control Register (SC_ACTCTL)

Register	Offset	R/W	Description	Reset Value
SC_ACTCTL	SCn_BA+0x4C	R/W	SC Activation Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**T1EXT**				

Bits	Description	
[31:5]	**Reserved**	Reserved.
[4:0]	**T1EXT**	**T1 Extend Time of Hardware Activation** This field provide the configurable cycles to extend the activation time T1 period. The cycle scaling factor is 2048. Extend cycles = (filled value * 2048) cycles. Refer to SC activation sequence in Figure 6.25-4. For example, SCLK = 4 MHz, each cycle = 0.25us. Filled 20 to this field Extend time = 20 * 2048 * 0.25us = 10.24 ms. **Note:** Setting 0 to this field conforms to the protocol ISO/IEC 7816-3

6.26 I²C Serial Interface Controller (I²C)

6.26.1 Overview

I²C is a two-wire, bi-directional serial bus that provides a simple and efficient method of data exchange between devices. The I²C standard is a true multi-master bus including collision detection and arbitration that prevents data corruption if two or more masters attempt to control the bus simultaneously.

There are six sets of I²C controllers that support Power-down wake-up function.

6.26.2 Features

The I²C bus uses two wires (SDA and SCL) to transfer information between devices connected to the bus. The main features of the I²C bus include:

- Supports up to six I²C ports
- Master/Slave mode
- Bidirectional data transfer between masters and slaves
- Multi-master bus (no central master)
- Supports Standard mode (100 kbps), Fast mode (400 kbps) and Fast mode plus (1 Mbps)
- Arbitration between simultaneously transmitting masters without corruption of serial data on the bus
- Serial clock synchronization allow devices with different bit rates to communicate via one serial bus
- Serial clock synchronization used as a handshake mechanism to suspend and resume serial transfer
- Built-in 14-bit time-out counter requesting the I²C interrupt if the I²C bus hangs up and timer-out counter overflow
- Programmable clocks allow for versatile rate control
- Supports 7-bit addressing and 10-bit addressing mode
- Supports multiple address recognition (four slave address with mask option)
- Supports Power-down wake-up function
- Supports PDMA with one buffer capability
- Supports setup/hold time programmable

6.26.3 Block Diagram

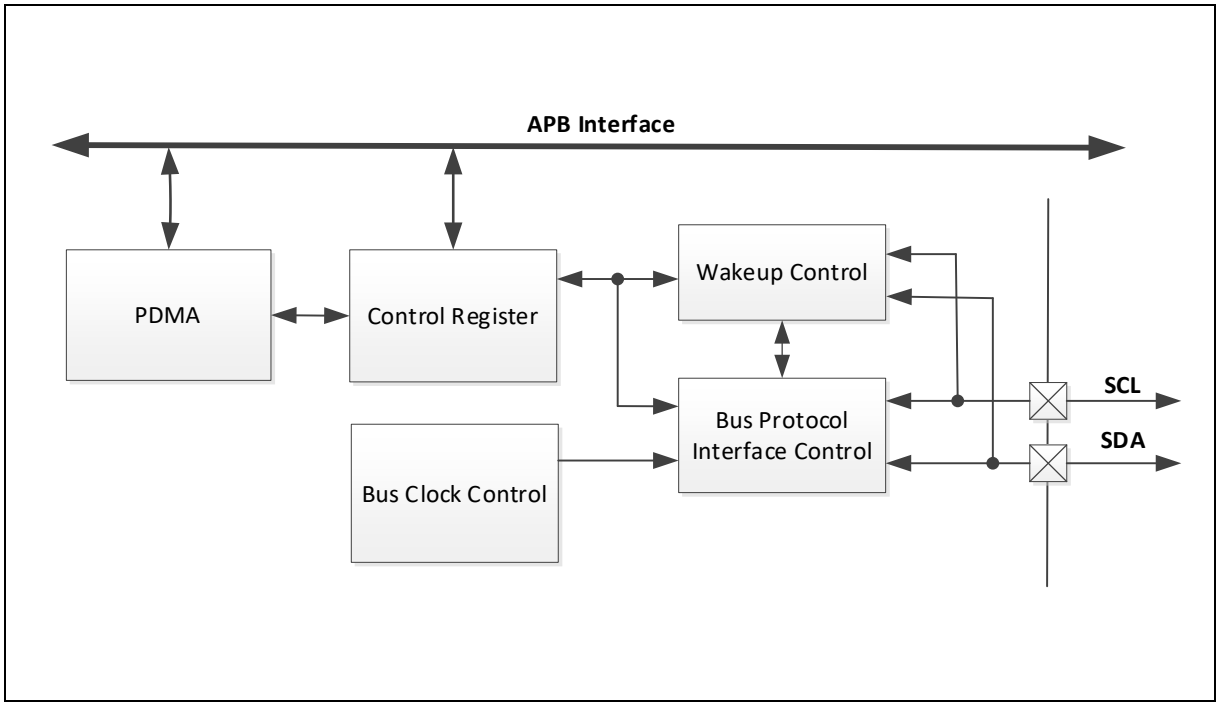


Figure 6.26-1 I²C Controller Block Diagram

6.26.4 Basic Configuration

6.26.4.1 I2C0 Basic Configuration

- Clock source Configuration
  - Enable I2C0 peripheral clock in I2C0CKEN (CLK_APBCLK1[0]).
- Reset Configuration
  - Reset I2C0 controller in I2C0RST (SYS_IPRST1[8]).
- Pin configuration

Group	Pin Name	GPIO	MFP
I2C0	I2C0_SCL	PM.9,	MFP4
		PD.7	MFP6
	I2C0_SDA	PM.8,	MFP4
		PD.6	MFP6

6.26.4.2 I2C1 Basic Configuration

- Clock Source Configuration
  - Enable I2C1 peripheral clock in I2C1CKEN (CLK_APBCLK1[1]).
- Reset Configuration
  - Reset I2C1 controller in I2C1RST (SYS_IPRST1[9]).
- Pin Configuration

Group	Pin Name	GPIO	MFP

I2C1	I2C1_SCL	PN.5	MFP4
		PB.11	MFP12
	I2C1_SDA	PN.4	MFP4
		PB.10	MFP12

6.26.4.3 I2C2 Basic Configuration

- Clock source Configuration
  - Enable I2C2 peripheral clock in I2C2CKEN (CLK_APBCLK1[2]).
- Reset Configuration
  - Reset I2C2 controller in I2C2RST (SYS_IPRST1[10]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
I2C2	I2C2_SCL	PL.13	MFP10
		PM.13	MFP5
		PI.1, PJ.1, PD.5, PN.1, PB.9	MFP4
	I2C2_SDA	PL.12	MFP10
		PM.12	MFP5
		PI.0, PJ.0, PD.4, PN.0, PB.8	MFP4

6.26.4.4 I2C3 Basic Configuration

- Clock source Configuration
  - Enable I2C3 peripheral clock in I2C3CKEN (CLK_APBCLK1[3]).
- Reset Configuration
  - Reset I2C3 controller in I2C3RST (SYS_IPRST1[11]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
I2C3	I2C3_SCL	PM.15	MFP6
		PI.5, PL.1, PK.9, PJ.5, PB.13	MFP4
		PG.9	MFP7
	I2C3_SDA	PM.14	MFP6
		PI.4, PL.0, PK.8, PJ.4, PB.12	MFP4
		PG.8	MFP7

6.26.4.5 I2C4 Basic Configuration

- Clock source Configuration
  - Enable I2C4 peripheral clock in I2C4CKEN (CLK_APBCLK1[4]).

- Reset Configuration
  - Reset I2C4 controller in I2C4RST (SYS_IPRST3[18]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
I2C4	I2C4_SCL	PB.15	MFP5
		PD.13, PG.1, PK.1, PL.5, PM.1, PJ.9, PK.5, PC.1, PK.13	MFP4
		PF.7	MFP10
	I2C4_SDA	PB.14	MFP5
		PD.12, PA.15, PK.0, PL.4, PM.0, PJ.8, PC.0, PK.12	MFP4
		PF.6	MFP10

6.26.4.6 I2C5 Basic Configuration

- Clock source Configuration
  - Enable I2C5 peripheral clock in I2C5CKEN (CLK_APBCLK1[5]).
- Reset Configuration
  - Reset I2C5 controller in I2C5RST (SYS_IPRST3[19]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
I2C5	I2C5_SCL	PJ.13, PL.9, PM.5, PC.5, PN.13, PF.9	MFP4
		PK.15	MFP10
	I2C5_SDA	PJ.12, PL.8, PM.4, PC.4, PN.12, PK.14, PF.8	MFP4
		PK.14	MFP10

6.26.5 Functional Description

On I²C bus, data is transferred between a Master and a Slave. Data bits transfer on the SCL and SDA lines are synchronously on a byte-by-byte basis. Each data byte is 8-bit long. There is one SCL clock pulse for each data bit with the MSB being transmitted first, and an acknowledge bit follows each transferred byte. Each bit is sampled during the high period of SCL; therefore, the SDA line may be changed only during the low period of SCL and must be held stable during the high period of SCL. A transition on the SDA line while SCL is high is interpreted as a command (START or STOP). Please refer to Figure 6.26-2 for more detailed I²C BUS Timing.

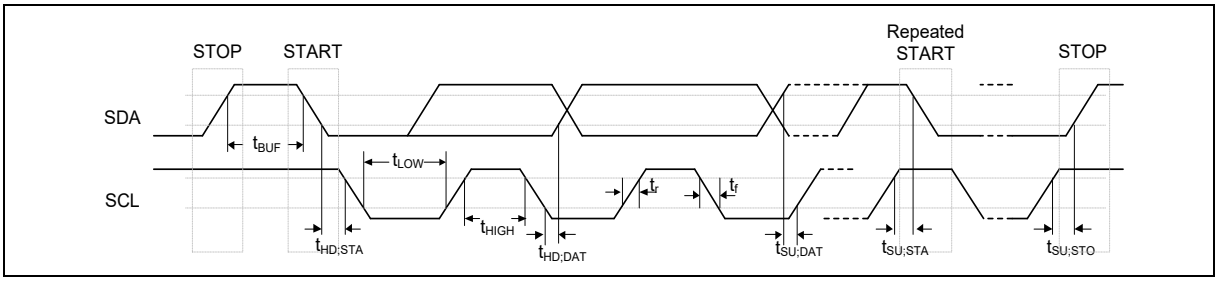


Figure 6.26-2 I²C Bus Timing

The device's on-chip I²C provides the serial interface that meets the I²C bus standard mode specification. The I²C port handles byte transfers autonomously. To enable this port, the bit I2CEN in I2C_CTL0 should be set to '1'. The I²C hardware interfaces to the I²C bus via two pins: SDA and SCL. When I/O pins are used as I²C ports, user must set the pins function to I²C in advance.

Note: Pull-up resistor is needed for I²C operation as the SDA and SCL are open-drain pins.

6.26.5.1 I²C Protocol

Figure 6.26-3 shows the typical I²C protocol. Normally, a standard communication consists of four parts:

- START or Repeated START signal generation
- Slave address and R/W bit transfer
- Data transfer
- STOP signal generation

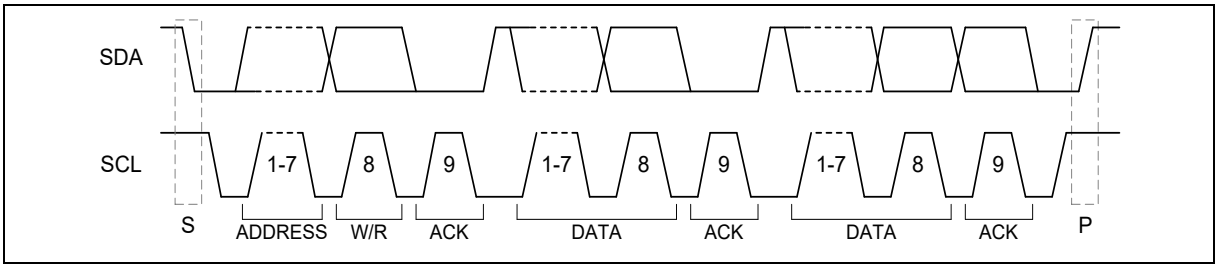


Figure 6.26-3 I²C Protocol

- START or Repeated START signal

  When the bus is free/idle, which means no master device is engaging the bus (both SCL and SDA lines are high), a master can initiate a transfer by sending a START signal. A START signal, usually referred to as the "S" bit, is defined as a HIGH to LOW transition on the SDA line while SCL is HIGH. The START signal denotes the beginning of a new data transmission.

  After having sent the address byte (address and read/write bit), the master may send any number of bytes followed by a stop condition. Instead of sending the stop condition it is also allowed to send another start condition again followed by an address (and of course including a read/write bit) and more data. The start condition is called as Repeat START (Sr). This is defined recursively allowing any number of start conditions to be sent. The purpose of this is to allow combined write/read operations to one or more devices without releasing the bus and thus with the guarantee that the operation is not interrupted. The controller uses this method to communicate with another slave or the same slave in a different transfer direction (e.g. from writing to a device to reading from a device) without releasing the bus.

- STOP signal

The master can terminate the communication by generating a STOP signal. A STOP signal, usually referred to as the “P” bit, is defined as a LOW to HIGH transition on the SDA line while SCL is HIGH.

Figure 6.26-4 shows the waveform of START, Repeat START and STOP.

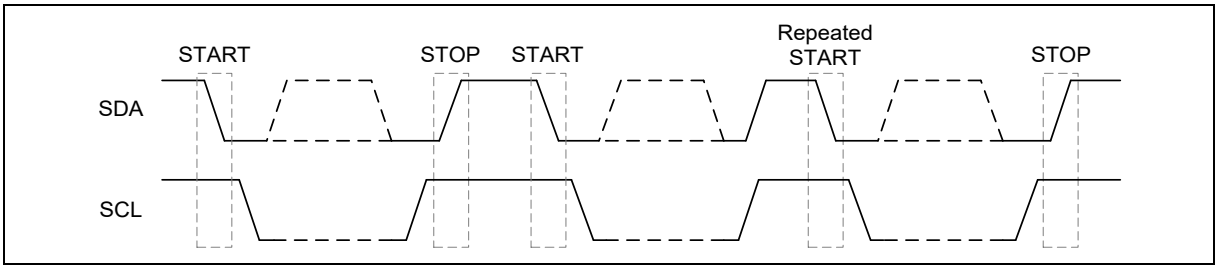


Figure 6.26-4 START and STOP Conditions

- Slave Address Transfer

  After a (Repeated) START condition, the master sends a slave address to identify the target device of the communication. The start address can comprise one or two address bytes (for 7-bit or for 10-bit addressing schemes). After an address byte, a slave sensitive to the transmitted address has to acknowledge the reception.

  Therefore, the slave’s address can be programmed in the device, where it is compared to the received address. In case of a match, the slave answers with an acknowledge (SDA = 0). Slaves that are not targeted answer with a non-acknowledge (SDA = 1). In addition to the match of the programmed address, another address byte value has to be answered with an acknowledge if the slave is capable to handle the corresponding requests.

- Data Transfer

  When a slave receives a correct address with an R/W bit, the data will follow R/W bit specified to transfer. Each transferred byte is followed by an acknowledge bit on the 9th SCL clock cycle. If the slave signals a Not Acknowledge (NACK), the master can generate a STOP signal to abort the data transfer or generate a Repeated START signal and start a new transfer cycle.

  If the master, as a receiving device, does Not Acknowledge (NACK) the slave, the slave releases the SDA line for the master to generate a STOP or Repeated START signal. Figure 6.26-5 and Figure 6.26-6 show the waveform of bit transfer and acknowledge.

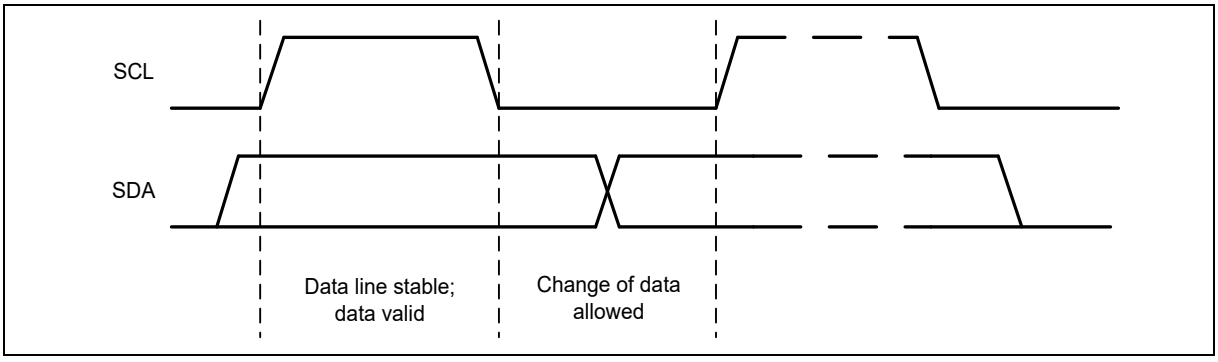


Figure 6.26-5 Bit Transfer on the I²C Bus

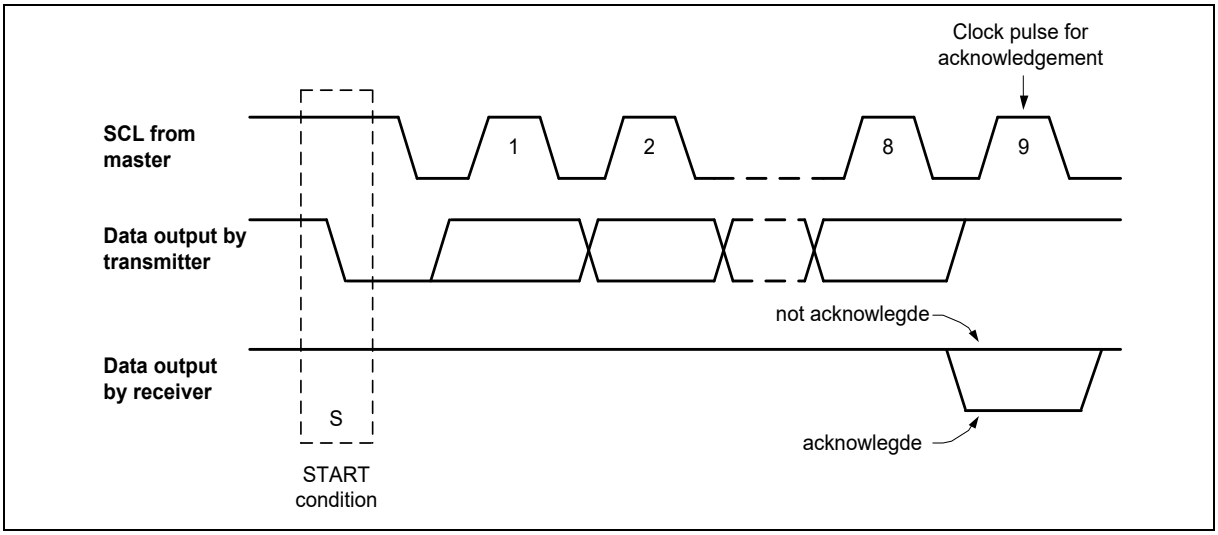


Figure 6.26-6 Acknowledge on the I²C Bus

- Data transfer on I²C bus

Figure 6.26-7 shows a master transmits data to slave by 7-bit. A master addresses a slave with a 7-bit address and 1-bit write index to denote that the master wants to transmit data to the slave. The master keeps transmitting data after the slave returns acknowledge to the master.

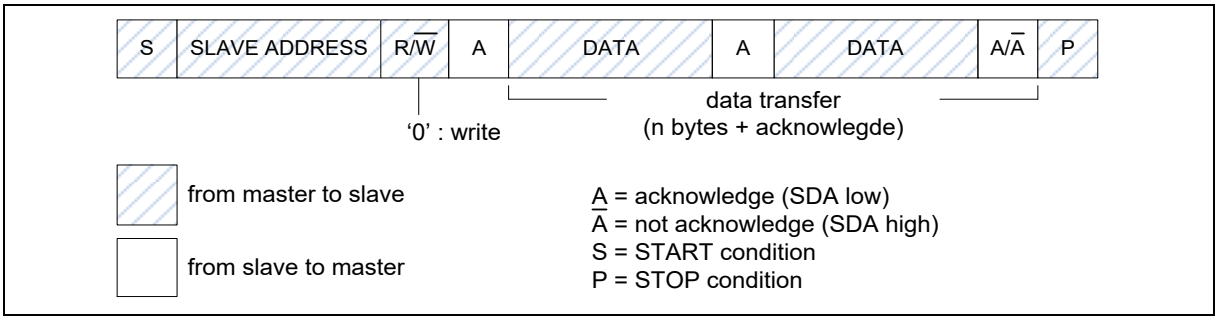


Figure 6.26-7 Master Transmits Data to Slave by 7-bit

Figure 6.26-8 shows a master read data from slave by 7-bit. A master addresses a slave with a 7-bit address and 1-bit read index to denote that the master wants to read data from the slave. The slave will start transmitting data after the slave returns acknowledge to the master.

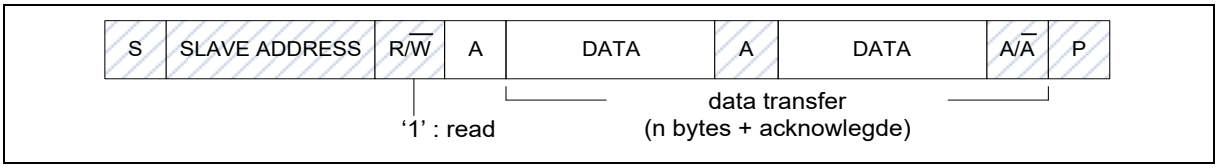


Figure 6.26-8 Master Reads Data from Slave by 7-bit

Figure 6.26-9 shows a master transmits data to slave by 10-bit. A master addresses a slave with a 10-bit address. First byte contains 10-bit address indicator (5'b11110) and 2-bit address with write index, second byte contains 8-bit address. The master keeps transmitting data after the second byte end. Note that 7-bit and 10-bit address device can work on the same bus.

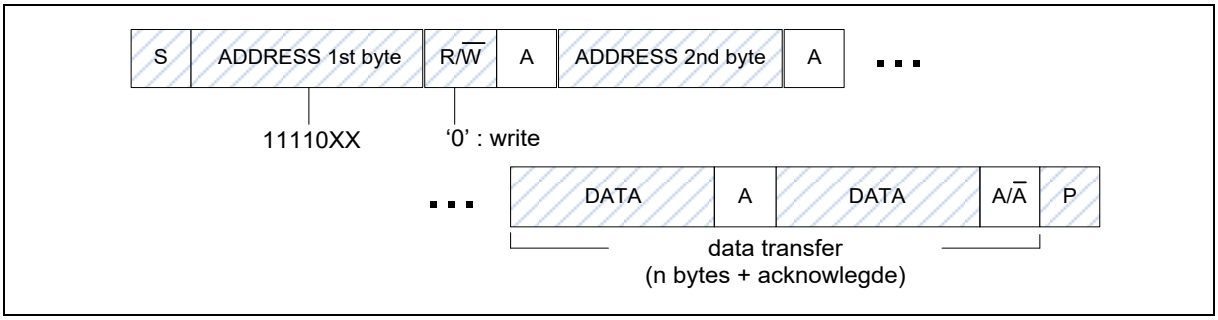


Figure 6.26-9 Master Transmits Data to Slave by 10-bit

Figure 6.26-10 shows a master read data from slave by 10-bit. A master addresses a slave with a 10-bit address. First mater transmits 10-bit address to slave, after that master transmits first byte with read index. The slave will start transmitting data after the first byte with read index.

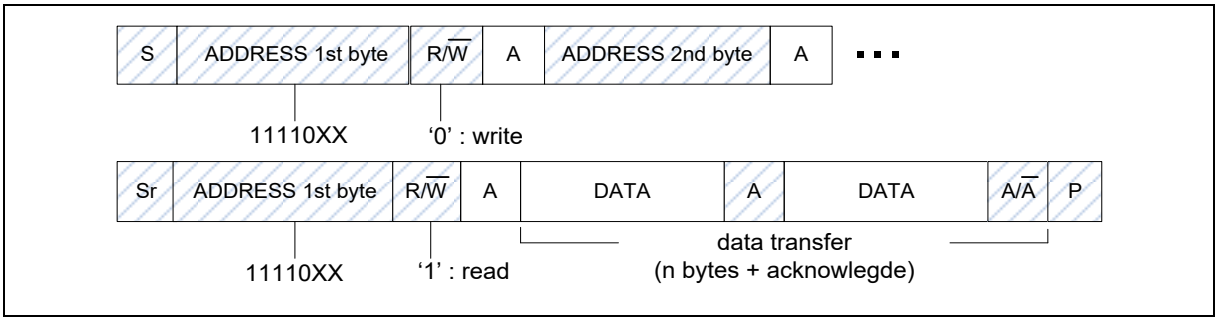


Figure 6.26-10 Master Reads Data from Slave by 10-bit

6.26.5.2 Operation Modes

The on-chip I²C ports support three operation modes, Master, Slave, and General Call Mode.

In a given application, I²C port may operate as a master or as a slave. In Slave mode, the I²C port hardware looks for its own slave address and the general call address. If one of these addresses is detected, and if the slave is willing to receive or transmit data from/to master(by setting the AA bit), acknowledge pulse will be transmitted out on the 9th clock, hence an interrupt is requested on both master and slave devices if interrupt is enabled. When the microcontroller wishes to become the bus master, hardware waits until the bus is free before entering Master mode so that a possible slave action is not be interrupted. If bus arbitration is lost in Master mode, I²C port switches to Slave mode immediately and can detect its own slave address in the same serial transfer.

To control the I²C bus transfer in each mode, user needs to set I2C_CTL0, I2C_DAT registers according to current status code of I2C_STATUS0 register. In other words, for each I²C bus action, user needs to check current status by I2C_STATUS0 register, and then set I2C_CTL0, I2C_DAT registers to take bus action. Finally, check the response status by I2C_STATUS0.

The bits, STA, STO and AA in I2C_CTL0 register are used to control the next state of the I²C hardware after SI flag of I2C_CTL0 [3] register is cleared. Upon completion of the new action, a new status code will be updated in I2C_STATUS0 register and the SI flag of I2C_CTL0 register will be set. But the SI flag will not be set when I²C STOP. If the I²C interrupt control bit INTEN (I2C_CTL0 [7]) is set, appropriate action or software branch of the new status code can be performed in the Interrupt service routine.

Figure 6.26-11 shows the current I²C status code is 0x08, and then set I2C_DATA=SLA+W and (STA,STO,SI,AA) = (0,0,1,x) to send the address to I²C bus. If a slave on the bus matches the address and response ACK, the I2C_STATUS0 will be updated by status code 0x18.

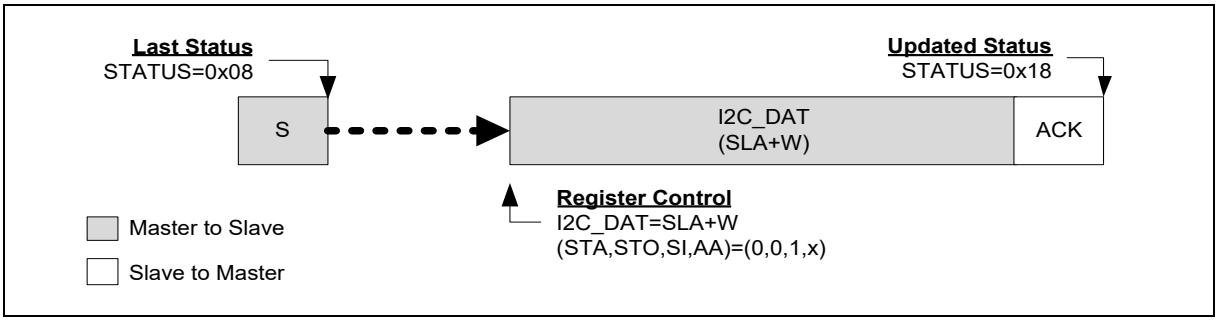


Figure 6.26-11 Control I²C Bus according to the Current I²C Status

Master Mode

In Figure 6.26-12 and Figure 6.26-13, all possible protocols for I²C master are shown. User needs to follow proper path of the flow to implement required I²C protocol.

In other words, user can send a START signal to bus and I²C will be in Master Transmitter (MT) mode (Figure 6.26-12) or Master receiver (MR) mode (Figure 6.26-13) after START signal has been sent successfully and new status code would be 0x08. Followed by START signal, user can send slave address, read/write bit, data and Repeat START, STOP to perform I²C protocol.

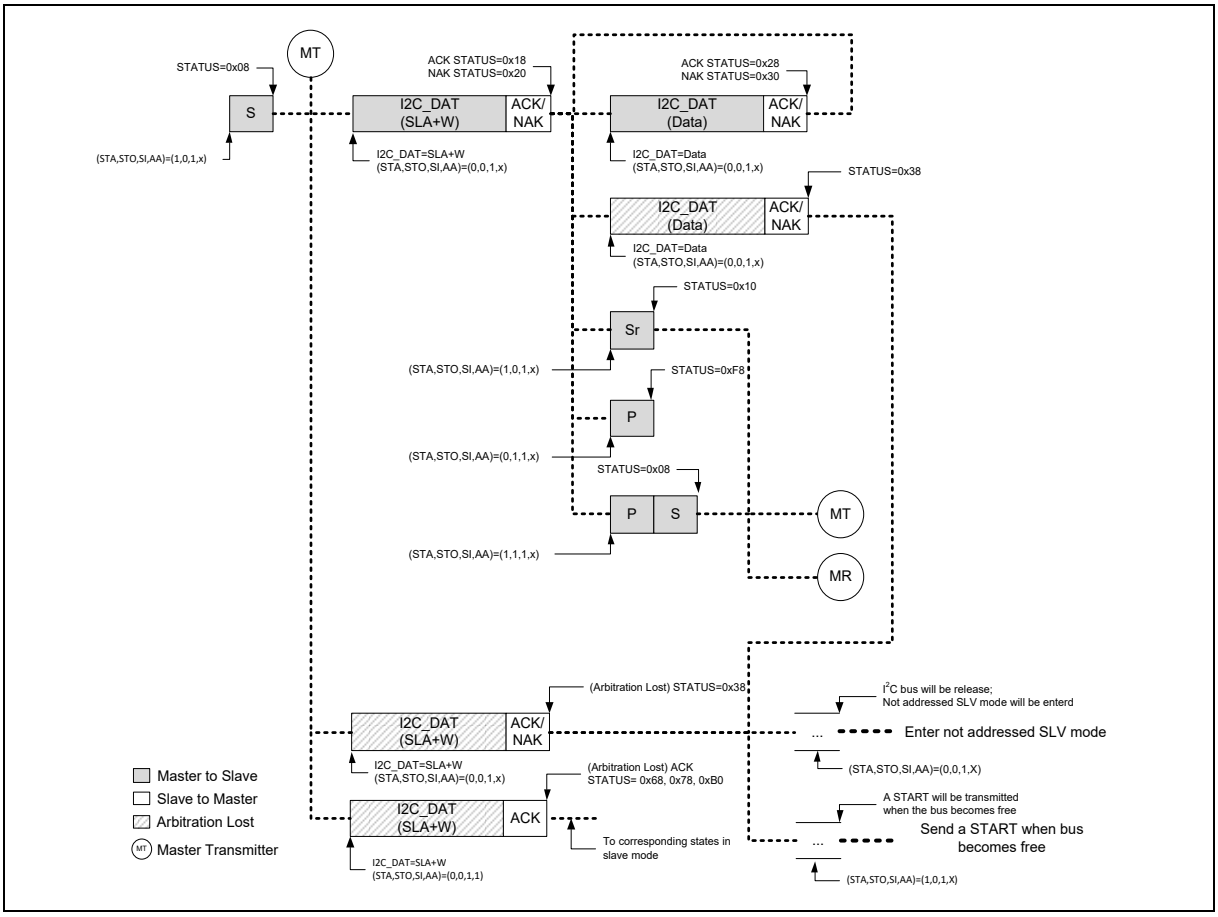


Figure 6.26-12 Master Transmitter Mode Control Flow

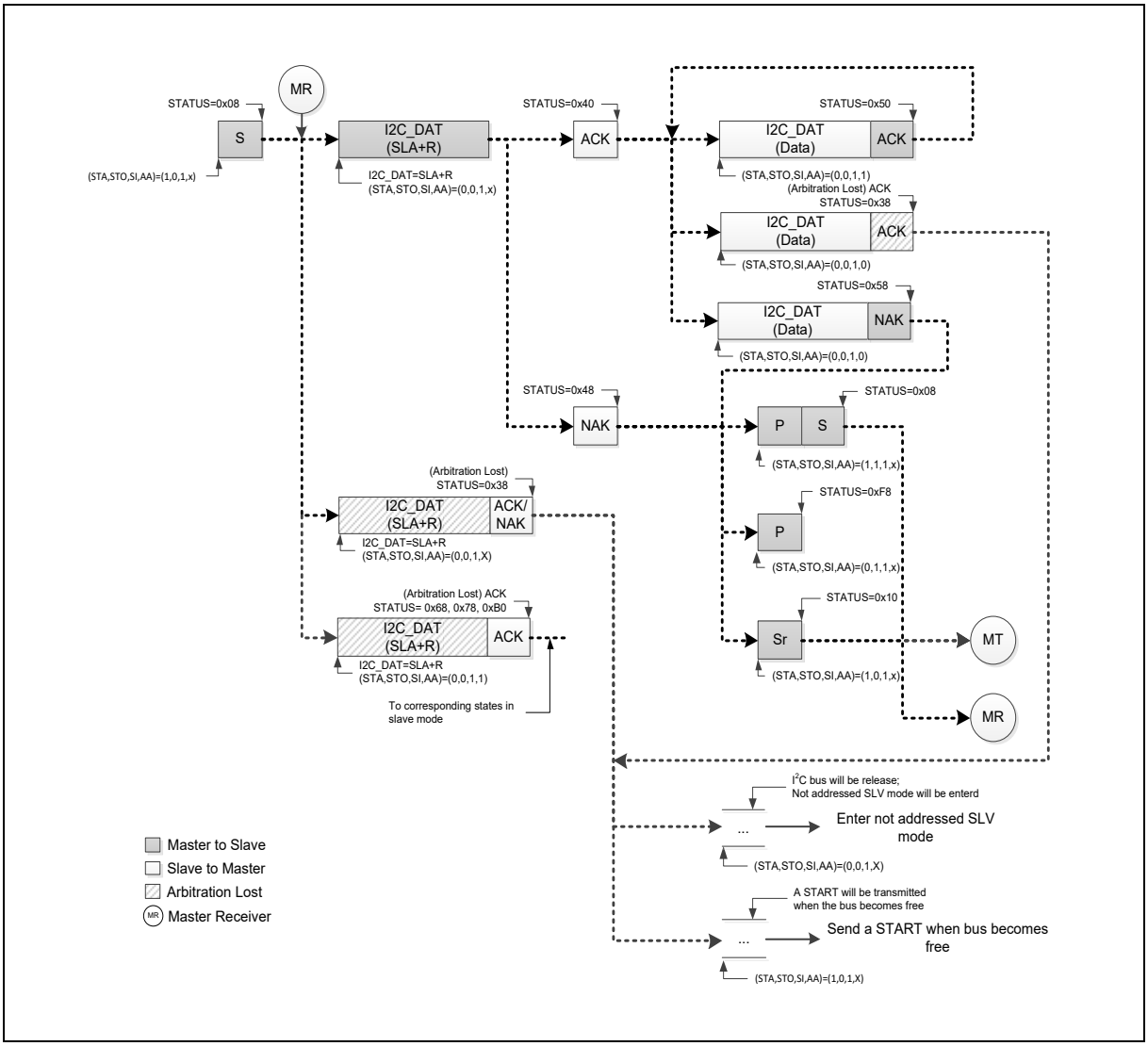


Figure 6.26-13 Master Receiver Mode Control Flow

If the I²C is in Master mode and gets arbitration lost, the status code will be 0x38. In status 0x38, user may set (STA, STO, SI, AA) = (1, 0, 1, X) to send START to re-start Master operation when bus become free. Otherwise, user may set (STA, STO, SI, AA) = (0, 0, 1, X) to release I²C bus and enter not addressed Slave mode.

Slave Mode

When reset default, I²C is not addressed and will not recognize the address on I²C bus. User can set slave address by I2C_ADDRn (n=0~3) and set (STA, STO, SI, AA) = (0, 0, 1, 1) to let I²C recognize the address sent by master. Figure 6.26-14 shows all the possible flow for I²C in Slave mode. Users need to follow a proper flow (as shown in Figure 6.26-14 to implement their own I²C protocol.

If bus arbitration is lost in Master mode, I²C port switches to Slave mode immediately and can detect its own slave address in the same serial transfer. If the detected address is SLA+W (Master want to write data to Slave) after arbitration lost, the status code is 0x68. If the detected address is SLA+R (Master want to read data from Slave) after arbitration lost, the status code is 0xB0.

Note: During I²C communication, the SCL clock will be released when writing '1' to clear SI flag in Slave mode.

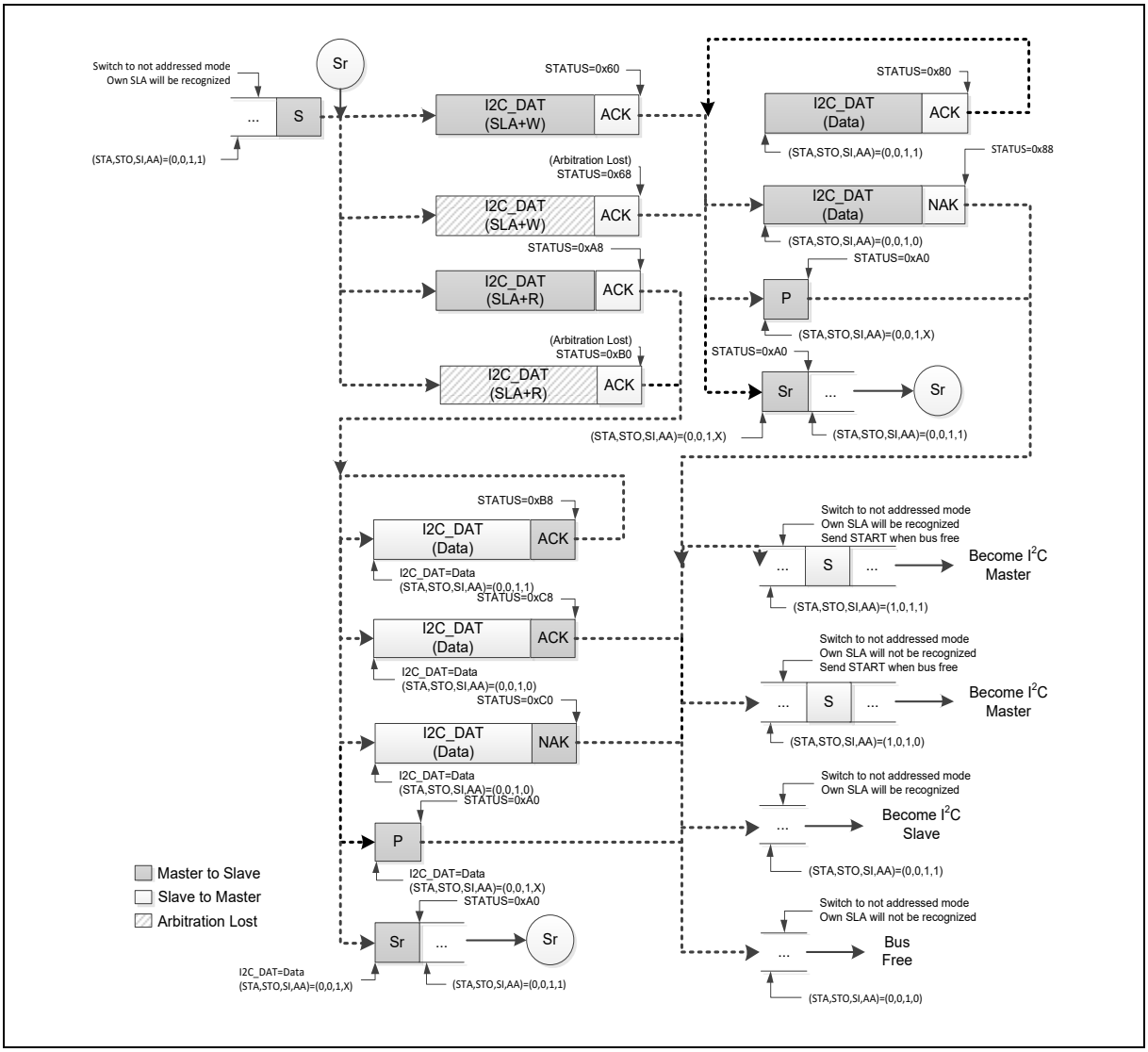


Figure 6.26-14 Slave Mode Control Flow

If I²C is still receiving data in addressed Slave mode but got a STOP or Repeat START, the status code will be 0xA0. User could follow the action for status code 0x88 as shown in the above figure when getting 0xA0 status.

If I²C is still transmitting data in addressed Slave mode but got a STOP or Repeat START, the status code will be 0xA0. User could follow the action for status code 0xC8 as shown in the above figure when getting 0xA0 status.

Note: After slave gets status of 0x88, 0xC8, 0xC0 and 0xA0, slave can switch to not address mode and own SLA will not be recognized. If entering this status, slave will not receive any I²C signal or address from master. At this status, I²C should enter idle mode.

General Call (GC) Mode

If the GC bit (I2C_ADDRn [0]) is set, the I²C port hardware will respond to General Call address (00H). User can clear GC bit to disable general call function. When the GC bit is set and the I²C in Slave mode, it can receive the general call address by 0x00 after master send general call address to I²C bus, then it will follow status of GC mode.

The GC mode can wake up when address matched. Note that the default address is 0x00, but user

must set an address except for 0x00.

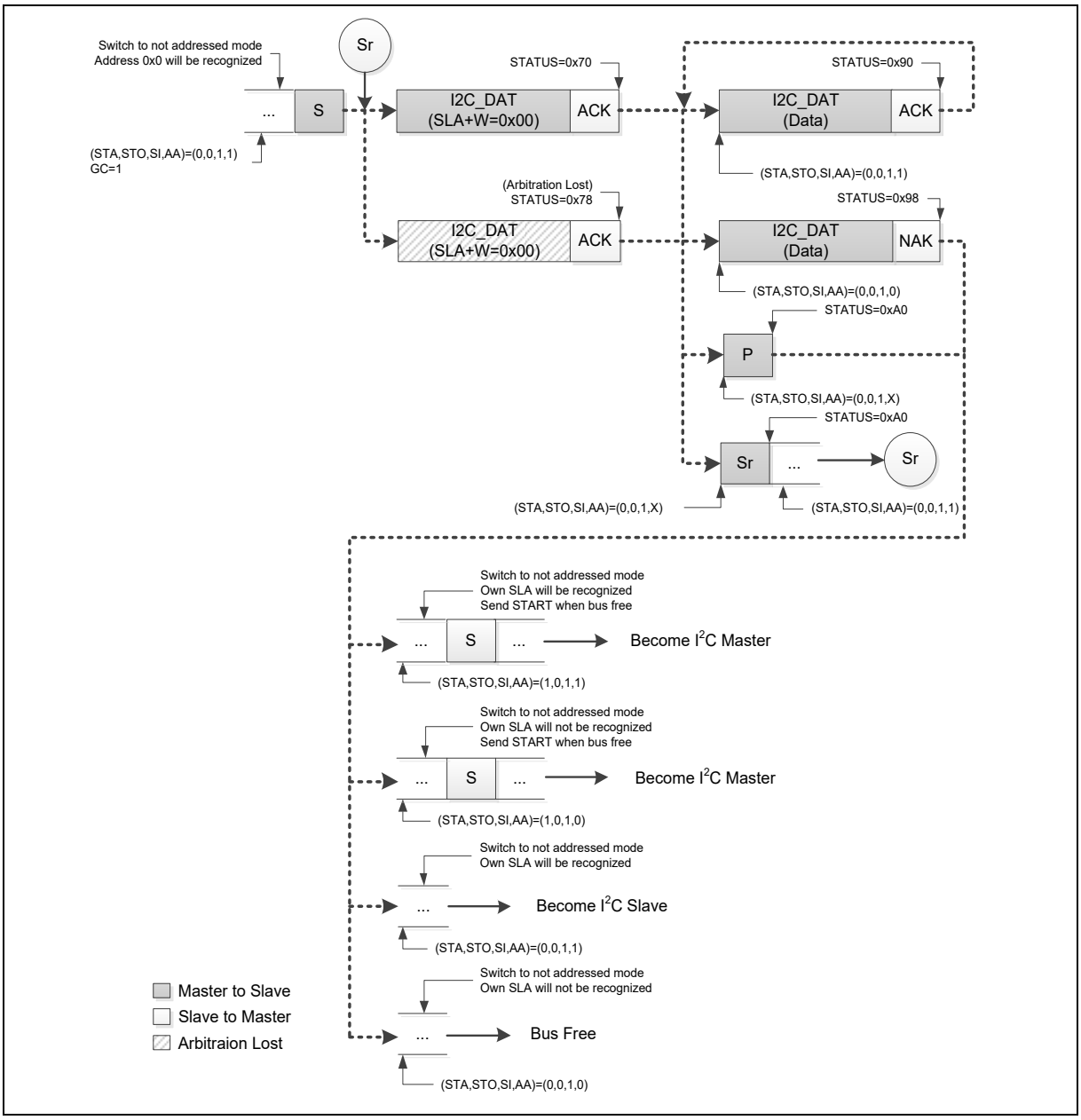


Figure 6.26-15 GC Mode

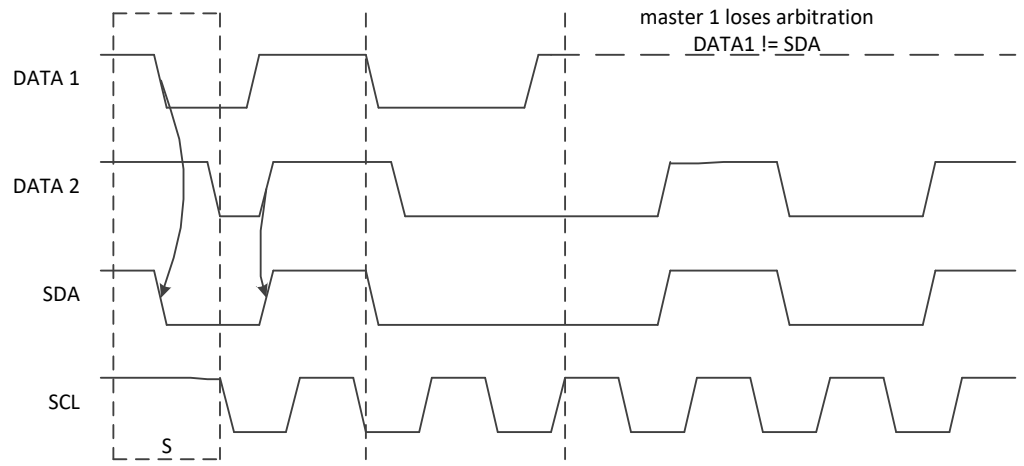
If I²C is still receiving data in GC mode but got a STOP or Repeat START, the status code will be 0xA0. User could follow the action for status code 0x98 in above figure when getting 0xA0 status.

Note: After slave gets status of 0x98 and 0xA0, slave can switch to not address mode and own SLA will not be recognized. If entering this status, slave will not receive any I²C signal or address from master. At this time, the I²C controller should enter idle mode.

Multi-Master

In some applications, there are two or more masters on the same I²C bus to access slaves, and the masters may transmit data simultaneously. The I²C supports multi-master by including collision detection and arbitration to prevent data corruption.

If for some reason two masters initiate command at the same time, the arbitration procedure determines which master wins and can continue with the command. Arbitration is performed on the SDA signal while the SCL signal is high. Each master checks if the SDA signal on the bus corresponds to the generated SDA signal. If the SDA signal on the bus is low but it should be high, then this master has lost arbitration. The device that has lost arbitration can generate SCL pulses until the byte ends and must then release the bus and go into slave mode. The arbitration procedure can continue until all the data is transferred. This means that in multi-master system each master must monitor the bus for collisions and act accordingly.



Arbitration procedure of two masters

Figure 6.26-16 Arbitration Lost

- When I2C_STATUS0 = 0x38, an “Arbitration Lost” is received. Arbitration lost event maybe occur during the send START bit, data bits or STOP bit. User could set (STA, STO, SI, AA) = (1, 0, 1, X) to send START again when bus free, or set (STA, STO, SI, AA) = (0, 0, 1, X) to not addressed Slave mode. User can detect bus free by ONBUSY (I2C_STATUS1 [8]).
- When I2C_STATUS0 = 0x00, a “Bus Error” is received. To recover I^{2}C bus from a bus error, STO should be set and SI should be cleared, and then STO is cleared to release bus.
  - Set (STA, STO, SI, AA) = (0, 1, 1, X) to stop current transfer

  Set (STA, STO, SI, AA) = (0, 0, 1, X) to release bus

6.26.5.3 PDMA Transfer Function

The I^{2}C controller supports PDMA transfer function. When TXPDMAEN (I2C_CTL1 [0]) is set to 1, the I^{2}C controller will issue request to PDMA controller to start the DMA transmission process automatically.

When RXPDMAEN (I2C_CTL1 [1]) is set to 1, the I^{2}C controller will start the receive PDMA process. The I^{2}C controller will issue the request to PDMA controller automatically when there is data written into the received BUFFER.

When I^{2}C enters PDMA mode, the mostly status interrupt will be masked. Let the interrupt not occur besides the bus error or NACK or STOP interrupt (0x20, 0x30, 0x38, 0x48, 0x58, 0x00, 0xA0, 0xC0, 0x88 and 0x98).

Set the PDMASTR (I2C_CTL1 [8]) only the I^{2}C controller in master TX mode. If PDMASTR is cleared to 0, I^{2}C will send STOP automatically after PDMA transfer done and buffer empty. If PDMASTR is set to

1, SI will be set to 1 and I²C bus will be stretched by hardware after PDMA transfer done and buffer empty.

6.26.5.4 Programmable setup and hold times

To guarantee a correct data setup and hold time, the timing must be configured. By programming HTCTL (I2C_TMCTL[24:16]) to configure hold time and STCTL (I2C_TMCTL[8:0]) to configure setup time.

The delay timing refer peripheral clock (PCLK). When device stretch master clock, the setup and hold time configuration value will not affected by stretched.

User should focus the limitation of setup and hold time configuration, the timing setting must follow I²C protocol. Once setup time configuration greater than design limitation, that means if setup time setting make SCL output less than three PCLKs, the I²C controller can't work normally due to SCL must sample three times. Once the hold time configuration is greater than I²C clock limitation, I²C will occur bus error. It is suggested that user calculate suitable timing with baud rate and protocol before setting timing. Table 6.26-1 shows the relationship between I²C baud rate and PCLK, the number of table represent one clock duty contain how many PCLKs. Setup and hold time configuration even can program some extreme values in the design, but user should follow I²C protocol standard.

I²C Baud Rate PCLK	100k	200k	400k	800k	1200k
12 MHz	120	60	30	15	10
24 MHz	240	120	60	30	20
48 MHz	480	240	120	60	40
72 MHz	720	360	180	90	60

Table 6.26-1 Relationship between I²C Baud Rate and PCLK

For setup time wrong adjustment example, assuming one SCL cycle contains 5 PCLKs and set STCTL (I2C_TMCTL[8:0]) to 3 that stretch three PCLKs for setup time setting. The setup time maximum setting value: ST_{limit} = (I2C_CLKDIV[7:0]+1) X 2 - 6.

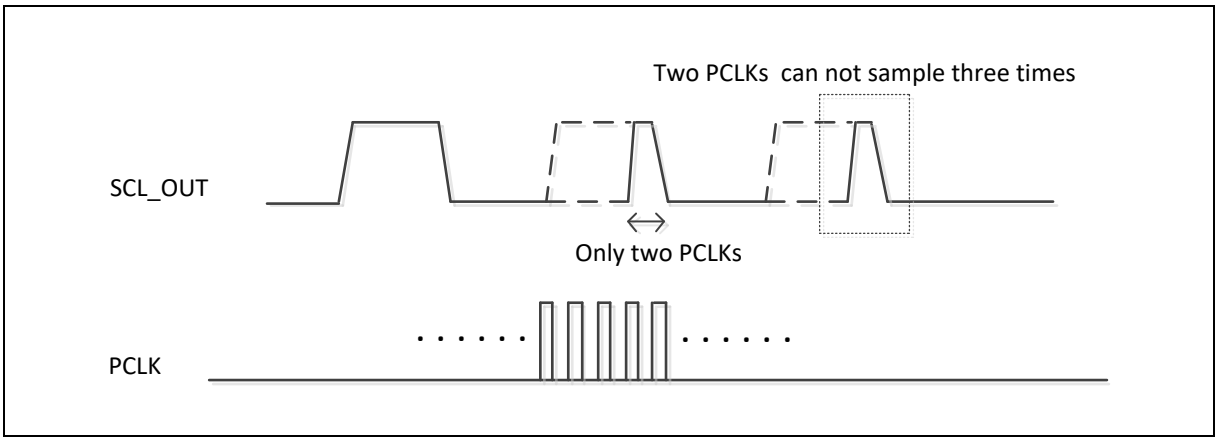


Figure 6.26-17 Setup Time Wrong Adjustment

For hold time wrong adjustment example, use I²C Baud Rate = 1200k and PCLK = 72 MHz, the SCL high/low duty = 60 PCLK. When HTCTL (I2C_TMCTL[24:16]) is set to 61 and STCTL (I2C_TMCTL[8:0]) is set to 0, then SDA output delay will over SCL high duty and cause bus error. The hold time maximum setting value: HT_{limit} = (I2C_CLKDIV[7:0]+1) X 2 - 9.

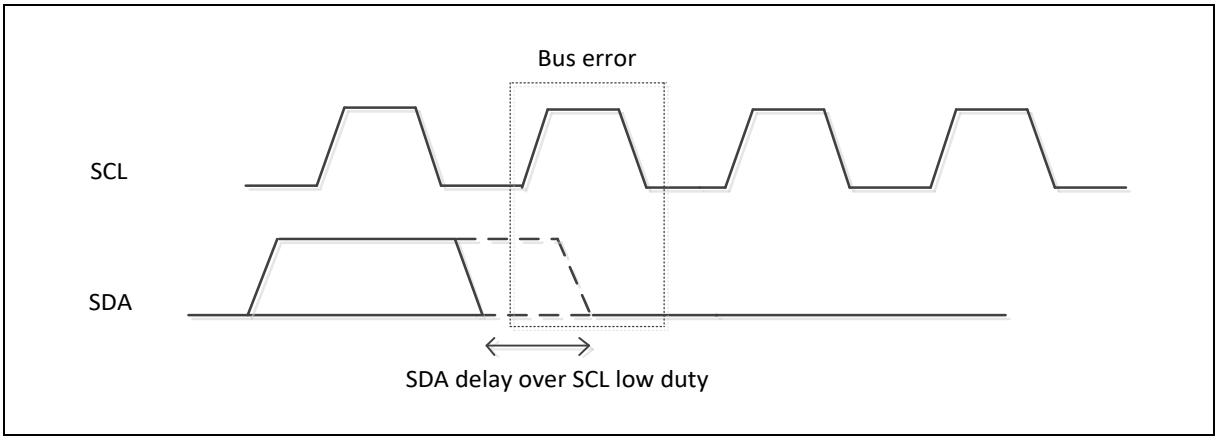


Figure 6.26-18 Hold Time Wrong Adjustment

6.26.5.5 I²C Protocol Registers

To control I²C port through the following fifteen special function registers: I2C_CTL0 (control register), I2C_STATUS0 (status register), I2C_DAT (data register), I2C_ADDRn (address registers, n=0~3), I2C_ADDRMSKn (address mask registers, n=0~3), I2C_CLKDIV (clock rate register), I2C_TOCTL (Time-out control register), I2C_WKCTL(wake up control register) and I2C_WKSTS(wake up status register).

Address Registers (I2C_ADDR)

The I²C port is equipped with four slave address registers, I2C_ADDRn (n=0~3). The contents of the register are irrelevant when I²C is in Master mode. In Slave mode, the bit field ADDR(I2C_ADDRn[7:1]) must be loaded with the chip's own slave address. The I²C hardware will react if the contents of I2C_ADDRn are matched with the received slave address.

The I²C ports support the "General Call" function. If the GC bit (I2C_ADDRn [0]) is set the I²C port hardware will respond to General Call address (00H). Clear GC bit to disable general call function.

When the GC bit is set and the I²C is in Slave mode, it can receive the general call address by 00H after Master send general call address to I²C bus, then it will follow status of GC mode.

Slave Address Mask Registers (I2C_ADDRMSK)

The I²C bus controller supports multiple address recognition with four address mask registers I2C_ADDRMSKn (n=0~3). When the bit in the address mask register is set to 1, it means the received corresponding address bit is "Don't care". If the bit is set to 0, it means the received corresponding register bit should be exactly the same as address register.

Data Register (I2C_DAT)

This register contains a byte of serial data to be transmitted or a byte which just has been received. The CPU can be read from or written to the 8-bit (I2C_DAT [7:0]) directly while it is not in the process of shifting a byte. When I²C is in a defined state and the serial interrupt flag (SI) is set, data in I2C_DAT [7:0] remains stable. While data is being shifted out, data on the bus is simultaneously being shifted in; I2C_DAT [7:0] always contains the last data byte presented on the bus.

The acknowledge bit is controlled by the I²C hardware and cannot be accessed by the CPU. Serial data is shifted into I2C_DAT [7:0] on the rising edges of serial clock pulses on the SCL line. When a byte has been shifted into I2C_DAT [7:0], the serial data is available in I2C_DAT [7:0], and the acknowledge bit (ACK or NACK) is returned by the control logic during the ninth clock pulse. In order to monitor bus status while sending data, the bus data will be shifted to I2C_DAT[7:0] when sending I2C_DAT[7:0] to bus. In the case of sending data, serial data bits are shifted out from I2C_DAT [7:0] on the falling edge of SCL clocks, and is shifted to I2C_DAT [7:0] on the rising edge of SCL clocks. Figure 6.26-19 shows I²C Data Shifting Direction.

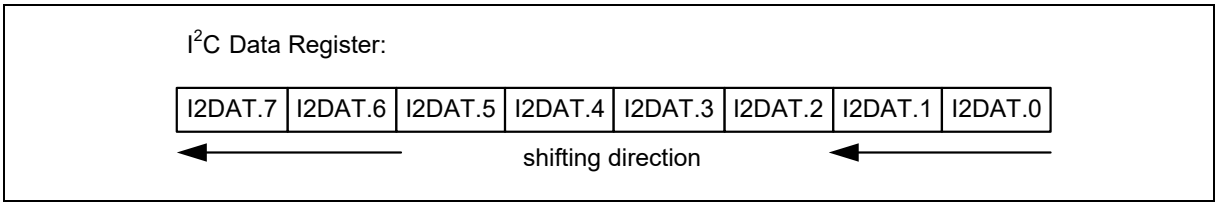


Figure 6.26-19 I²C Data Shifting Direction

Control Register (I2C_CTL0)

The CPU can be read from and written to I2C_CTL0 [7:0] directly. When the I²C port is enabled by setting I2CEN (I2C_CTL0 [6]) to high, the internal states will be controlled by I2C_CTL0 and I²C logic hardware.

There are two bits are affected by hardware: the SI bit is set when the I²C hardware requests a serial interrupt, and the STO bit is cleared when a STOP condition is present on the bus. The STO bit is also cleared when I2CEN = 0.

Once a new status code is generated and stored in I2C_STATUS0, the I²C Interrupt Flag bit SI (I2C_CTL0 [3]) will be set automatically. If the Enable Interrupt bit INTEN (I2C_CTL0 [7]) is set at this time, the I²C interrupt will be generated. The bit field I2C_STATUS0[7:0] stores the internal state code, the content keeps stable until SI is cleared by software.

Status Register (I2C_STATUS0)

I2C_STATUS0 [7:0] is an 8-bit read-only register. The bit field I2C_STATUS0 [7:0] contains the status code and there are 26 possible status codes. All states are listed in Table 6.26-2. When I2C_STATUS0 [7:0] is F8H, no serial interrupt is requested. All other I2C_STATUS0 [7:0] values correspond to the defined I²C states. When each of these states is entered, a status interrupt is requested (SI = 1). A valid status code is present in I2C_STATUS0[7:0] one cycle PCLK after SI set by hardware and is still present one cycle PCLK after SI reset by software.

In addition, the state 00H stands for a Bus Error, which occurs when a START or STOP condition is present at an incorrect position in the I²C format frame. A Bus Error may occur during the serial transfer of an address byte, a data byte or an acknowledge bit. To recover I²C from bus error, STO should be set and SI should be cleared to enter Not Addressed Slave mode. Then STO is cleared to release bus and to wait for a new communication. The I²C bus cannot recognize stop condition during this action when a bus error occurs.

Master Mode		Slave Mode	
STATUS	Description	STATUS	Description
0x08[1]	Start	0xA0	Slave Transmit Repeat Start or Stop
0x10[1]	Master Repeat Start	0xA8[1]	Slave Transmit Address ACK
0x18[1]	Master Transmit Address ACK	0xB8[1]	Slave Transmit Data ACK
0x20	Master Transmit Address NACK	0xC0	Slave Transmit Data NACK
0x28[1]	Master Transmit Data ACK	0xC8[1]	Slave Transmit Last Data ACK
0x30	Master Transmit Data NACK	0x60[1]	Slave Receive Address ACK
0x38	Master Arbitration Lost	0x68[1]	Slave Receive Arbitration Lost
0x40[1]	Master Receive Address ACK	0x80[1]	Slave Receive Data ACK
0x48	Master Receive Address NACK	0x88	Slave Receive Data NACK
0x50[1]	Master Receive Data ACK	0x70[1]	GC mode Address ACK

0x58	Master Receive Data NACK	0x78[1]	GC mode Arbitration Lost
0x00	Bus error	0x90[1]	GC mode Data ACK
		0x98	GC mode Data NACK
		0xB0[1]	Address Transmit Arbitration Lost
0xF0	If the BMDEN =1 and the ACKMEN bit is enabled, the information of I2C_STATUS0 will be fixed as 0xF0 in slave receive condition.		
0xF8	Bus Released **Note:** Status "0xF8" exists in both master/slave modes, and it won't raise interrupt. Note [1]: No interrupt in PDMA mode		

Table 6.26-2 I²C Status Code Description

Clock Baud Rate Bits (I2C_CLKDIV)

The data baud rate of I²C is determined by DIVIDER(I2C_CLKDIV [7:0]) register when I²C is in Master mode, and it is not necessary in a Slave mode. In the Slave mode, I²C will automatically synchronize it with any clock frequency from master I²C device. In the slave mode, system clock frequency should be greater than I²C bus maximum clock 20 times.

The data baud rate of I²C setting is Data Baud Rate of I²C = (system clock) / (4x (I2C_CLKDIV [7:0] +1)). If system clock = 16 MHz, the I2C_CLKDIV [7:0] = 40 (28H), the data baud rate of I²C = 16 MHz/ (4x (40 +1)) = 97.5 Kbits/sec.

Time-out Control Register (I2C_TOCTL)

There is a 14-bit time-out counter which can be used to deal with the I²C bus hang-up. If the time-out counter is enabled, the counter starts up counting until it overflows (TOIF=1) and generates I²C interrupt to CPU or stops counting by clearing TOCEN to 0. When time-out counter is enabled, writing 1 to the SI flag will reset counter and re-start up counting after SI is cleared. If I²C bus hangs up, it causes the I2C_STATUS0 and flag SI are not updated for a period, the 14-bit time-out counter may overflow and acknowledge CPU the I²C interrupt. Refer to Figure 6.26-20 for the 14-bit time-out counter. User may write 1 to clear TOIF to 0.

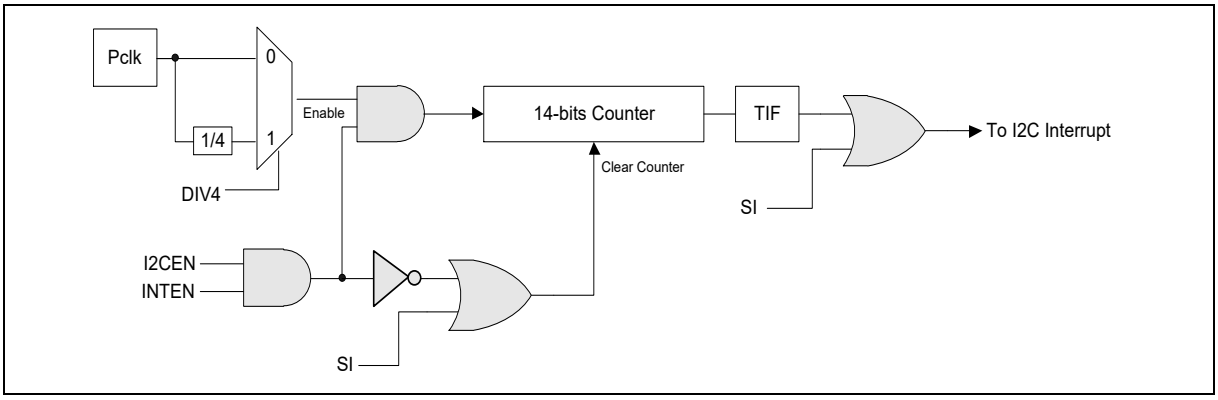


Figure 6.26-20 I²C Time-out Count Block Diagram

Wake-up Control Register (I2C_WKCTL)

When chip enters Power-down mode and sets WKEN (I2C_WKCTL [0]) to 1, other I²C master can wake up the chip by addressing the I²C device, user must configure the related setting before entering sleep mode. The ACK bit cycle of address match frame is done in power-down. The controller will stretch the SCL to low when the address is matched the device's address and the ACK cycle done, then the I²C controller will go ahead. If NHDBUSEN (I2C_WKCTL [7]) is set, the controller will don't stretch the SCL to low. Note that when the controller doesn't stretch the SCL to low, transmit or receive data will perform

immediately. If data transmitted or received when SI event is not clear, user must reset the I²C controller and execute the original operation again.

Wake-up Status Register (I2C_WKSTS)

When system is woken up by other I²C master device, WKIF is set to indicate this event. User needs write "1" to clear this bit.

When the chip is woken-up by address match with one of the device address register (I2C_ADDRn), the user shall check the WKAKDONE (I2C_WKSTS [1]) bit is set to 1 to confirm the address byte has done. The WKAKDONE bit indicates that the ACK bit cycle of address byte is done in power-down. The controller will stretch the SCL to low when the address is matched the device's slave address and the ACK cycle done. The SCL is stretched until WKAKDONE is clear by user. If the frequency of SCL is low speed and the system has wakeup from address match frame, the user shall check WKAKDONE to confirm this frame has transaction done and then to do the wakeup procedure. Note that user can't release WKIF through clearing the WKAKDONE bit to 0.

The WRSTSWK (I2C_WKSTS [2]) bit records the Read/Write command before the I²C controller sends address. The user can read this bit's status to prepare the next transmitted data (WRSTSWK = 0) or to wait the incoming data (WRSTSWK = 1) can be stored in time after the system is woken up by the address match frame. Note that the WRSTSWK (I2C_WKSTS [2]) bit is cleared when writing one to the WKAKDONE (I2C_WKSTS [1]) bit.

When system is woken up by other I²C master device, WKIF is set to indicate this event. User needs to write "1" to clear this bit.

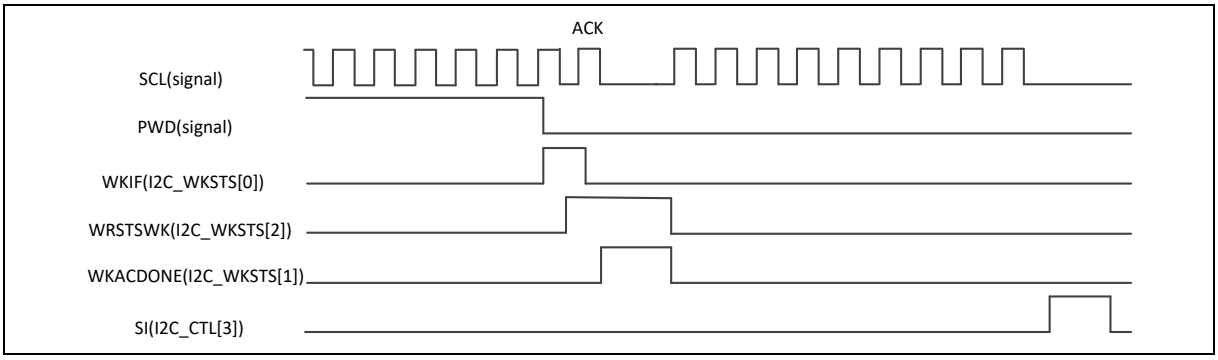


Figure 6.26-21 I²C Wake-Up Related Signals Waveform

I²C Control Register 1 (I2C_CTL1)

If enable 10-bit addressing mode ADDR10EN (I2C_CTL1 [9]) is set, the I²C will run in 10-bit mode.

For PDMA function, set TXPDMAEN (I2C_CTL1 [0]) and RXPDMAEN (I2C_CTL1 [1]) can be set to operate. And set PDMARST (I2C_CTL1 [2]) to reset the PDMA control logic.

I²C Status Register 1 (I2C_STATUS1)

The I²C controller supports four slave address flag registers, ADMAT0, ADMAT1, ADMAT2 and ADMAT3 (I2C_STATUS1[3:0]). Every control register represent which address is used and set 1 to inform software.

I²C Timing Configure Control Register (I2C_TMCTL)

In order to configure setup/hold time, the HTCTL (I2C_TMCTL[24:16]) and STCTL (I2C_TMCTL[8:0]) are set based on actual demand.

6.26.5.6 Example for Random Read on EEPROM

The following steps are used to configure the I2C0 related registers when using I²C to read data from EEPROM.

1. Set I2C0 the multi-function pin as SCL and SDA pins. The muti-function configuration reference Basic Configuration.
2. Enable I2C0 APB clock. The clock configuration reference Basic Configuration.
3. Set I2C0RST=1 to reset I2C0 controller then set I2C0 controller to normal operation. The reset controller configuration reference Basic Configuration.
4. Set I2CEN=1 to enable I2C0 controller in the “I2C_CTL0” register.
5. Give I2C0 clock a divided register value for I^{2}C clock rate in the “I2C_CLKDIV”.
6. Enable system I2C0 IRQ in system “NVIC” control register.
7. Set INTEN=1 to enable I2C0 Interrupt in the “I2C_CTL0” register.
8. Set I2C0 address registers “I2C_ADDR0 ~ I2C_ADDR3”.

Random read operation is one of the methods of access EEPROM. The method allows the master to access any address of EEPROM space. Figure 6.26-22 shows the EEPROM random read operation.

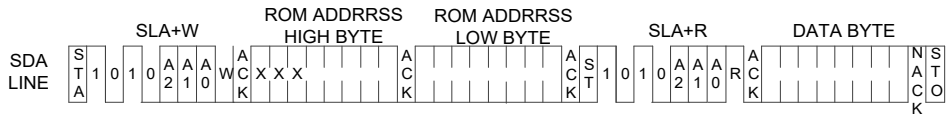


Figure 6.26-22 EEPROM Random Read

Figure 6.26-23 shows how to use the I^{2}C controller to implement the protocol of EEPROM random read.

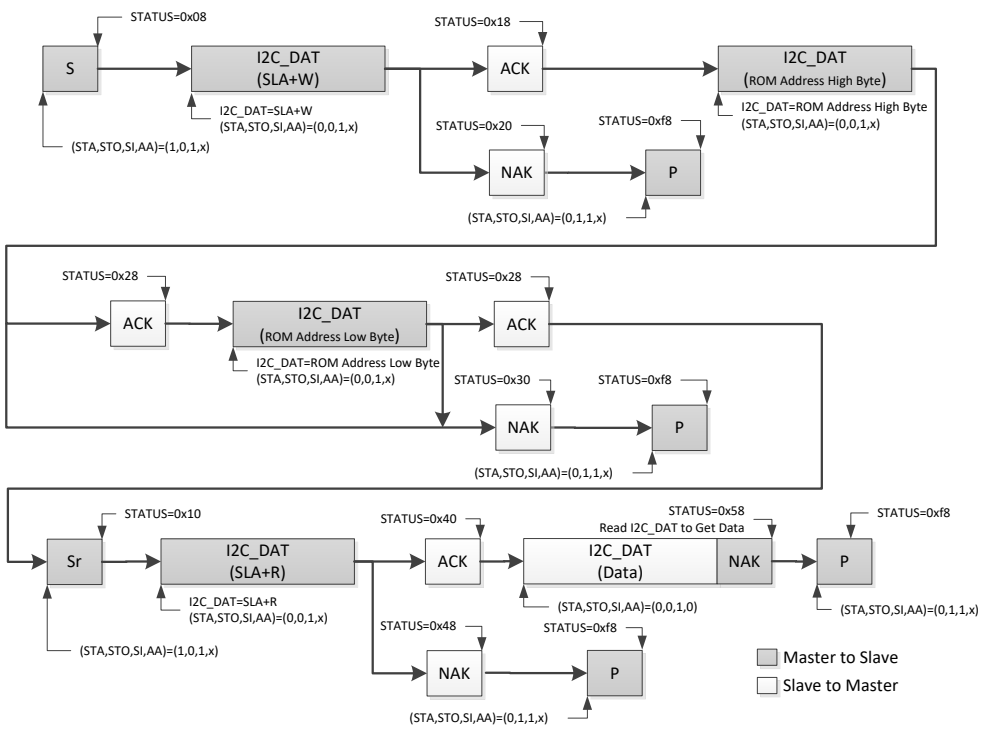


Figure 6.26-23 Protocol of EEPROM Random Read

The I^{2}C controller, which is a master, sends START to bus. Then, it sends a SLA+W (Slave address + Write bit) to EERPOM followed by two bytes data address to set the EEPROM address to read. Finally,

a Repeat START followed by SLA+R is sent to read the data from EEPROM.

6.26.6 Register Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
I²C Base Address: **I2Cn_BA = 0x4080_0000 + (0x10000 *n)** **n= 0,1,2,3,4,5**				
I2C_CTL0	I2Cn_BA+0x00	R/W	I²C Control Register 0	0x0000_0000
I2C_ADDR0	I2Cn_BA+0x04	R/W	I²C Slave Address Register0	0x0000_0000
I2C_DAT	I2Cn_BA+0x08	R/W	I²C Data Register	0x0000_0000
I2C_STATUS0	I2Cn_BA+0x0C	R	I²C Status Register 0	0x0000_00F8
I2C_CLKDIV	I2Cn_BA+0x10	R/W	I²C Clock Divided Register	0x0000_0000
I2C_TOCTL	I2Cn_BA+0x14	R/W	I²C Time-out Control Register	0x0000_0000
I2C_ADDR1	I2Cn_BA+0x18	R/W	I²C Slave Address Register1	0x0000_0000
I2C_ADDR2	I2Cn_BA+0x1C	R/W	I²C Slave Address Register2	0x0000_0000
I2C_ADDR3	I2Cn_BA+0x20	R/W	I²C Slave Address Register3	0x0000_0000
I2C_ADDRMSK0	I2Cn_BA+0x24	R/W	I²C Slave Address Mask Register0	0x0000_0000
I2C_ADDRMSK1	I2Cn_BA+0x28	R/W	I²C Slave Address Mask Register1	0x0000_0000
I2C_ADDRMSK2	I2Cn_BA+0x2C	R/W	I²C Slave Address Mask Register2	0x0000_0000
I2C_ADDRMSK3	I2Cn_BA+0x30	R/W	I²C Slave Address Mask Register3	0x0000_0000
I2C_WKCTL	I2Cn_BA+0x3C	R/W	I²C Wake-up Control Register	0x0000_0000
I2C_WKSTS	I2Cn_BA+0x40	R/W	I²C Wake-up Status Register	0x0000_0000
I2C_CTL1	I2Cn_BA+0x44	R/W	I²C Control Register 1	0x0000_0000
I2C_STATUS1	I2Cn_BA+0x48	R/W	I²C Status Register 1	0x0000_0000
I2C_TMCTL	I2Cn_BA+0x4C	R/W	I²C Timing Configure Control Register	0x0000_0000

6.26.7 Register Description

I²C Control Register (I2C_CTL0)

Register	Offset	R/W	Description	Reset Value
I2C_CTL0	I2Cn_BA+0x00	R/W	I²C Control Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INTEN	**I2CEN**	**STA**	**STO**	**SI**	**AA**	Reserved	

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**INTEN**	**Enable Interrupt** 0 = I²C interrupt Disabled. 1 = I²C interrupt Enabled.
[6]	**I2CEN**	**I²C Controller Enable Bit** Set to enable I²C serial function controller. When I2CEN=1 the I²C serial function enable. The multi-function pin function must set to SDA, and SCL of I²C function first. 0 = I²C controller Disabled. 1 = I²C controller Enabled.
[5]	**STA**	**I²C START Control** Setting STA to logic 1 to enter Master mode, the I²C hardware sends a START or Repeat START condition to bus when the bus is free.
[4]	**STO**	**I²C STOP Control** In Master mode, setting STO to transmit a STOP condition to bus then I²C controller will check the bus condition if a STOP condition is detected. This bit will be cleared by hardware automatically.
[3]	**SI**	**I²C Interrupt Flag** When a new I²C state is present in the I2C_STATUS0 register, the SI flag is set by hardware. If bit INTEN (I2C_CTL0 [7]) is set, the I²C interrupt is requested. SI must be cleared by software. Clear SI by writing 1 to this bit. For ACKMEN is set in slave read mode, the SI flag is set in 8th clock period for user to confirm the acknowledge bit and 9th clock period for user to read the data in the data buffer.
[2]	**AA**	**Assert Acknowledge Control** When AA =1 prior to address or data is received, an acknowledged (low level to SDA) will be returned during the acknowledge clock pulse on the SCL line when (1) A slave is acknowledging the address sent from master, (2) The receiver devices are acknowledging the data sent by transmitter. When AA=0 prior to address or data received, a Not acknowledged (high level to SDA) will be returned during the acknowledge clock pulse on the SCL line.
[1:0]	**Reserved**	Reserved.

I²C Data Register (I2C_DAT)

Register	Offset	R/W	Description	Reset Value
I2C_DAT	I2Cn_BA+0x08	R/W	I²C Data Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DAT							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**DAT**	**I²C Data** Bit [7:0] is located with the 8-bit transferred/received data of I²C serial port.

I²C Status Register (I2C_STATUS0)

Register	Offset	R/W	Description	Reset Value
I2C_STATUS0	I2Cn_BA+0x0C	R	I²C Status Register 0	0x0000_00F8

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
STATUS							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**STATUS**	**I²C Status** The three least significant bits are always 0. The five most significant bits contain the status code. There are 28 possible status codes. When the content of I2C_STATUS0 is F8H, no serial interrupt is requested. Others I2C_STATUS0 values correspond to defined I²C states. When each of these states is entered, a status interrupt is requested (SI = 1). A valid status code is present in I2C_STATUS0 one cycle after SI is set by hardware and is still present one cycle after SI has been reset by software. In addition, states 00H stands for a Bus Error. A Bus Error occurs when a START or STOP condition is present at an illegal position in the formation frame. Example of illegal position are during the serial transfer of an address byte, a data byte or an acknowledge bit.

I²C Clock Divided Register (I2C_CLKDIV)

Register	Offset	R/W	Description	Reset Value
I2C_CLKDIV	I2Cn_BA+0x10	R/W	I²C Clock Divided Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
NFCNT				Reserved		DIVIDER	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DIVIDER							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:12]	**NFCNT**	**Noise Filter Count** The bits control the input filter width. 0 = Filter width 3*PCLK. 1 = Filter width 4*PCLK. *N* = Filter width (3+*N*)*PCKL. **Note:** Filter width Min :3*PCLK, Max : 18*PCLK
[11:10]	**Reserved**	Reserved.
[9:0]	**DIVIDER**	**I²C Clock Divided** Indicates the I²C clock rate: Data Baud Rate of I²C = (system clock) / (4x (I2C_CLKDIV+1)). **Note:** The minimum value of I2C_CLKDIV is 4.

I²C Time-out Control Register (I2C_TOCTL)

Register	Offset	R/W	Description	Reset Value
I2C_TOCTL	I2Cn_BA+0x14	R/W	I²C Time-out Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**TOCEN**	**TOCDIV4**	**TOIF**

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**TOCEN**	**Time-out Counter Enable Bit** When enabled, the 14-bit time-out counter will start counting when SI is cleared. Setting flag SI to '1' will reset counter and re-start up counting after SI is cleared. 0 = Time-out counter Disabled. 1 = Time-out counter Enabled.
[1]	**TOCDIV4**	**Time-out Counter Input Clock Divided by 4** When enabled, the time-out period is extended 4 times. 0 = Time-out period is extend 4 times Disabled. 1 = Time-out period is extend 4 times Enabled.
[0]	**TOIF**	**Time-out Flag** This bit is set by hardware when I²C time-out happened and it can interrupt CPU if I²C interrupt enable bit (INTEN) is set to 1. **Note:** Software can write 1 to clear this bit.

I²C Slave Address Register (ADDRx)

Register	Offset	R/W	Description	Reset Value
I2C_ADDR0	I2Cn_BA+0x04	R/W	I²C Slave Address Register0	0x0000_0000
I2C_ADDR1	I2Cn_BA+0x18	R/W	I²C Slave Address Register1	0x0000_0000
I2C_ADDR2	I2Cn_BA+0x1C	R/W	I²C Slave Address Register2	0x0000_0000
I2C_ADDR3	I2Cn_BA+0x20	R/W	I²C Slave Address Register3	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					ADDR		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADDR							GC

Bits	Description	
[31:11]	**Reserved**	Reserved.
[10:1]	**ADDR**	**I²C Address** The content of this register is irrelevant when I²C is in Master mode. In the slave mode, the seven most significant bits must be loaded with the chip's own address. The I²C hardware will react if either of the address is matched. **Note:** When software set 10'h000, the address cannot be used.
[0]	**GC**	**General Call Function** 0 = General Call Function Disabled. 1 = General Call Function Enabled.

I²C Slave Address Mask Register (ADDRMSKx)

Register	Offset	R/W	Description	Reset Value
I2C_ADDRMSK0	I2Cn_BA+0x24	R/W	I²C Slave Address Mask Register0	0x0000_0000
I2C_ADDRMSK1	I2Cn_BA+0x28	R/W	I²C Slave Address Mask Register1	0x0000_0000
I2C_ADDRMSK2	I2Cn_BA+0x2C	R/W	I²C Slave Address Mask Register2	0x0000_0000
I2C_ADDRMSK3	I2Cn_BA+0x30	R/W	I²C Slave Address Mask Register3	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					ADDRMSK		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADDRMSK							Reserved

Bits	Description	
[31:11]	**Reserved**	Reserved.
[10:1]	**ADDRMSK**	**I²C Address Mask** 0 = Mask Disabled (the received corresponding register bit should be exact the same as address register.). 1 = Mask Enabled (the received corresponding address bit is don't care.). I²C bus controllers support multiple address recognition with four address mask register. When the bit in the address mask register is set to one, it means the received corresponding address bit is don't-care. If the bit is set to 0, that means the received corresponding register bit should be exact the same as address register. **Note:** The wake-up function cannot use address mask.
[0]	**Reserved**	Reserved.

I²C Wake-up Control Register (I2C_WKCTL)

Register	Offset	R/W	Description	Reset Value
I2C_WKCTL	I2Cn_BA+0x3C	R/W	I²C Wake-up Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NHDBUSEN	**Reserved**						**WKEN**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**NHDBUSEN**	**I²C No Hold BUS Enable Bit** 0 = I²C hold bus after wake-up. 1= I²C don't hold bus after wake-up. **Note:** The I²C controller could respond when WKIF event is not clear, it may cause error data transmitted or received. If data transmitted or received when WKIF event is not clear, user must reset I²C controller and execute the original operation again.
[6:1]	**Reserved**	Reserved.
[0]	**WKEN**	**I²C Wake-up Enable Bit** 0 = I²C wake-up function Disabled. 1= I²C wake-up function Enabled.

I²C Wake-up Status Register (I2C_WKSTS)

Register	Offset	R/W	Description	Reset Value
I2C_WKSTS	I2Cn_BA+0x40	R/W	I²C Wake-up Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**WRSTSWK**	**WKAKDONE**	**WKIF**

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**WRSTSWK**	**Read/Write Status Bit in Address Wakeup Frame** 0 = Write command be record on the address match wakeup frame. 1 = Read command be record on the address match wakeup frame. **Note:** This bit will be cleared when software can write 1 to WKAKDONE bit.
[1]	**WKAKDONE**	**Wakeup Address Frame Acknowledge Bit Done** 0 = The ACK bit cycle of address match frame isn't done. 1 = The ACK bit cycle of address match frame is done in power down. **Note:** This bit can't release WKIF. Software can write 1 to clear this bit.
[0]	**WKIF**	**I²C Wake-up Flag** When chip is woken up from Power-down mode by I²C, this bit is set to 1. Software can write 1 to clear this bit.

I²C Control Register 1 (I2C_CTL1)

Register	Offset	R/W	Description	Reset Value
I2C_CTL1	I2Cn_BA+0x44	R/W	I²C Control Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						ADDR10EN	PDMASTR
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					PDMARST	RXPDMAEN	TXPDMAEN

Bits	Description	
[31:10]	**Reserved**	Reserved.
[9]	**ADDR10EN**	**Address 10-bit Function Enable Bit** 0 = Address match 10-bit function Disabled. 1 = Address match 10-bit function Enabled.
[8]	**PDMASTR**	**PDMA Stretch Bit** 0 = I²C send STOP automatically after PDMA transfer done. (only master TX) 1 = I²C SCL bus is stretched by hardware after PDMA transfer done if the SI is not cleared. (only master TX)
[7:3]	**Reserved**	Reserved.
[2]	**PDMARST**	**PDMA Reset** 0 = No effect. 1 = Reset the I²C request to PDMA.
[1]	**RXPDMAEN**	**PDMA Receive Channel Available** 0 = Receive PDMA function Disabled. 1 = Receive PDMA function Enabled.
[0]	**TXPDMAEN**	**PDMA Transmit Channel Available** 0 = Transmit PDMA function Disabled. 1 = Transmit PDMA function Enabled.

I²C Status Register 1 (I2C_STATUS1)

Register	Offset	R/W	Description	Reset Value
I2C_STATUS1	I2Cn_BA+0x48	R/W	I²C Status Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							**RXPDMAERR**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**ADMAT3**	**ADMAT2**	**ADMAT1**	**ADMAT0**

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**RXPDMAERR**	**RX PDMA ERROR Status** When RX PDMA is busy and received fail data. This bit will set to 1, and software can write 1 to clear this bit. 0 = PDMA receive data correct. 1 = PDMA receive data failed. **Note:** This bit is vaild when RXPDMAEN (I2C_CTL1[1]) is enabled.
[15:4]	**Reserved**	Reserved.
[3]	**ADMAT3**	**I²C Address 3 Match Status** When address 3 is matched, hardware will inform which address used. This bit will set to 1, and software can write 1 to clear this bit.
[2]	**ADMAT2**	**I²C Address 2 Match Status** When address 2 is matched, hardware will inform which address used. This bit will set to 1, and software can write 1 to clear this bit.
[1]	**ADMAT1**	**I²C Address 1 Match Status** When address 1 is matched, hardware will inform which address used. This bit will set to 1, and software can write 1 to clear this bit.
[0]	**ADMAT0**	**I²C Address 0 Match Status** When address 0 is matched, hardware will inform which address used. This bit will set to 1, and software can write 1 to clear this bit.

I²C Timing Configure Control Register (I2C_TMCTL)

Register	Offset	R/W	Description	Reset Value
I2C_TMCTL	I2Cn_BA+0x4C	R/W	I²C Timing Configure Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							HTCTL
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
HTCTL							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							STCTL
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
STCTL							

Bits	Description	
[31:25]	**Reserved**	Reserved.
[24:16]	**HTCTL**	**Hold Time Configure Control** This field is used to generate the delay timing between SCL falling edge and SDA rising edge in transmission mode. The delay hold time is numbers of peripheral clock = HTCTL x PCLK.
[15:9]	**Reserved**	Reserved.
[8:0]	**STCTL**	**Setup Time Configure Control** This field is used to generate a delay timing between SDA falling edge and SCL rising edge in transmission mode. The delay setup time is numbers of peripheral clock = STCTL x PCLK. Note: Setup time setting should not make SCL output less than three PCLKs.

6.27 Serial Peripheral Interface (SPI)

6.27.1 Overview

The Serial Peripheral Interface (SPI) applies to synchronous serial data communication and allows full duplex transfer. Devices communicate in Master/Slave mode with the 4-wire bi-direction interface. The chip contains up to four sets of SPI controllers performing a serial-to-parallel conversion on data received from a peripheral device, and a parallel-to-serial conversion on data transmitted to a peripheral device. Each SPI controller can be configured as a master or a slave device and supports the PDMA function to access the data buffer. Each SPI controller also supports I^2S mode to connect external audio CODEC.

6.27.2 Features

- SPI Mode
  - Up to four sets of SPI controllers
  - Supports Master or Slave mode operation
  - Configurable bit length of a transaction word from 8 to 32-bit
  - For SPI PDMA function disable, provides separate 8-level of 32-bit (or 16-level of 16-bit) transmit and receive FIFO buffers which depended on SPI setting of data width
  - For SPI PDMA function enable, provides separate 8-level of 32-bit (or 16-level of 16-bit or 32-level of 8-bit) transmit and receive FIFO buffers which depended on SPI setting of data width
  - Supports MSB first or LSB first transfer sequence
  - Supports Byte Reorder function
  - Supports Byte or Word Suspend mode
  - Supports PDMA transfer
  - Supports 3-Wire, no slave selection signal, bi-direction interface
  - Supports one data channel half-duplex transfer
  - Supports receive-only mode
- I^2S Mode
  - Supports Master or Slave
  - Capable of handling 8-, 16-, 24- and 32-bit word sizes
  - Each provides two 8-level FIFO data buffers, one for transmitting and the other for receiving
  - Supports monaural and stereo audio data
  - Supports PCM mode A, PCM mode B, I^2S and MSB justified data format
  - Supports two PDMA requests, one for transmitting and the other for receiving

6.27.3 Block Diagram

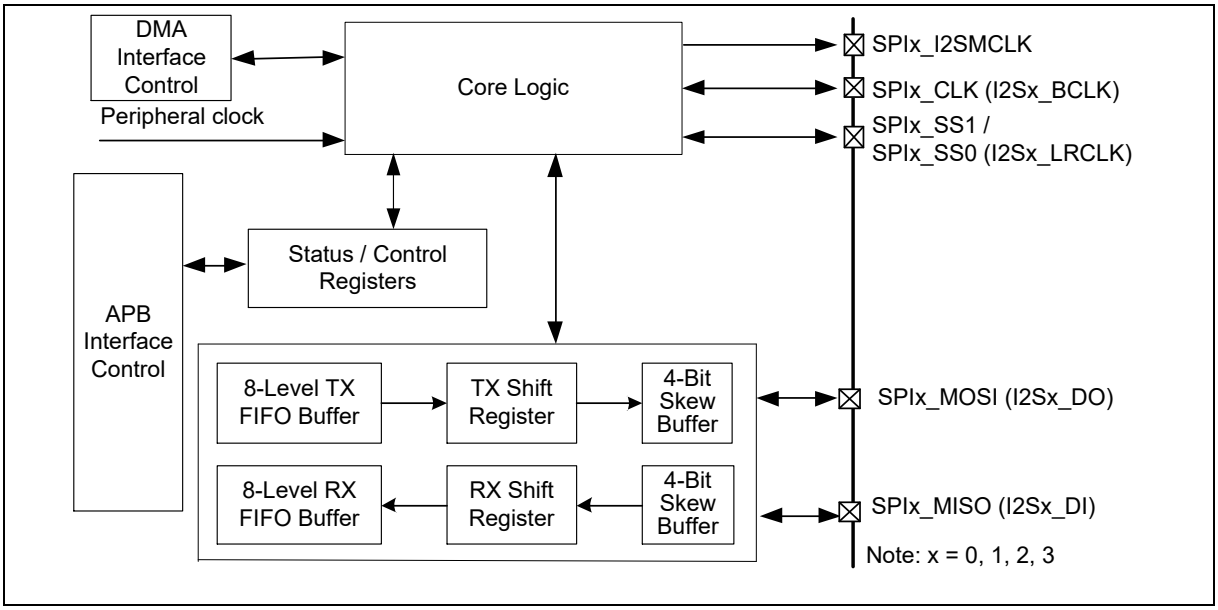


Figure 6.27-1 SPI Block Diagram

TX FIFO Buffer:

The transmit FIFO buffer is a 8-level depth, 32-bit wide, first-in, first-out register buffer. The data can be written to the transmit FIFO buffer in advance through software by writing the SPIx_TX register. In SPI mode, the transmit FIFO will be configured as 16-level while data length is set as 8~16 bits.

RX FIFO Buffer:

The receive FIFO buffer is also a 8-level depth, 32-bit wide, first-in, first-out register buffer. The receive control logic will store the receive data to this buffer. The FIFO buffer data can be read from SPIx_RX register by software. In SPI mode, the receive FIFO will be configured as 16-level while data length is set as 8~16 bits.

TX Shift Register:

The transmit shift register is a 32-bit wide register buffer. The transmit data is loaded from the TX FIFO buffer and shifted out bit-by-bit to the skew buffer.

RX Shift Register:

The receive shift register is also a 32-bit wide register buffer. The receive data is shift in bit-by-bit from the skew buffer and is loaded into RX FIFO buffer when a transaction is done.

Skew Buffer:

The skew buffer is a 4-level 1-bit buffer. There are two skew buffers in transmitting and received side. In received side, it is used to shift bits into RX shift register from SPI bus. In transmitting side, it is used to shift bits into SPI bus from TX shift register.

6.27.4 Basic Configuration

6.27.4.1 SPI0 Basic Configuration

- Clock Source Configuration
  - Select the source of SPI0 peripheral clock on SPI0SEL (CLK_CLKSEL4[1:0]).
  - Enable SPI0 peripheral clock in SPI0CKEN (CLK_APBCLK2[4]).
- Reset Configuration

- Reset SPI0 controller in SPI0RST (SYS_IPRST1[13]).

- Pin Configuration

Group	Pin Name	GPIO	MFP
SPI0	SPI0_CLK	PB.9, PL.13	MFP5
	SPI0_I2SMCLK	PG.3, PF.14	MFP5
		PB.8	MFP6
	SPI0_MISO	PB.11, PL.15	MFP5
	SPI0_MOSI	PB.10, PL.14	MFP5
	SPI0_SS0	PF.8, PG.0, PL.12	MFP5
	SPI0_SS1	PB.8, PF.9, PG.2	MFP5

6.27.4.2 SPI1 Basic Configuration

- Clock Source Configuration
  - Select the source of SPI1 peripheral clock on SPI1SEL (CLK_CLKSEL4[3:2]).
  - Enable SPI1 peripheral clock in SPI1CKEN (CLK_APBCLK2[5]).
- Reset Configuration
  - Reset SPI1 controller in SPI1RST (SYS_IPRST1[14]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
SPI1	SPI1_CLK	PE.11, PF.5	MFP5
		PF.11, PK.13	MFP6
	SPI1_I2SMCLK	PF.14	MFP6
		PN.14, PN.15	MFP7
	SPI1_MISO	PE.13, PF.7	MFP5
		PF.13, PK.15	MFP6
	SPI1_MOSI	PE.12, PF.6	MFP5
		PF.12, PK.14	MFP6
	SPI1_SS0	PE.10, PF.4	MFP5
		PF.10, PK.12	MFP6
	SPI1_SS1	PN.14	MFP5
		PN.15	MFP6
		PF.14	MFP13

6.27.4.3 SPI2 Basic Configuration

- Clock Source Configuration

- Select the source of SPI2 peripheral clock on SPI2SEL (CLK_CLKSEL4[5:4]).
- Enable SPI2 peripheral clock in SPI2CKEN (CLK_APBCLK2[6]).

- Reset Configuration
  - Reset SPI2 controller in SPI2RST (SYS_IPRST1[15]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
SPI2	SPI2_CLK	PG.9, PK.5, PL.3, PJ.1	MFP5
	SPI2_I2SMCLK	PM.10	MFP7
	SPI2_MISO	PJ.3, PK.4, PK.7, PL.1	MFP5
	SPI2_MOSI	PG.10, PJ.2, PK.6, PL.0	MFP5
	SPI2_SS0	PG.8, PJ.0, PL.2	MFP5
	SPI2_SS1	PI.4	MFP5
		PM.11	MFP7

6.27.4.4 SPI3 Basic Configuration

- Clock Source Configuration
  - Select the source of SPI3 peripheral clock on SPI3SEL (CLK_CLKSEL4[7:6]).
  - Enable SPI3 peripheral clock in SPI3CKEN (CLK_APBCLK2[7]).
- Reset Configuration
  - Reset SPI3 controller in SPI3RST (SYS_IPRST2[6]).
- Pin configuration

Group	Pin Name	GPIO	MFP
SPI3	SPI3_CLK	PG.5, PI.1, PJ.15, PL.9, PK.8	MFP5
	SPI3_I2SMCLK	PK.2, PM.1	MFP5
	SPI3_MISO	PG.7, PI.3, PJ.14, PL.11, PL.5	MFP5
	SPI3_MOSI	PG.6, PI.2, PJ.13, PL.10, PL.4	MFP5
	SPI3_SS0	PG.4, PI.0, PJ.12, PM.10, PL.8	MFP5
	SPI3_SS1	PM.11, PK.3	MFP5
		PD.6	MFP10

SPI/I^{2}S (SPI0~SPI3) Interface Controller Pin description is shown as follows:

Pin	SPI Mode	I^{2}S Mode
SPIx_SS0 / SPIx_SS1	SPI slave selection pin	I^{2}S left/right channel synchronization clock pin (I2Sx_LRCLK)
SPIx_CLK	SPI clock pin	I^{2}S bit clock pin (I2Sx_BCLK)

SPIx_MISO	SPI master input or slave output pin	I²S data input pin (I2Sx_DI)
SPIx_MOSI	SPI master output or slave input pin	I²S data output pin (I2Sx_DO)
SPIx_I2SMCLK	Not available	I²S Master clock output pin

Table 6.27-1 SPI/I²S Interface Controller Pin Description (SPI0~SPI3)

6.27.5 Functional Description

6.27.5.1 Terminology

SPI Peripheral Clock and SPI Bus Clock

The SPI controller needs the peripheral clock to drive the SPI logic unit to perform the data transfer. The peripheral clock rate is determined by the settings of clock divisor (SPIx_CLKDIV) and the clock source which can be PCLK or APLL. SPIxSEL of CLK_CLKSEL4 register determines the clock source of the peripheral clock. The DIVIDER (SPIx_CLKDIV[8:0]) setting determines the divisor of the clock rate calculation.

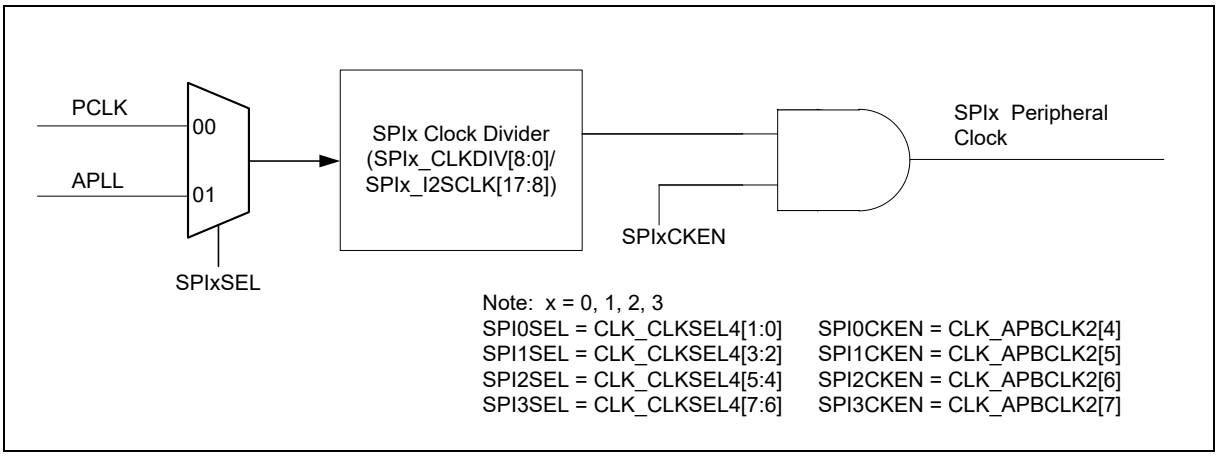


Figure 6.27-2 SPI Peripheral Clock

In Master mode, the frequency of the SPI bus clock is equal to the peripheral clock rate. In general, the SPI bus clock is denoted as SPI clock. In Slave mode, the SPI bus clock is provided by a master device. The frequency of SPI peripheral clock cannot be faster than the system clock rate regardless of Master or Slave mode. If the clock source of peripheral clock is not system clock, the frequency of SPI peripheral clock shall be slower than the system clock frequency regardless of Master or Slave mode.

In I²S mode, the peripheral clock rate is equal to I²S bit clock rate determined by SPIx_I2SCLK register.

Master/Slave mode

The SPI controllers can be set as Master or Slave mode by setting the SLAVE (SPIx_CTL[18]) to communicate with the off-chip SPI slave or master device. The HALFDPX (SPIx_CTL[14]) can be used to select the full-duplex or half-duplex in SPI transmission. The application block diagrams in Master and Slave mode are shown below.

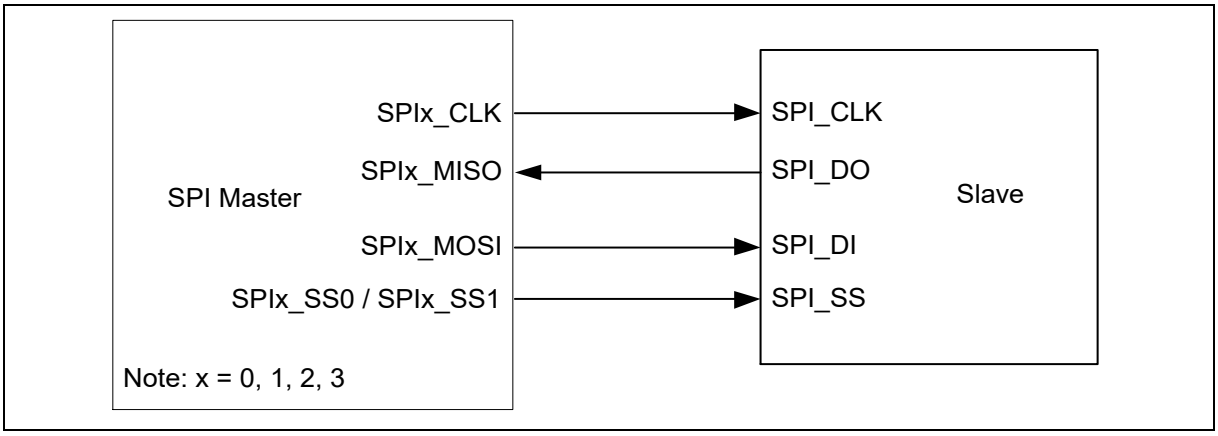


Figure 6.27-3 SPI Full-Duplex Master Mode Application Block Diagram

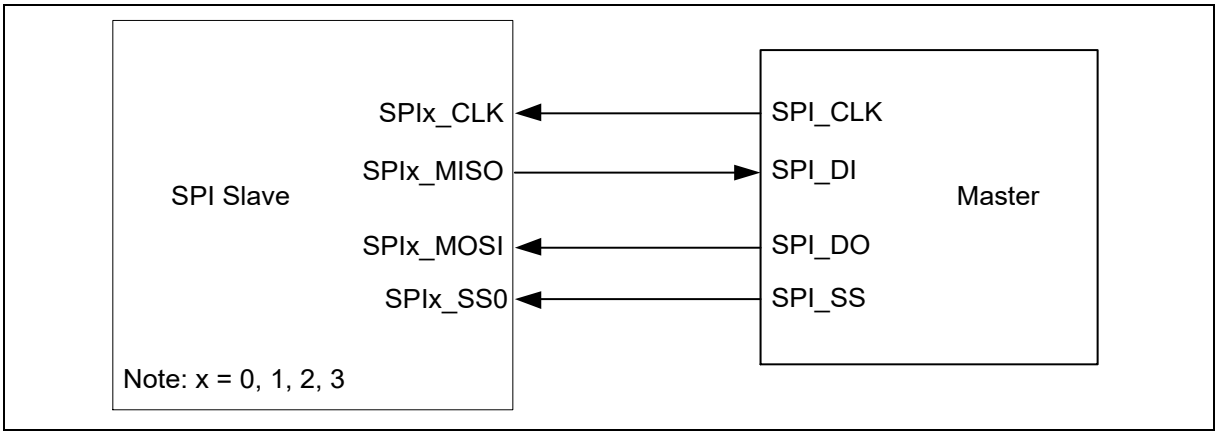


Figure 6.27-4 SPI Full-Duplex Slave Mode Application Block Diagram

Slave Selection

In Master mode, the SPI controller can drive off-chip slave device through the slave select output pin SPIx_SS0 or SPIx_SS1. In Slave mode, the off-chip master device drives the slave selection signal from the SPIx_SS0 input port to this SPI controller. The duration between the slave select active edge and the first SPI clock input shall over 3 SPI peripheral clock cycles of slave.

In Master/Slave mode, the active state of slave selection signal can be programmed to low or high active in SSACTPOL (SPIx_SSCTL[2]). The selection of slave select conditions depends on what type of device is connected. In Slave mode, to recognize the inactive state of the slave selection signal, the inactive period of the slave selection signal must be larger than or equal to 3 peripheral clock cycles between two successive transactions.

In Slave mode, the SPIx_MISO signal type is controlled by the SPIx_SS0 pin. If the SPIx_SS0 pin is inactive state, the SPIx_MISO is set to tri-state to avoid interference with other Slave devices. If the SPIx_SS0 pin is active state, the SPIx_MISO is set to output mode for data transfer.

Timing Condition

The CLKPOL (SPIx_CTL[3]) defines the SPI clock idle state. If CLKPOL = 1, the output SPI clock is idle at high state; if CLKPOL = 0, it is idle at low state.

TXNEG (SPIx_CTL[2]) defines the data transmitted out either on negative edge or on positive edge of SPI clock. RXNEG (SPIx_CTL[1]) defines the data received either on negative edge or on positive edge of SPI clock.

Note: The settings of TXNEG and RXNEG are mutual exclusive. In other words, do not transmit and

receive data at the same clock edge.

Transmit/Receive Bit Length

The bit length of a transaction word is defined in DWIDTH (SPIx_CTL[12:8]) and can be configured up to 32-bit length in a transaction word for transmitting and receiving.

When SPI controller finishes a transaction, i.e. receives or transmits a specific count of bits defined in DWIDTH (SPIx_CTL[12:8]), the unit transfer interrupt flag will be set to 1.

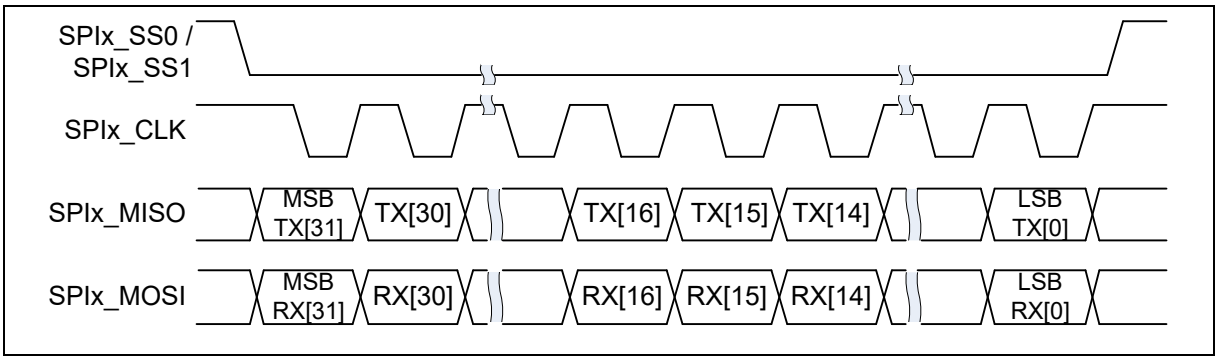


Figure 6.27-5 32-bit in One Transaction

LSB/MSB First

LSB (SPIx_CTL[13]) defines the bit transfer sequence in a transaction. If the LSB (SPIx_CTL[13]) is set to 1, the transfer sequence is LSB first. The bit 0 will be transferred firstly. If the LSB (SPIx_CTL[13]) is cleared to 0, the transfer sequence is MSB first.

Suspend Interval

SUSPITV (SPIx_CTL[7:4]) provides a configurable suspend interval, 0.5 ~ 15.5 SPI clock periods, between two successive transaction words in Master mode. The definition of the suspend interval is the interval between the last clock edge of the preceding transaction word and the first clock edge of the following transaction word. The default value of SUSPITV is 0x3 (3.5 SPI clock cycles).

6.27.5.2 Automatic Slave Selection

In Master mode, if AUTOSS (SPIx_SSCTL[3]) is set, the slave selection signal will be generated automatically and output to the SPIx_SS0 or SPIx_SS1 pin according to whether SS0 (SPIx_SSCTL[0]) or SS1 (SPIx_SSCTL[1]) is enabled or not. The slave selection signal will be set to active state by the SPI controller when the SPI data transfer is started by writing to FIFO. It will be set to inactive state when SPI bus is idle. If SPI bus is not idle, i.e. TX FIFO, TX shift register or TX skew buffer is not empty, the slave selection signal will be set to inactive state between transactions if the value of SUSPITV (SPIx_CTL[7:4]) is greater than or equal to 3.

In Master mode, if the value of SUSPITV is less than 3 and the AUTOSS is set as 1, the slave selection signal will be kept at active state between two successive transactions.

If the AUTOSS bit is cleared, the slave selection output signal will be determined by the SS0 or SS1 setting. The active state of the slave selection output signal is specified in SSACTPOL (SPIx_SSCTL[2]).

The duration between the slave selection signal active edge and the first SPI bus clock edge is 1 SPI bus clock cycle and the duration between the last SPI bus clock and the slave selection signal inactive edge is 1.5 SPI bus clock cycle.

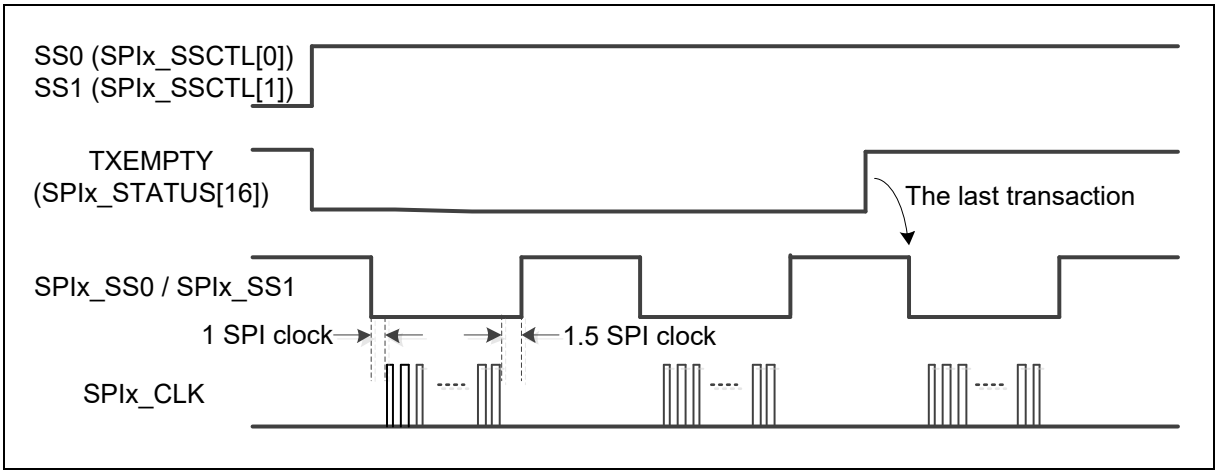


Figure 6.27-6 Automatic Slave Selection (SSACTPOL = 0, SUSPITV > 0x2)

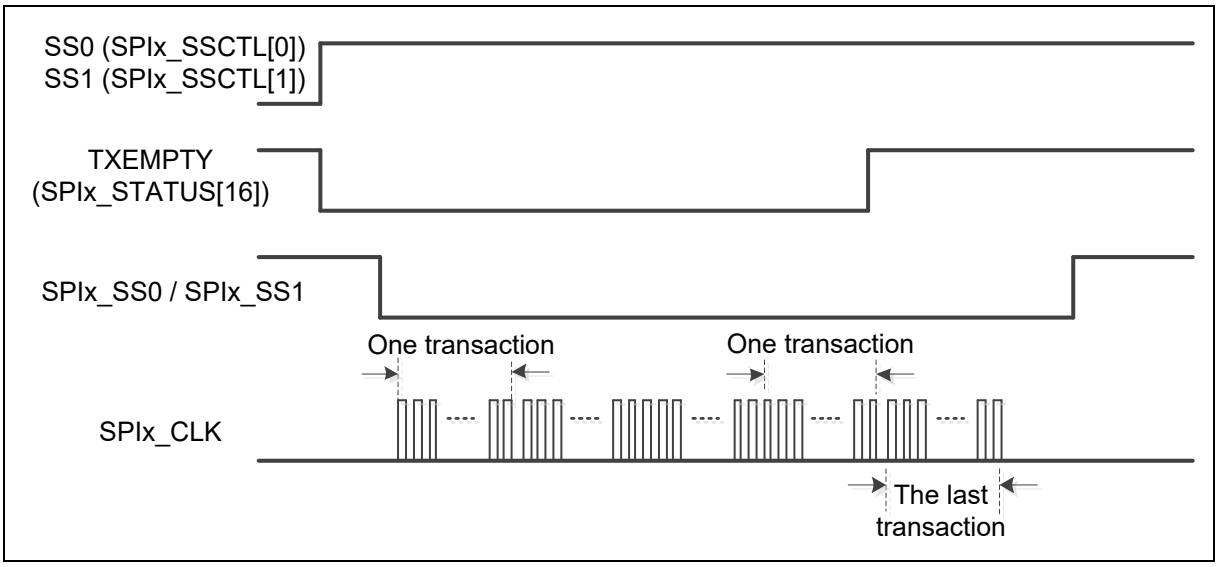


Figure 6.27-7 Automatic Slave Selection (SSACTPOL = 0, SUSPITV < 0x3)

6.27.5.3 Byte Reorder and Suspend Function

When the transfer is set as MSB first (LSB = 0) and the REORDER (SPIx_CTL[19]) is set to 1, the data stored in the TX buffer and RX buffer will be rearranged in the order as [Byte0, Byte1, Byte2, Byte3] in 32-bit transfer (DWIDTH = 0). The sequence of transmitted/received data will be Byte0, Byte1, Byte2, and then Byte3. If the DWIDTH is set as 24-bit transfer mode, the data in TX buffer and RX buffer will be rearranged as [unknown byte, Byte0, Byte1, Byte2]. The SPI controller will transmit/receive data with the sequence of Byte0, Byte1 and then Byte2. Each byte will be transmitted/received with MSB first. The rule of 16-bit mode is the same as above. Byte Reorder function is only available when DWIDTH is configured as 16, 24, and 32 bits.

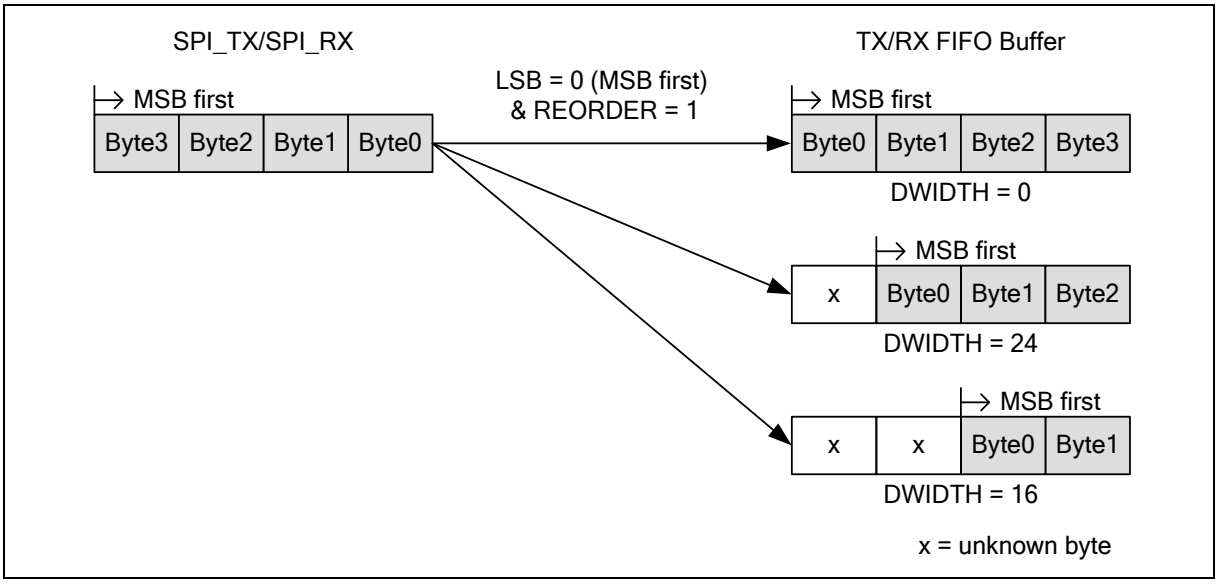


Figure 6.27-8 Byte Reorder Function

In Master mode, if REORDER (SPIx_CTL[19]) is set to 1, a suspend interval of 0.5 ~ 15.5 SPI clock periods will be inserted by hardware between two successive bytes in a transaction word. The suspend interval is configured in SUSPITV (SPIx_CTL[7:4]).

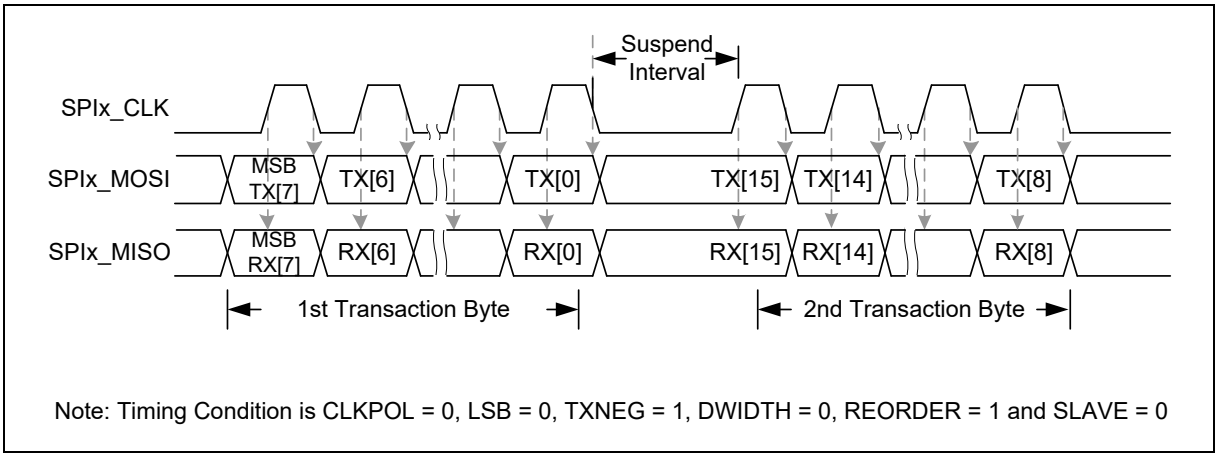


Figure 6.27-9 Timing Waveform for Byte Suspend

6.27.5.4 Half-Duplex Communication

The SPI controller can communicate in half-duplex mode by setting HALFDPX (SPIx_CTL[14]) bit. In half-duplex mode, there is only one data line for receiving or transmitting data direction which is defined by DATDIR (SPIx_CTL[20]). In half-duplex configuration, the SPIx_MISO pin is free for other applications and it can be configured as GPIO. Enabling or disabling the control bit HALFDPX (SPIx_CTL[14]) will produce TXFBCLR (SPIx_FIFOCTL[9]) and RXFBCLR (SPIx_FIFOCTL[8]) at the same time automatically.

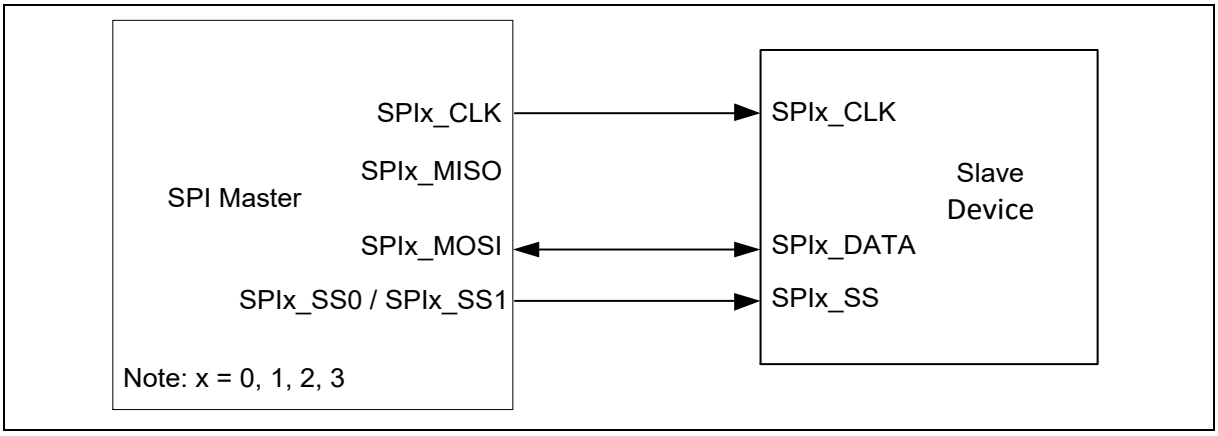


Figure 6.27-10 SPI Half-Duplex Master Mode Application Block Diagram

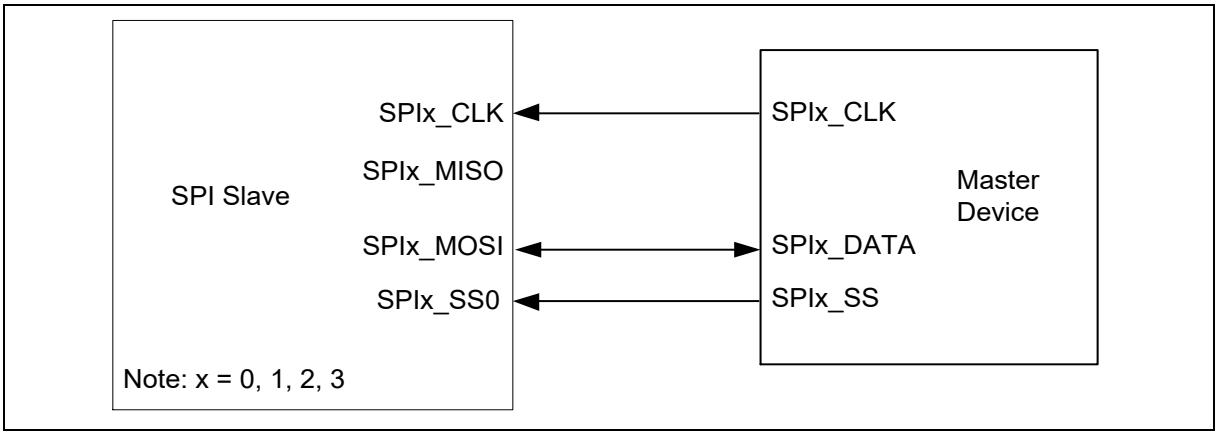


Figure 6.27-11 SPI Half-Duplex Slave Mode Application Block Diagram

6.27.5.5 Receive-Only Mode

The SPI Master device can communicate in receive-only mode by setting RXONLY (SPIx_CTL[15]). In this configuration, the SPI Master device will generate SPI bus clock continuously as long as the receive-only mode is enabled for receiving data bit from SPI slave device. If AUTOSS (SPIx_SSCTL[3]) is enabled in receive-only mode, SPI Master will keep activating the slave select signal.

The remaining SPIx_MOSI pin of SPI Master device is not used for communication and can be configured as GPIO. The status BUSY (SPIx_STATUS[0]) will be asserted in receive-only mode due to the generation of SPI bus clock. Entering this mode will produce the TXFBCLR (SPIx_FIFOCTL[9]) and RXFBCLR (SPIx_FIFOCTL[8]) at the same time automatically. When user enables this mode, the output SPI bus clock will be sent out after 6 peripheral clock cycles. In this mode, the data which has been written into transmit FIFO will be loaded into transmit shift register and sent out.

When user sets RXONLY (SPIx_CTL[15]) enable, SPI RX data with data bit width of DWIDTH (SPIx_CTL[12:8]) will be received into RX FIFO and SPI clock will be sent to SPI slave device until RX FIFO is full.

For data bit width of 8~16 bits, the SPI master will send SPI output clock to SPI slave and receive RX data when RX FIFO counter RXCNT (SPIx_STATUS[27:24]) is less than or equal to 14.

For data bit width of 17~32 bits, the SPI master will send SPI output clock to SPI slave and receive RX data when RX FIFO counter RXCNT (SPIx_STATUS[27:24]) is less than or equal to 6.

6.27.5.6 Slave 3-Wire Mode

When SLV3WIRE (SPIx_SSCTL[4]) is set by software to enable the Slave 3-Wire mode, the SPI controller can work with no slave selection signal in Slave mode. The SLV3WIRE (SPIx_SSCTL[4]) only takes effect in Slave mode. Only three pins, SPIx_CLK, SPIx_MISO, and SPIx_MOSI, are required to communicate with a SPI master. The SPIx_SS0 and SPIx_SS1 pin can be configured as a GPIO. When the SLV3WIRE (SPIx_SSCTL[4]) is set to 1, the SPI slave will be ready to transmit/receive data after the SPIEN (SPIx_CTL[0]) is set to 1.

6.27.5.7 PDMA Transfer Function

SPI controller supports PDMA transfer function.

When TXPDMAEN (SPIx_PDMACTL[0]) is set to 1, the controller will issue request to PDMA controller to start the PDMA transmission process automatically. Data width DWIDTH (SPIx_CTL[12:8]) only supports 8-bits, 16-bits, and 32-bits when TXPDMAEN (SPIx_PDMACTL[0]) is set to 1 or RXPDMAEN (SPIx_PDMACTL[1]) is set to 1.

The TX FIFO will be configured to 8x32 bits automatically when TXPDMAEN (SPIx_PDMACTL[0]) is set to 1 and SPI data width DWIDTH (SPIx_CTL[12:8]) is 32 bits. The TX FIFO will be configured to 16x16 bits automatically when TXPDMAEN (SPIx_PDMACTL[0]) is set to 1 and SPI data width DWIDTH (SPIx_CTL[12:8]) is 16 bits. The TX FIFO will be configured to 32x8 bits automatically when TXPDMAEN (SPIx_PDMACTL[0]) is set to 1 and SPI data width DWIDTH (SPIx_CTL[12:8]) is 8 bits.

When RXPDMAEN (SPIx_PDMACTL[1]) is set to 1, the controller will start the PDMA reception process. SPI controller will issue request to PDMA controller automatically when there is data in the RX FIFO buffer.

The RX FIFO will be configured to 8x32 bits automatically when RXPDMAEN (SPIx_PDMACTL[1]) is set to 1 and SPI data width DWIDTH (SPIx_CTL[12:8]) is 32 bits. The RX FIFO will be configured to 16x16 bits automatically when RXPDMAEN (SPIx_PDMACTL[1]) is set to 1 and SPI data width DWIDTH (SPIx_CTL[12:8]) is 16 bits. The RX FIFO will be configured to 32x8 bits automatically when RXPDMAEN (SPIx_PDMACTL[1]) is set to 1 and SPI data width DWIDTH (SPIx_CTL[12:8]) is 8 bits.

Note: SPI supports single request PDMA (Read/Write) only, burst request PDMA is not supported.

6.27.5.8 FIFO Buffer Operation

The SPI controllers equip with eight 32-bit wide transmit and receive FIFO buffers. The data stored in the transmit FIFO buffer will be read and sent out by the transmission control logic. If the transmit FIFO buffer is full, the TXFULL (SPIx_STATUS[17]) will be set to 1. When the SPI transmission logic unit draws out the last datum of the transmit FIFO buffer, so that the transmit FIFO buffer is empty, the TXEMPTY (SPIx_STATUS[16]) will be set to 1. Note that the TXEMPTY (SPIx_STATUS[16]) flag is set to 1 while the last transaction is still in progress. In Master mode, the BUSY (SPIx_STATUS[0]) is set to 1 when the FIFO buffer is written any data or there is any transaction on the SPI bus. (e.g. the slave selection signal is active and the SPI controller is receiving data in Slave mode). It will be set to 0 when the transmit FIFO is empty and the current transaction has done. Thus, the status of BUSY (SPIx_STATUS[0]) should be checked by software to make sure whether the SPI is in idle or not.

The receive control logic will store the SPI input data into the receive FIFO buffer. There are FIFO related status bits, like RXEMPTY (SPIx_STATUS[8]) and RXFULL (SPIx_STATUS[9]), to indicate the current status of RX FIFO buffer.

The transmitting and receiving threshold can be configured by setting TXTH (SPIx_FIFOCTL[31:28]) and RXTH (SPIx_FIFOCTL[27:24]). When the count of valid data stored in transmit FIFO buffer is less than or equal to TXTH (SPIx_FIFOCTL[31:28]) setting, TXTHIF (SPIx_STATUS[18]) will be set to 1. When the count of valid data stored in receive FIFO buffer is larger than RXTH (SPIx_FIFOCTL[27:24]) setting, RXTHIF (SPIx_STATUS[10]) will be set to 1.

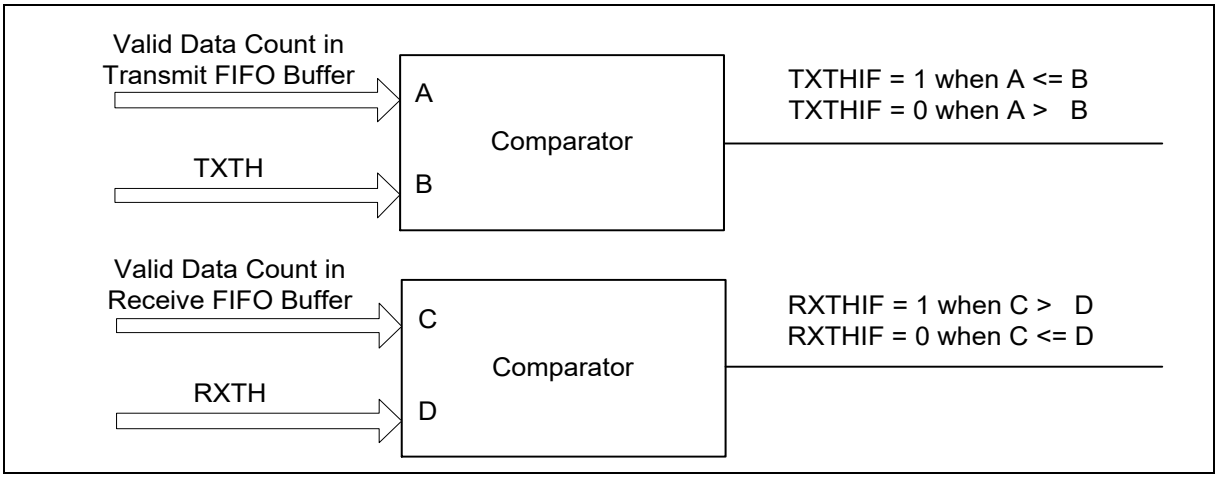


Figure 6.27-12 FIFO Threshold Comparator

In Master mode, when the first datum is written to the SPIx_TX register, the TXEMPTY flag (SPIx_STATUS[16]) will be cleared to 0. The transmission will start after 1 PCLK clock cycles and 6 peripheral clock cycles. User can write the next data into SPIx_TX register immediately. The SPI controller will insert a suspend interval between two successive transactions. The period of suspend interval is decided by the setting of SUSPITV (SPIx_CTL[7:4]). If the SUSPITV (SPIx_CTL[7:4]) equals 0, the SPI controller can perform continuous transfer. User can write data into SPIx_TX register as long as the TXFULL (SPIx_STATUS[17]) is 0.

In the Example 1 of Figure 6.27-13, it indicates the updated condition of TXEMPTY (SPIx_STATUS[16]) and the relationship among the FIFO buffer, shift register and the skew buffer for 8~16 bits of data length. The TXEMPTY (SPIx_STATUS[16]) is set to 0 when the Data0 is written into the FIFO buffer. The Data0 will be loaded into the shift register by the core logic and the TXEMPTY (SPIx_STATUS[16]) will be to 1. The Data0 in shift register will be shift into skew buffer by bit for transmission until the transfer is done.

In the Example 2 of Figure 6.27-13, it indicates the updated condition of TXFULL (SPIx_STATUS[17]) when there are 16 data in the FIFO buffer and the next data of Data17 does not be written into the FIFO buffer when the TXFULL = 1.

In the Example 1 of Figure 6.27-14, it indicates the updated condition of TXEMPTY (SPIx_STATUS[16]) and the relationship among the FIFO buffer, shift register and the skew buffer for 17~32 bits of data length.

In the Example 2 of Figure 6.27-14, it indicates the updated condition of TXFULL (SPIx_STATUS[17]) when there are 8 data in the FIFO buffer and the next data of Data9 does not be written into the FIFO buffer when the TXFULL = 1.

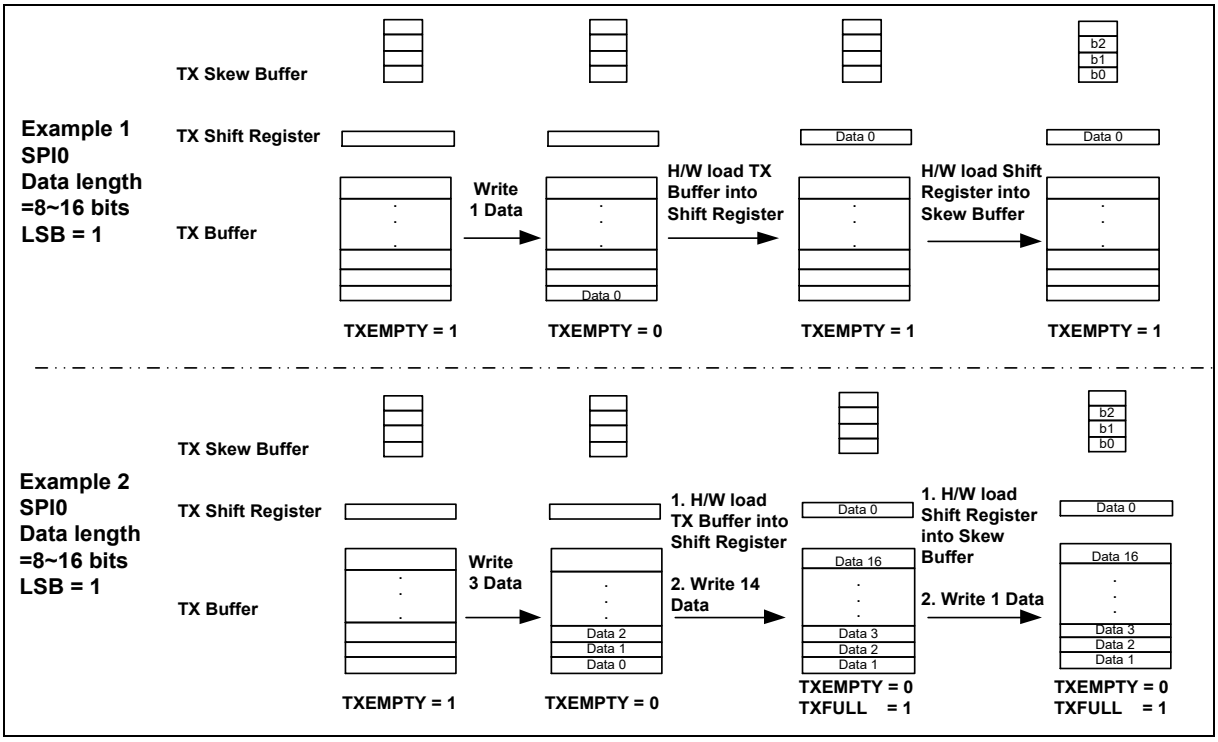


Figure 6.27-13 Transmit FIFO Buffer Example for 8~16 Bits of Data Length

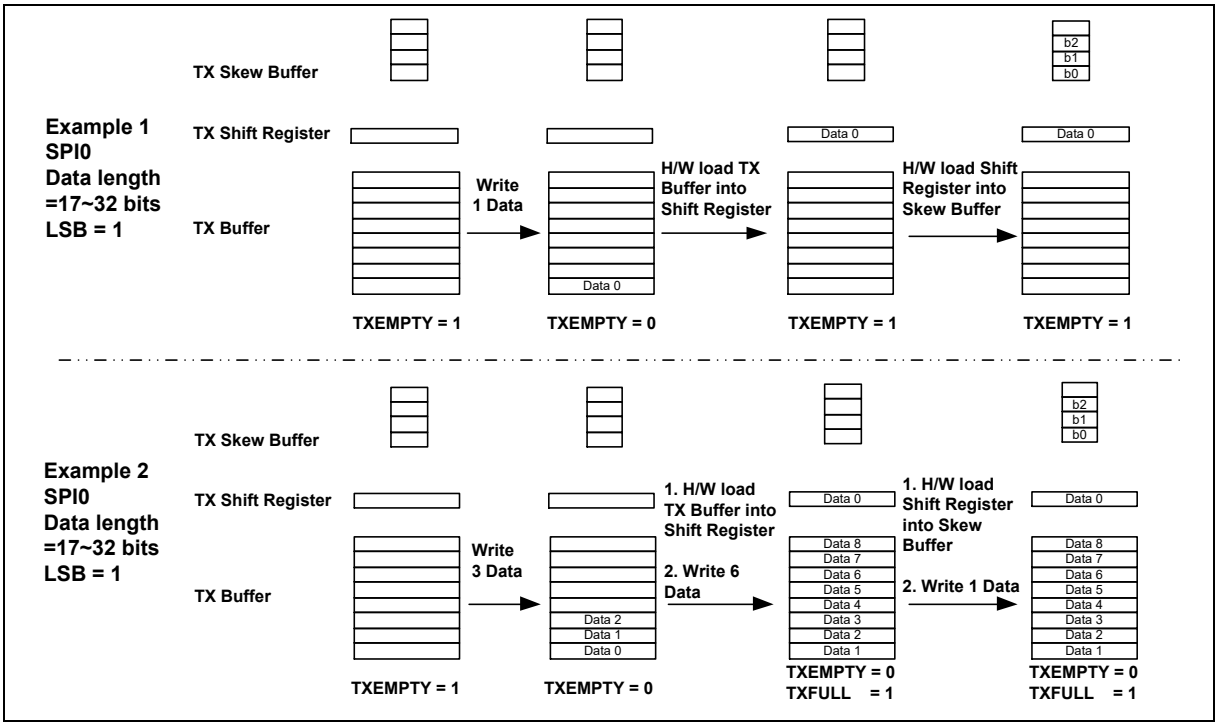


Figure 6.27-14 Transmit FIFO Buffer Example for 17~32 Bits of Data Length

The subsequent transactions will be triggered automatically if the transmitted data are updated in time. If the SPIx_TX register does not be updated after all data transfers are done, the transfer will stop.

In Master mode, during receiving operation, the serial data are received from SPIx_MISO pin and stored to receive FIFO buffer.

The received data (Data0's b0, b1, …b31) is stored into skew buffer first according the serial clock (SPIx_CLK) and then it is shift into the shift register bit by bit. The core logic will load the data in shift register into FIFO buffer when the received data bit count reach the value of DWIDTH (SPIx_CTL[12:8]).

The RXEMPTY (SPIx_STATUS[8]) will be cleared to 0 while the receive FIFO buffer contains unread data (see the Example 1 of Receive FIFO Buffer Example in Figure 6.27-15). The received data can be read by software from SPIx_RX register as long as the RXEMPTY (SPIx_STATUS[8]) is 0. If the receive FIFO buffer contains 16 unread data, the RXFULL (SPIx_STATUS[9]) will be set to 1 (see the Example 2 of Receive FIFO Buffer Example in Figure 6.27-15).

The RXEMPTY (SPIx_STATUS[8]) will be cleared to 0 while the receive FIFO buffer contains unread data (see the Example 1 of Receive FIFO Buffer Example in Figure 6.27-16). The received data can be read by software from SPIx_RX register as long as the RXEMPTY (SPIx_STATUS[8]) is 0. If the receive FIFO buffer contains 8 unread data, the RXFULL (SPIx_STATUS[9]) will be set to 1 (see the Example 2 of Receive FIFO Buffer Example in Figure 6.27-16).

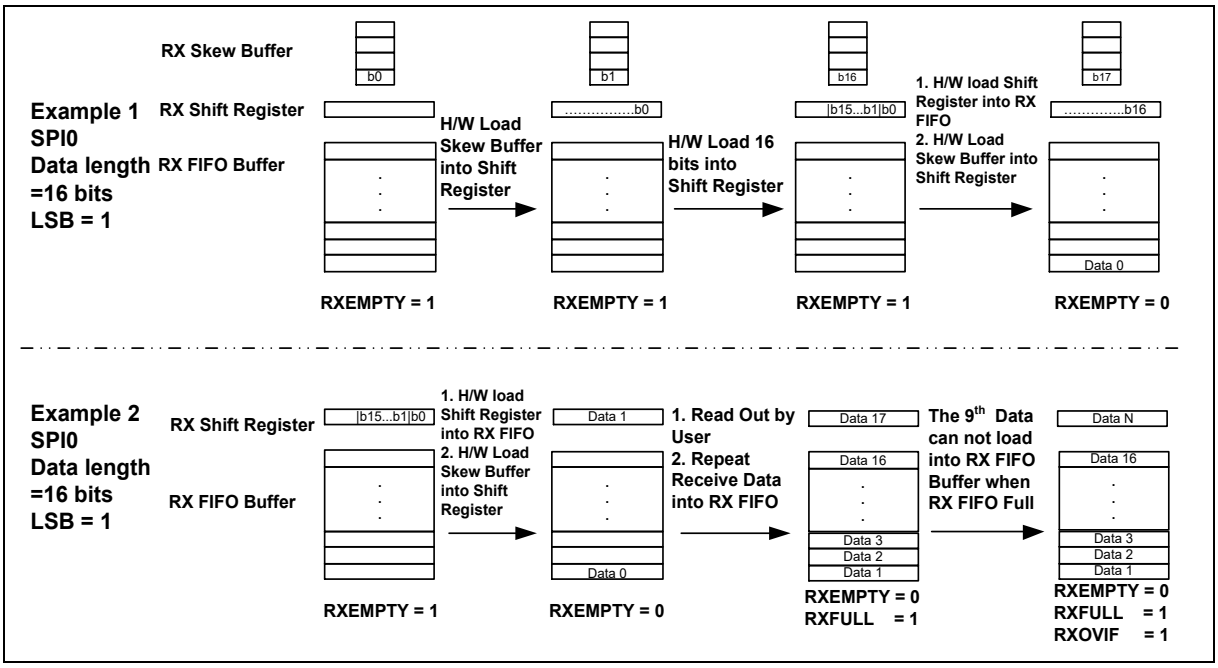


Figure 6.27-15 Receive FIFO Buffer Example for 16 Bits of Data Length

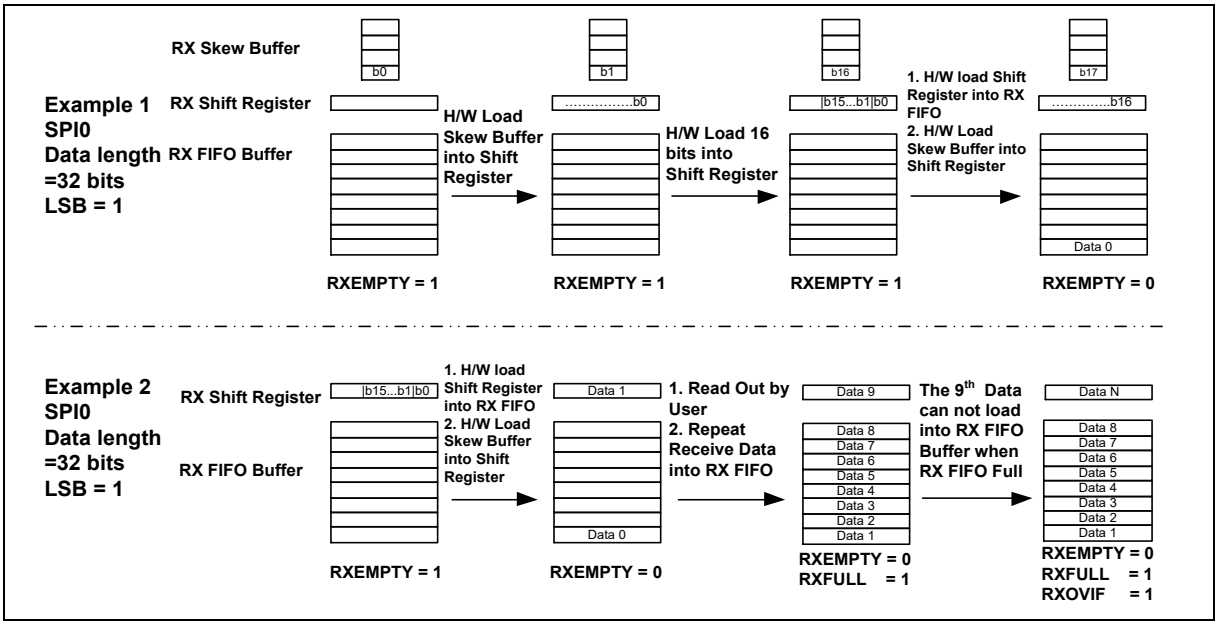


Figure 6.27-16 Receive FIFO Buffer Example for 32 Bits of Data Length

In Slave mode, during transmission operation, when data is written to the SPIx_TX register by software, the data will be loaded into transmit FIFO buffer and the TXEMPTY (SPIx_STATUS[16]) will be set to 0. The transmission will start when the slave device receives clock signal from master. Data can be written to SPIx_TX register as long as the TXFULL (SPIx_STATUS[17]) is 0. After all data have been drawn out by the SPI transmission logic unit and the SPIx_TX register is not updated by software, the TXEMPTY (SPIx_STATUS[16]) will be set to 1.

If there is no any data written to the SPIx_TX register, the transmit underflow interrupt flag, TXUFIF (SPIx_STATUS[19]) will be set to 1 when the slave selection signal is active. The output data will be held by TXUFPOL (SPIx_FIFOCTL[6]) setting during this transfer until the slave selection signal goes to inactive state. When the transmit underflow event occurs, the slave under run interrupt flag, SLVURIF (SPIx_STATUS[7]), will be set to 1 as SPIx_SS0 goes to inactive state.

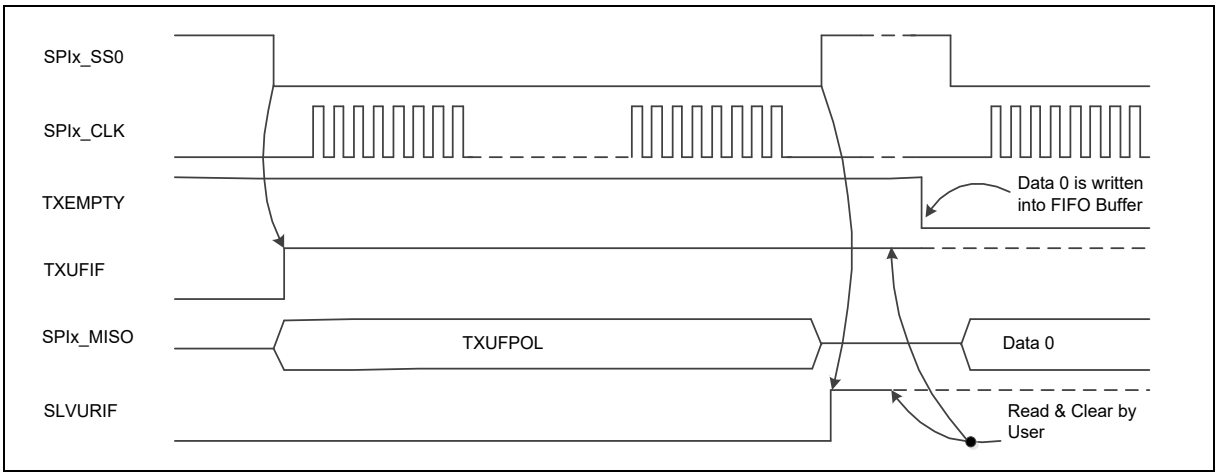


Figure 6.27-17 TX Underflow Event and Slave Under Run Event

In SPI Slave 3-Wire mode, the first 2-bit data is un-predicted (kept on the level of last bit in previously transfer) if the data is written into TX FIFO among 3 peripheral clock cycles before the SPI bus clock is presented. The other bits are held by TXUFPOL (SPIx_FIFOCTL[6]) because there is TX underflow event. The written data will be transmitted in the next transfer.

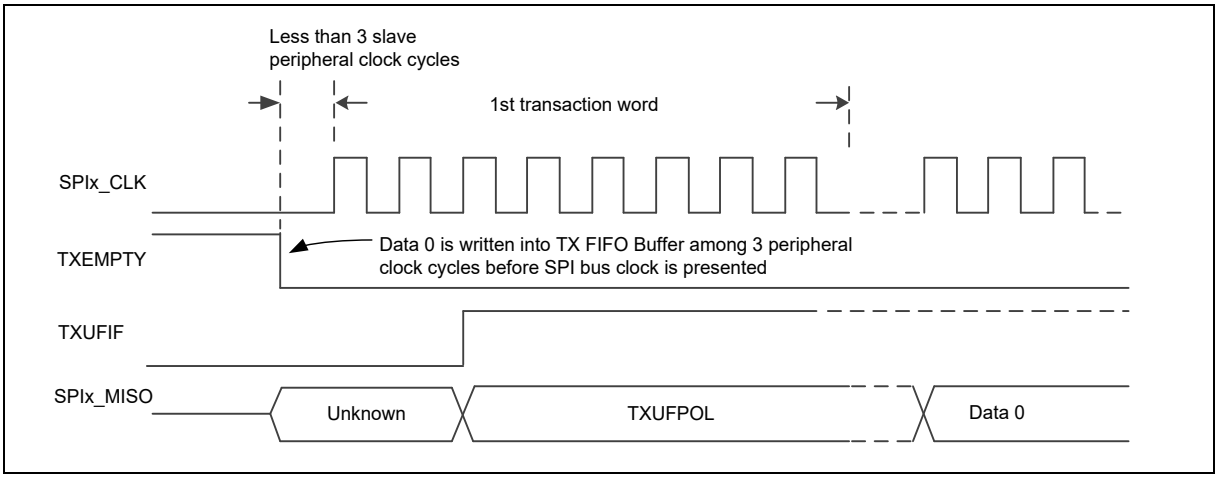


Figure 6.27-18 TX Underflow Event (SPI Slave 3-Wire Mode Enabled)

In Slave mode, during receiving operation, the serial data is received from SPIx_MOSI pin and stored to SPIx_RX register. The reception mechanism is similar to Master mode reception operation. If the receive FIFO buffer contains 8 unread data, the RXFULL (SPIx_STATUS[9]) will be set to 1 and the RXOVIF (SPIx_STATUS[11]) will be set to 1 if there is more serial data received from SPIx_MOSI and follow-up data will be dropped (refer to the Receive FIFO Buffer Examples in Figure 6.27-15 and Figure 6.27-16).

When the receive bit count is mismatched with the DWIDTH (SPIx_CTL[12:8]) and the slave selection line goes to inactive state, the SLVBEIF (SPIx_STATUS[6]) will be set to 1. RX data will be written into RX FIFO and SLVBENUM (SPIx_STATUS2[29:24]) will be updated when SLVBERX (SPIx_FIFOCTL[10]) is enabled and error event of SPI slave bit count happened. The RX data will be dropped and SLVBENUM (SPIx_STATUS2[29:24]) will be fixed to 0x0 when the control register SLVBERX (SPIx_FIFOCTL[10]) is disabled and error event of SPI slave bit count happened. The status register SLVBENUM (SPIx_STATUS2[29:24]) indicates that the effective bit number of uncompleted RX data when error event of SPI slave bit count happened. In Figure 6.27-19, the timing diagram of SPI slave bit count error and SLVBENUM is shown. SLVBENUM will be updated to 0x8 when SPI slave device receives 8 bits data and DWIDTH (SPIx_CTL[12:8]) set to 16 bits. The SPI slave bit count error event (SLVBEIF) generates the pulse of RX FIFO write operation to write RX data from RX shift register to RX FIFO, and RX FIFO count (RXCNT) updates to 1.

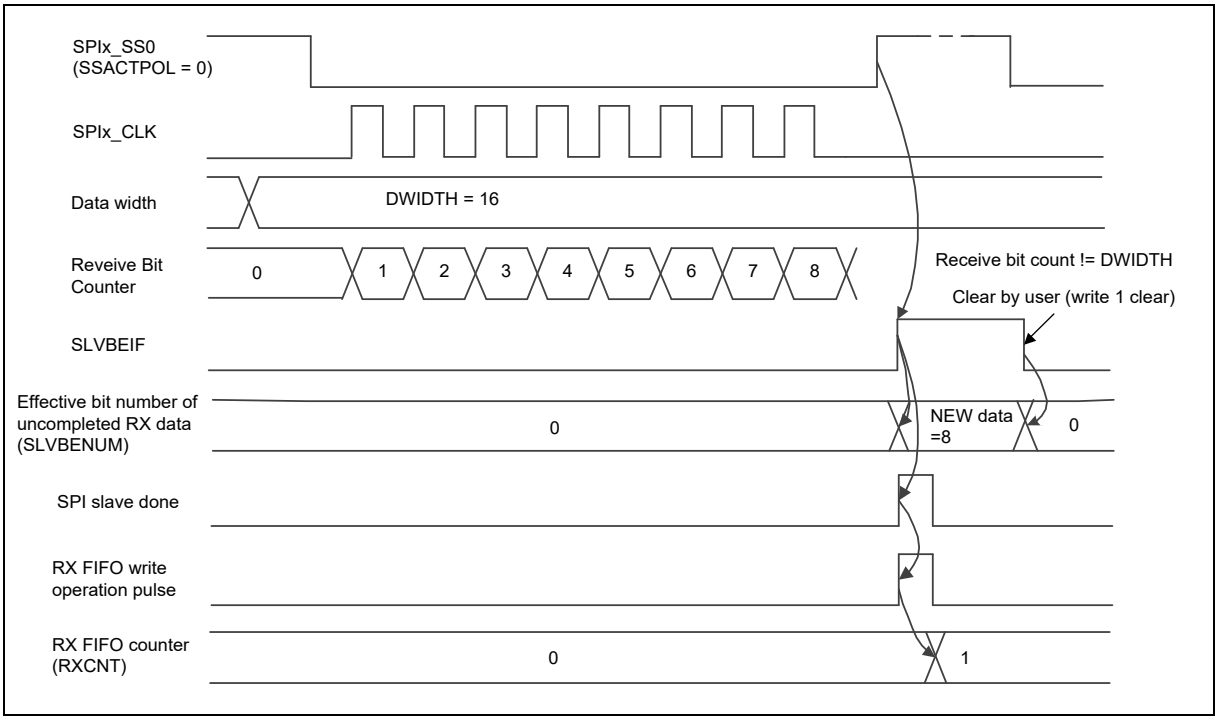


Figure 6.27-19 Slave Mode Bit Count Error and Effective Bit Number of Uncompleted RX Data

A receive time-out function is built-in in this controller. When the receive FIFO is not empty and no read operation in receive FIFO over 64 SPI peripheral clock periods in Master mode or over 576 SPI peripheral clock periods in Slave mode, the receive time-out occurs and the RXTOIF (SPIx_STATUS[12]) will be set to 1. When the receive FIFO is read by user, the time-out status will be cleared automatically.

6.27.5.9 Interrupt

- SPI unit transfer interrupt

  As the SPI controller finishes a unit transfer, the unit transfer interrupt flag UNITIF (SPIx_STATUS[1]) will be set to 1. The unit transfer interrupt event will generate an interrupt to CPU if the unit transfer interrupt enable bit UNITIEN (SPIx_CTL[17]) is set. The unit transfer interrupt flag can be cleared only by writing 1 to it.

- SPI slave selection active/inactive interrupt

  In Slave mode, the slave selection active/inactive interrupt flag, SSACTIF (SPIx_STATUS[2]) and SSINAIF (SPIx_STATUS[3]), will be set to 1 when the SPIEN (SPIx_CTL[0]) and SLAVE (SPIx_CTL[18]) are set to 1 and the slave selection signal goes to active/inactive state. The SPI controller will issue an interrupt if the SSINAIEN (SPIx_SSCTL[13]) or SSACTIEN (SPIx_SSCTL[12]), are set to 1.

- Slave bit count error interrupt

  In Slave mode, if the transmit/receive bit count is mismatched with the DWIDTH (SPIx_CTL[12:8]) when the slave selection line goes to inactive state, the SLVBEIF (SPIx_STATUS[6]) will be set to 1. The uncompleted transaction will be dropped from TX shift register. When the control register SLVBERX (SPIx_FIFOCTL[10]) is disabled and SPI slave bit count error event happened, the uncompleted transaction data will be dropped from RX shift registers. When control register SLVBERX (SPIx_FIFOCTL[10]) is enabled and SPI slave bit count error event happened, the uncompleted transaction data will be written from RX shift registers into RX FIFO. The SPI controller will issue an interrupt if the SLVBEIEN (SPIx_SSCTL[8]) is set to 1.

Note: If the slave selection signal is active but there is no any serial clock input, the SLVBEIF (SPIx_STATUS[6]) will be set to 1 when the slave selection signal goes to inactive state.

- TX underflow interrupt

  In SPI Slave mode, if there is no any data is written to the SPIx_TX register, the TXUFIF (SPIx_STATUS[19]) will be set to 1 when the slave selection signal is active. The SPI controller will issue a TX underflow interrupt if the TXUFIEN (SPIx_FIFOCTL[7]) is set to 1.

  Note: If underflow event occurs in SPI Slave mode, there are two conditions which make SPI Slave mode return to idle state and then goes for next transfer: (1) set TXRST to 1 (2) slave select signal is changed to inactive state while SLV3WIRE=0.

- Slave TX under run interrupt

  If the TX underflow event occurs, the SLVURIF (SPIx_STATUS[7]) will be set to 1 when SPIx_SS goes to inactive state. The SPI controller will issue a TX under run interrupt if the SLVURIEN (SPIx_SSCTL[9]) is set to 1.

  Note: In Slave 3-Wire mode, the slave selection signal is considered active all the time so that user shall poll the TXUFIF (SPIx_STATUS[19]) to know if there is TX underflow event or not.

- Receive Overrun interrupt

  In Slave mode, if the receive FIFO buffer contains 8 unread data, the RXFULL (SPIx_STATUS[9]) will be set to 1 and the RXOVIF (SPIx_STATUS[11]) will be set to 1 if there is more serial data received from SPI bus and follow-up data will be dropped. The SPI controller will issue an interrupt if the RXOVIEN (SPIx_FIFOCTL[5]) is set to 1.

- Receive FIFO time-out interrupt

  If there is a received data in the FIFO buffer and it is not read by software over 64 SPI peripheral clock periods in Master mode or over 576 SPI peripheral clock periods in Slave mode, it will send a RX time-out interrupt to the system if the RX time-out interrupt enable bit, RXTOIEN (SPIx_FIFOCTL[4]), is set to 1.

- Transmit FIFO interrupt

  In FIFO mode, if the valid data count of the transmit FIFO buffer is less than or equal to the setting value of TXTH (SPIx_FIFOCTL[31:28]), the transmit FIFO interrupt flag TXTHIF (SPIx_STATUS[18]) will be set to 1. The SPI controller will generate a transmit FIFO interrupt to the system if the transmit FIFO interrupt enable bit, TXTHIEN (SPIx_FIFOCTL[3]), is set to 1.

- Receive FIFO interrupt

  In FIFO mode, if the valid data count of the receive FIFO buffer is larger than the setting value of RXTH (SPIx_FIFOCTL[27:24]), the receive FIFO interrupt flag RXTHIF (SPIx_STATUS[10]) will be set to 1. The SPI controller will generate a receive FIFO interrupt to the system if the receive FIFO interrupt enable bit, RXTHIEN (SPIx_FIFOCTL[2]), is set to 1.

6.27.5.10 I²S Mode

The SPI0~SPI3 controllers support I²S mode with PCM mode A, PCM mode B, MSB justified and I²S data format. The bit count of an audio channel is determined by WDWIDTH (SPIx_I2SCTL[5:4]). The transfer sequence is always first from the most significant bit, MSB. Data are read on rising clock edge and are driven on falling clock edge.

In I²S data format, the MSB is sent and latched on the second clock of an audio channel. The I2Sx_LRCLK signal indicates which audio channel is in transferring.

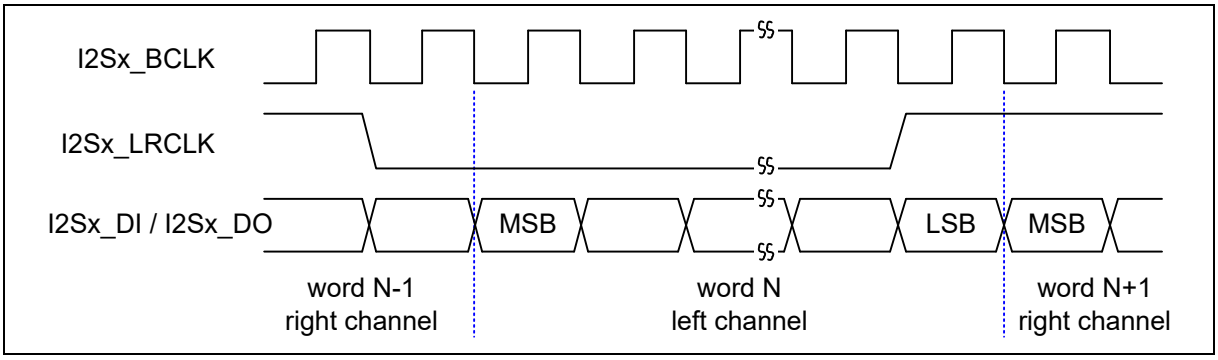


Figure 6.27-20 I²S Data Format Timing Diagram

In MSB justified data format, the MSB is sent and latched on the first clock of an audio channel.

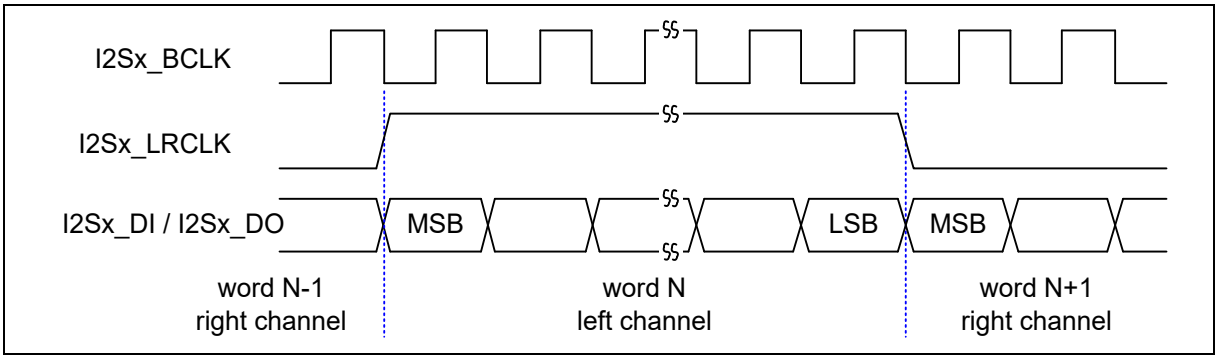


Figure 6.27-21 MSB Justified Data Format Timing Diagram

The I2Sx_LRCLK signal also supports PCM mode A and PCM mode B. The I2Sx_LRCLK signal in PCM mode indicates the beginning of an audio frame.

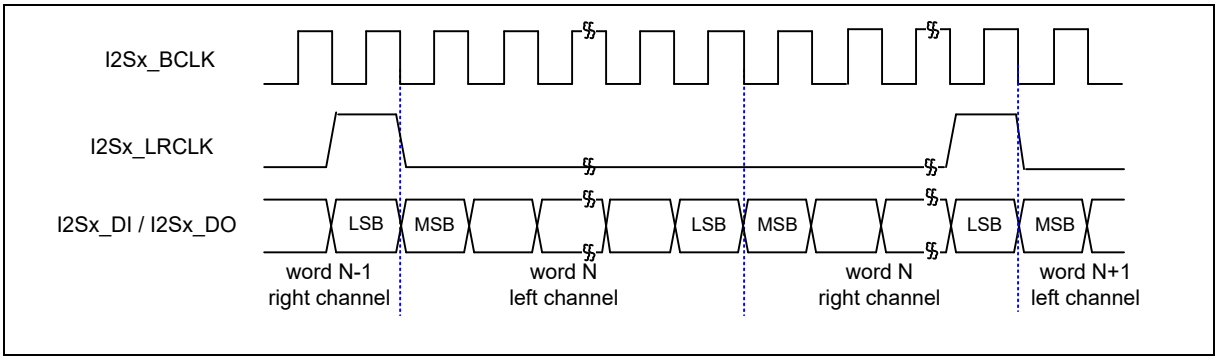


Figure 6.27-22 PCM Mode A Timing Diagram

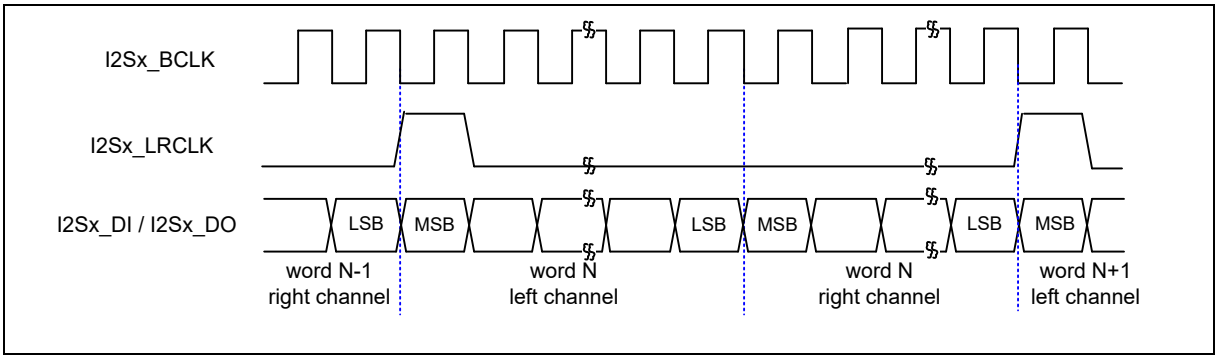


Figure 6.27-23 PCM Mode B Timing Diagram

6.27.5.11 I²S Mode FIFO Operation

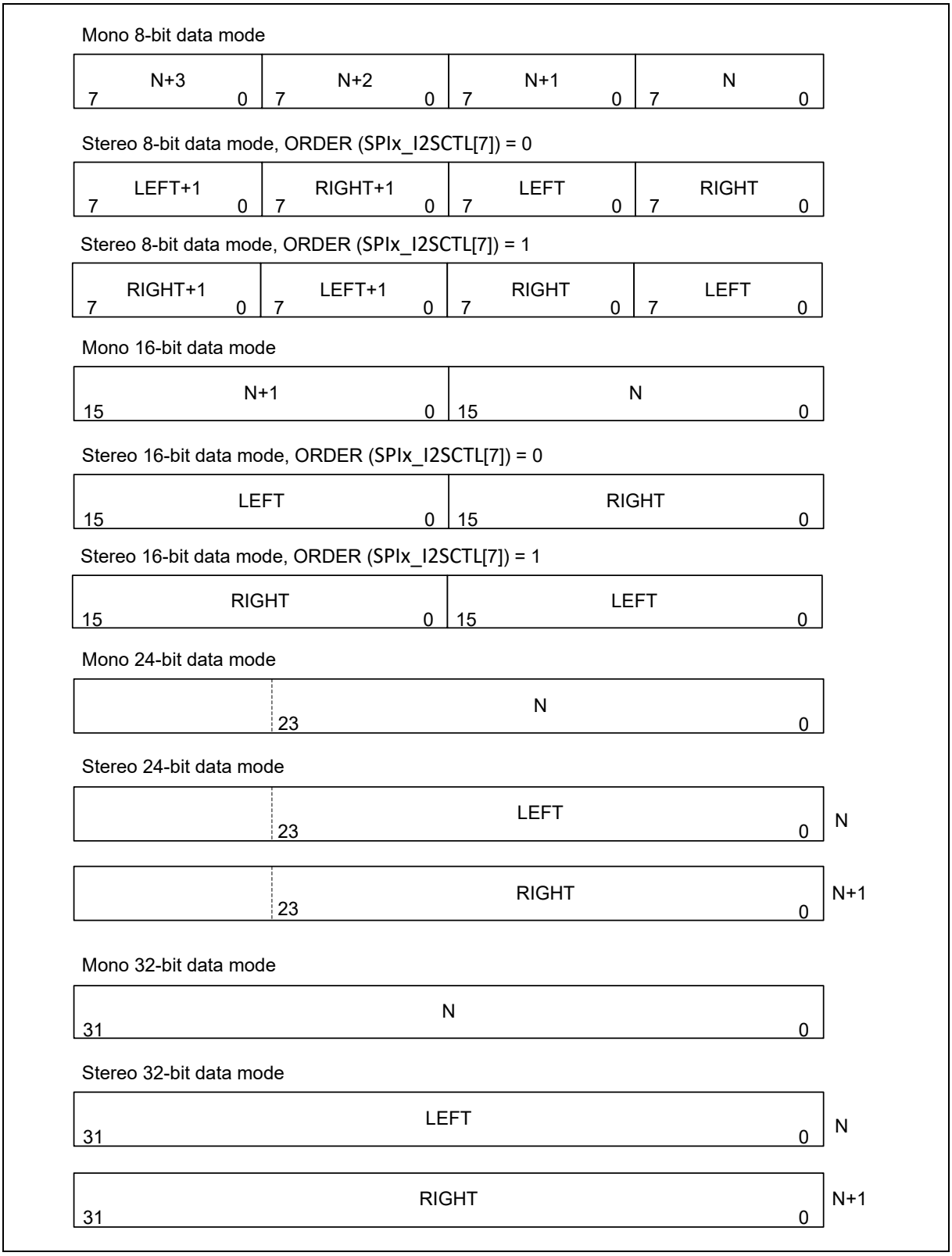


Figure 6.27-24 FIFO Contents for Various I²S Modes

6.27.5.12 Dummy Data Number for I²S / PCM Master mode and Monaural Mode

Before I²S / PCM master starts to send TX data to external slave device, we set control registers I2SEN (SPIx_I2SCTL[0]) enable, TXEN (SPIx_I2SCTL[1]) enable, and write TX data to TX FIFO. After master sends dummy data (data with zero value) to external slave device, master will send TX FIFO data to external slave device. Table 6.27-2 shows number of dummy data for I²S / PCM master monaural mode, and the unit of dummy data number is L channel + R channel.

Data Width (SPIx_I2SCTL[5:4])	Dummy Data Number (Unit = L Channel + R Channel)
8 bits	0
16 bits	0
24 bits	1
32 bits	1

Table 6.27-2 Dummy Data Number for I²S / PCM Master Mode and Monaural Mode

6.27.5.13 Dummy Data Number for I²S / PCM Master Mode and Stereo Mode

Before I²S / PCM master starts to send TX data to external slave device, we set control registers I2SEN (SPIx_I2SCTL[0]) enable, TXEN (SPIx_I2SCTL[1]) enable, and write TX data to TX FIFO. After master sends dummy data (data with zero value) to external slave device, master will send TX FIFO data to external slave device. Table 6.27-3 shows number of dummy data for I²S / PCM master stereo mode, and the unit of dummy data number is L channel + R channel.

Data Width (SPIx_I2SCTL[5:4])	Dummy Data Number (Unit = L Channel + R Channel)
8 bits	0
16 bits	0
24 bits	1
32 bits	1

Table 6.27-3 Dummy Data Number for I²S / PCM Master Mode and Stereo Mode

6.27.5.14 Dummy Data Number for I²S Slave Mode and Monaural Mode

Before I²S slave starts to send TX data to external master device, the I²S control registers I2SEN (SPIx_I2SCTL[0]) and TXEN (SPIx_I2SCTL[1]) should be enabled, meanwhile, the register SLAVE (SPIx_I2SCTL[8]) must be configured with SLAVE mode. Then I²S slave writes TX data to TX FIFO. After slave sends dummy data (data with zero value) to external master device, slave will send TX FIFO data to external master device. Table 6.27-4 shows number of dummy data for I²S slave monaural mode, and the unit of dummy data number is L channel + R channel.

Data Width (SPIx_I2SCTL[5:4])	Dummy Data Number (Unit = L Channel + R Channel)
8 bits	3
16 bits	2
24 bits	2
32 bits	2

Table 6.27-4 Dummy Data Number for I²S Slave Mode and Monaural Mode

6.27.5.15 Dummy Data Number for PCM Slave Mode and Monaural Mode

Before PCM slave starts to send TX data to external master device, the PCM control registers I2SEN

(SPIx_I2SCTL[0]) and TXEN (SPIx_I2SCTL[1]) should be enabled, meanwhile, the register SLAVE (SPIx_I2SCTL[8]) must be configured with SLAVE mode. Then PCM slave writes TX data to TX FIFO. After slave sends dummy data (data with zero value) to external master device, slave will send TX FIFO data to external master device. Table 6.27-5 shows number of dummy data for PCM slave monaural mode, and the unit of dummy data number is L channel + R channel.

Data Width (SPIx_I2SCTL[5:4])	Dummy Data Number (Unit = L Channel + R Channel)
8 bits	2
16 bits	1
24 bits	1
32 bits	1

Table 6.27-5 Dummy Data Number for PCM Slave Mode and Monaural Mode

6.27.5.16 Dummy Data Number for I²S Slave Mode and Stereo Mode

Before I²S slave starts to send TX data to external master device, the I²S control registers I2SEN (SPIx_I2SCTL[0]) and TXEN (SPIx_I2SCTL[1]) should be enabled, meanwhile, the register SLAVE (SPIx_I2SCTL[8]) must be configured with SLAVE mode. Then I²S slave writes TX data to TX FIFO. After slave sends dummy data (data with zero value) to external master device, slave will send TX FIFO data to external master device. Table 6.27-6 shows number of dummy data for I²S slave stereo mode, and the unit of dummy data number is L channel + R channel.

Data Width (SPIx_I2SCTL[5:4])	Dummy Data Number (Unit = L Channel + R Channel)
8 bits	3
16 bits	2
24 bits	2
32 bits	2

Table 6.27-6 Dummy Data Number for I²S Slave Mode and Stereo Mode

6.27.5.17 Dummy Data Number for PCM Slave Mode and Stereo Mode

Before PCM slave starts to send TX data to external master device, the PCM control registers I2SEN (SPIx_I2SCTL[0]) and TXEN (SPIx_I2SCTL[1]) should be enabled, meanwhile, the register SLAVE (SPIx_I2SCTL[8]) must be configured with SLAVE mode. Then PCM slave writes TX data to TX FIFO. After slave sends dummy data (data with zero value) to external master device, slave will send TX FIFO data to external master device. Table 6.27-7 shows number of dummy data for PCM slave stereo mode, and the unit of dummy data number is L channel + R channel.

Data Width (SPIx_I2SCTL[5:4])	Dummy Data Number (Unit = L Channel + R Channel)
8 bits	2
16 bits	1
24 bits	1
32 bits	1

Table 6.27-7 Dummy Data Number for PCM Slave Mode and Stereo Mode

6.27.6 Timing Diagram

The active state of slave selection signal can be defined by setting the SSACTPOL (SPIx_SSCTL[2]). The SPI clock which is in idle state can be configured as high or low state by setting the CLKPOL

(SPIx_CTL[3]). It also provides the bit length of a transaction word in DWIDTH (SPIx_CTL[12:8]), and transmitting/receiving data from MSB or LSB first in LSB (SPIx_CTL[13]). User can also select which edge of SPI clock to transmit/receive data in TXNEG/RXNEG (SPIx_CTL[2:1]). Four SPI timing diagrams for master/slave operations and the related settings are shown below.

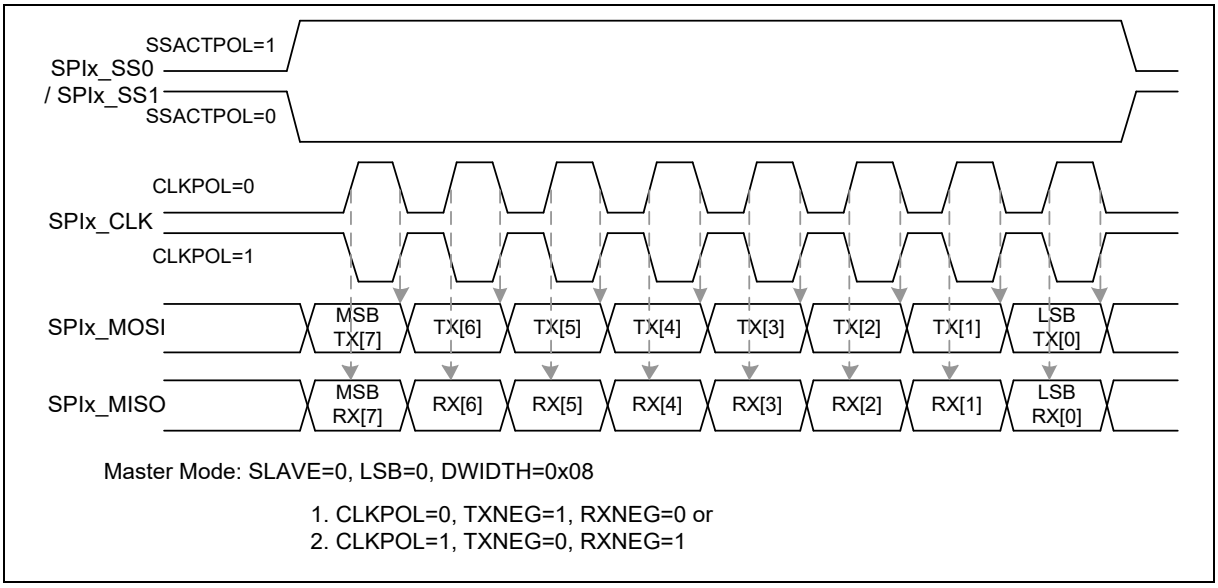


Figure 6.27-25 SPI Timing in Master Mode

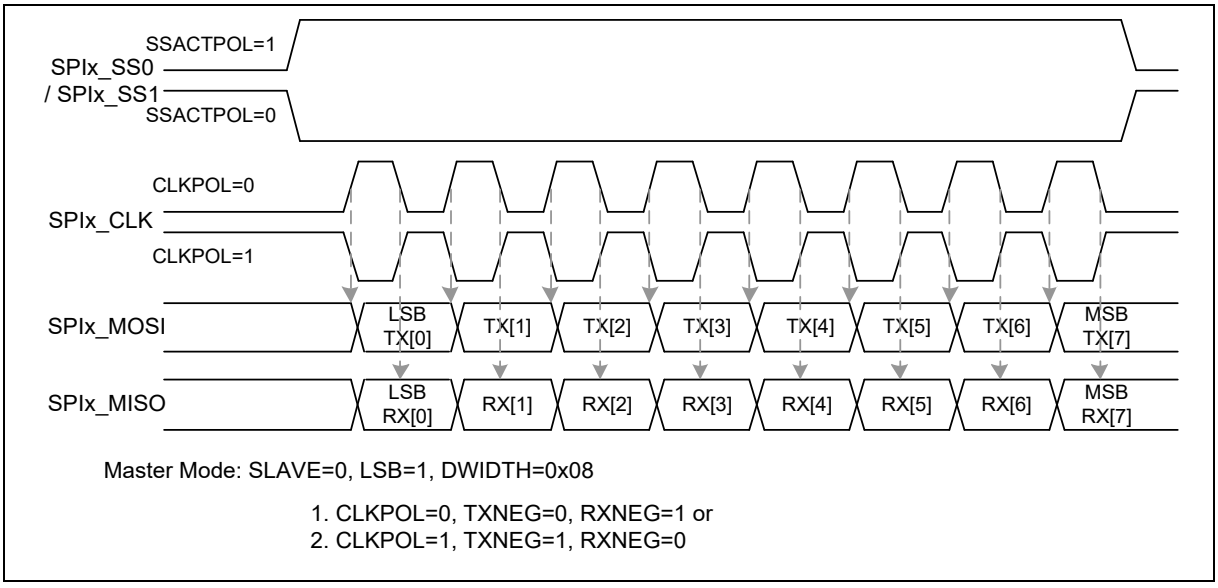


Figure 6.27-26 SPI Timing in Master Mode (Alternate Phase of SPIx_CLK)

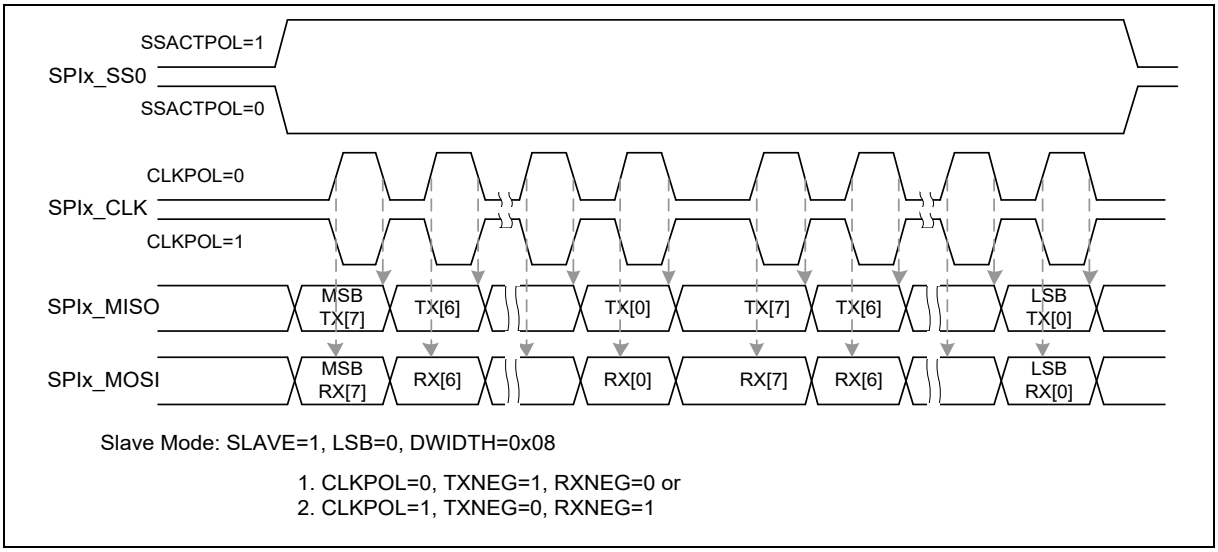


Figure 6.27-27 SPI Timing in Slave Mode

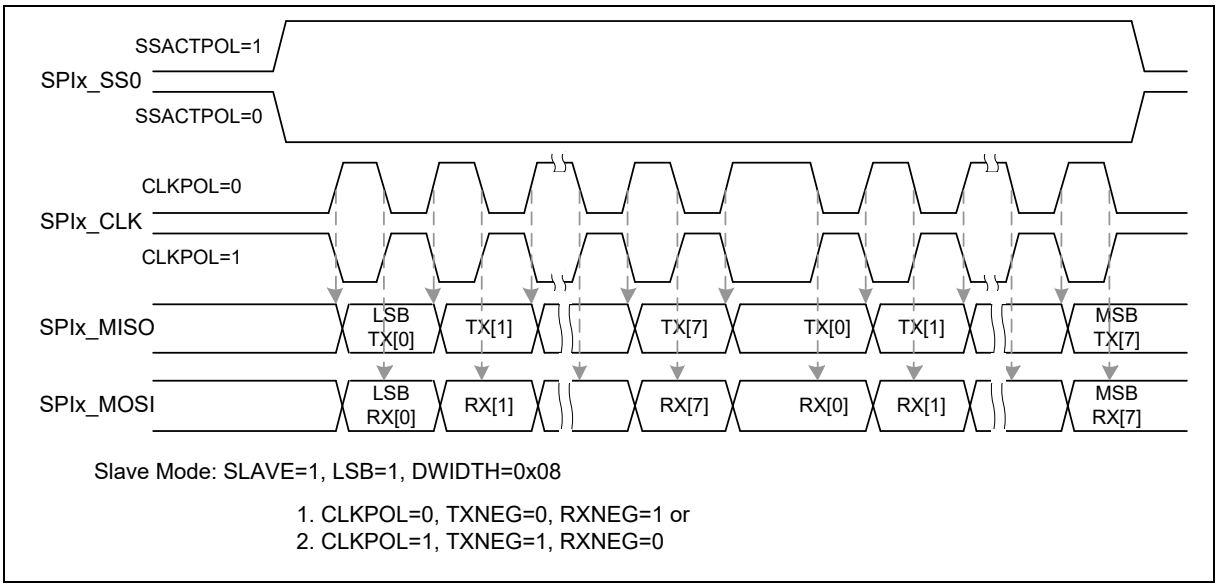


Figure 6.27-28 SPI Timing in Slave Mode (Alternate Phase of SPIx_CLK)

6.27.7 Programming Examples

Example 1:

The SPI controller is set as a full-duplex master to access an off-chip slave device with the following specifications:

- Data bit is latched on positive edge of SPI bus clock.
- Data bit is driven on negative edge of SPI bus clock.
- Data is transferred from MSB first.
- SPI bus clock is idle at low state.
- Only one byte of data to be transmitted/received in a transaction.
- Uses the first SPI slave select pin to connect with an off-chip slave device. The slave

selection signal is active low.

The operation flow is as follows:

1. Set DIVIDER (SPIx_CLKDIV [8:0]) to determine the output frequency of SPI clock.
2. Write the SPIx_SSCTL register a proper value for the related settings of Master mode:
   1) Clear AUTOSS (SPIx_SSCTL[3]) to 0 to disable the Automatic Slave Selection function.
   2) Configure slave selection signal as active low by clearing SSACTPOL (SPIx_SSCTL[2]) to 0.
   3) Enable slave selection signal by setting SS0 (SPIx_SSCTL[0]) or SS1 (SPIx_SSCTL[1]) to 1 to activate the off-chip slave device.
3. Write the related settings into the SPIx_CTL register to control the SPI master actions.
   1) Configure this SPI controller as master device by setting SLAVE (SPIx_CTL[18]) to 0.
   2) Force the SPI clock idle state at low by clearing CLKPOL (SPIx_CTL[3]) to 0.
   3) Select data transmitted on negative edge of SPI bus clock by setting TXNEG (SPIx_CTL[2]) to 1.
   4) Select data latched on positive edge of SPI bus clock by clearing RXNEG (SPIx_CTL[1]) to 0.
   5) Set the bit length of a transaction as 8-bit in DWIDTH bit field (SPIx_CTL[12:8] = 0x08).
   6) Set MSB transfer first by clearing LSB (SPIx_CTL[13]) to 0.
4. Set SPIEN (SPIx_CTL[0]) to 1 to enable the data transfer with the SPI interface.
5. If this SPI master attempts to transmit (write) one byte data to the off-chip slave device, write the byte data that will be transmitted into the SPIx_TX register.
6. Waiting for SPI interrupt if the UNITIEN (SPIx_CTL[17]) is set to 1, or just polling the unit transfer interrupt flag UNITIF (SPIx_STATUS[1]).
7. Read out the received one byte data from SPIx_RX register.
8. Go to Step 5) to continue another data transfer or set SS0 (SPIx_SSCTL[0]) or SS1 (SPIx_SSCTL[1]) to 0 to inactivate the off-chip slave device.

Example 2:

The SPI controller is set as a full-duplex slave device and connects with an off-chip master device. The off-chip master device communicates with the on-chip SPI slave controller through the SPI interface with the following specifications:

- Data bit is latched on positive edge of SPI bus clock.
- Data bit is driven on negative edge of SPI bus clock.
- Data is transferred from LSB first.
- SPI bus clock is idle at high state.
- Only one byte of data to be transmitted/received in a transaction.
- Slave selection signal is active high.

The operation flow is as follows:

1. Write the SPIx_SSCTL register a proper value for the related settings of Slave mode.
2. Select high level for the input of slave selection signal by setting SSACTPOL (SPIx_SSCTL[2]) to 1.
3. Write the related settings into the SPIx_CTL register to control this SPI slave actions

1) Set the SPI controller as slave device by setting SLAVE (SPIx_CTL[18]) to 1.
2) Select the SPI clock idle state at high by setting CLKPOL (SPIx_CTL[3]) to 1.
3) Select data transmitted on negative edge of SPI bus clock by setting TXNEG (SPIx_CTL[2]) to 1.
4) Select data latched on positive edge of SPI bus clock by clearing RXNEG (SPIx_CTL[1]) to 0.
5) Set the bit length of a transaction as 8-bit in DWIDTH bit field (SPIx_CTL[12:8] = 0x08).

4. Set LSB transfer first by setting LSB (SPIx_CTL[13]) to 1.
5. Set the SPIEN (SPIx_CTL[0]) to 1. Wait for the slave select trigger input and SPI clock input from the off-chip master device to start the data transfer.
6. If this SPI slave attempts to transmit (be read) one byte data to the off-chip master device, write the byte data that will be transmitted into the SPIx_TX register.
7. If this SPI slave just only attempts to receive (be written) one byte data from the off-chip master device and does not care what data will be transmitted, the SPIx_TX register does not need to be updated by software.
8. Waiting for SPI interrupt if the UNITIEN (SPIx_CTL[17]) is set to 1, or just polling the unit transfer interrupt flag UNITIF (SPIx_STATUS[1]).
9. Read out the received one byte data from SPIx_RX register.
10. Go to Step 7 to continue another data transfer or stop data transfer.

6.27.8 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
SPI Base Address: **SPIx_BA = 0x4060_0000 + (0x0001_0000 * x)** **x=0, 1, 2, 3**				
SPIx_CTL	SPIx_BA+0x00	R/W	SPI Control Register	0x0000_0034
SPIx_CLKDIV	SPIx_BA+0x04	R/W	SPI Clock Divider Register	0x0000_0000
SPIx_SSCTL	SPIx_BA+0x08	R/W	SPI Slave Select Control Register	0x0000_0000
SPIx_PDMACTL	SPIx_BA+0x0C	R/W	SPI PDMA Control Register	0x0000_0000
SPIx_FIFOCTL	SPIx_BA+0x10	R/W	SPI FIFO Control Register	0x4400_0000
SPIx_STATUS	SPIx_BA+0x14	R/W	SPI Status Register	0x0005_0110
SPIx_STATUS2	SPIx_BA+0x18	R	SPI Status2 Register	0x0000_0000
SPIx_FIFOCTL2	SPIx_BA+0x1C	R/W	SPI FIFO2 Control Register	0x0404_0000
SPIx_TX	SPIx_BA+0x20	W	SPI Data Transmit Register	0x0000_0000
SPIx_RX	SPIx_BA+0x30	R	SPI Data Receive Register	0x0000_0000
SPIx_I2SCTL	SPIx_BA+0x60	R/W	I^2S Control Register	0x0000_0000
SPIx_I2SCLK	SPIx_BA+0x64	R/W	I^2S Clock Divider Control Register	0x0000_0000
SPIx_I2SSTS	SPIx_BA+0x68	R/W	I^2S Status Register	0x0005_0100

6.27.9 Register Description

SPI Control Register (SPIx_CTL)

Register	Offset	R/W	Description	Reset Value
SPIx_CTL	SPIx_BA+0x00	R/W	SPI Control Register	0x0000_0034

Note: Not supported in I^2S mode.

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			DATDIR	REORDER	SLAVE	UNITIEN	Reserved
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RXONLY	HALFDPX	LSB	DWIDTH				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SUSPITV				CLKPOL	TXNEG	RXNEG	SPIEN

Bits	Description	
[31:21]	**Reserved**	Reserved.
[20]	**DATDIR**	**Data Port Direction Control** This bit is used to select the data input/output direction in half-duplex transfer and Dual/Quad transfer 0 = SPI data is input direction. 1 = SPI data is output direction.
[19]	**REORDER**	**Byte Reorder Function Enable Bit** 0 = Byte Reorder function Disabled. 1 = Byte Reorder function Enabled. A byte suspend interval will be inserted among each byte. The period of the byte suspend interval depends on the setting of SUSPITV. **Note:** Byte Reorder function is only available if DWIDTH is defined as 16, 24, and 32 bits.
[18]	**SLAVE**	**Slave Mode Control** 0 = Master mode. 1 = Slave mode.
[17]	**UNITIEN**	**Unit Transfer Interrupt Enable Bit** 0 = SPI unit transfer interrupt Disabled. 1 = SPI unit transfer interrupt Enabled.
[16]	**Reserved**	Reserved.
[15]	**RXONLY**	**Receive-only Mode Enable Bit for Master Only** This bit field is only available in Master mode. In receive-only mode, SPI Master will generate SPI bus clock continuously for receiving data bit from SPI slave device and assert the BUSY status. 0 = Receive-only mode Disabled. 1 = Receive-only mode Enabled.
[14]	**HALFDPX**	**SPI Half-duplex Transfer Enable Bit** This bit is used to select full-duplex or half-duplex for SPI transfer. The bit field DATDIR (SPIx_CTL[20]) can be used to set the data direction in half-duplex transfer. 0 = SPI operates in full-duplex transfer Disabled.

		1 = SPI operates in half-duplex transfer Enabled.
[13]	**LSB**	**Send LSB First** 0 = The MSB, which bit of transmit/receive register depends on the setting of DWIDTH, is transmitted/received first. 1 = The LSB, bit 0 of the SPI TX register, is sent first to the SPI data output pin, and the first bit received from the SPI data input pin will be put in the LSB position of the RX register (bit 0 of SPI_RX).
[12:8]	**DWIDTH**	**Data Width** This field specifies how many bits can be transmitted / received in one transaction. The minimum bit length is 8 bits and can up to 32 bits. DWIDTH = 0x08 …. 8 bits. DWIDTH = 0x09 …. 9 bits. …… DWIDTH = 0x1F …. 31 bits. DWIDTH = 0x00 …. 32 bits. **Note 1:** This bit field will decide the depth of TX/RX FIFO configuration in SPI mode**.** Therefore, changing this bit field will clear TX/RX FIFO by hardware automatically. **Note 2:** When user changes data width, the clear functions of TXFIFO and RXFIFO will be triggered by hardware automatically. User can read status register FIFOCLR (SPI_STATUS[22]) to check status of clear function of TXFIFO and RXFIFO. After status register FIFOCLR is 0, user can write TXFIFO or read RXFIFO correctly.
[7:4]	**SUSPITV**	**Suspend Interval for Master Only** The four bits provide configurable suspend interval between two successive transmit/receive transaction in a transfer. The definition of the suspend interval is the interval between the last clock edge of the preceding transaction word and the first clock edge of the following transaction word. The default value is 0x3. The period of the suspend interval is obtained according to the following equation. (SUSPITV[3:0] + 0.5) * period of SPI_CLK clock cycle Example: SUSPITV = 0x0 …. 0.5 SPI_CLK clock cycle. SUSPITV = 0x1 …. 1.5 SPI_CLK clock cycle. …… SUSPITV = 0xE …. 14.5 SPI_CLK clock cycle. SUSPITV = 0xF …. 15.5 SPI_CLK clock cycle.
[3]	**CLKPOL**	**Clock Polarity** 0 = SPI bus clock is idle low. 1 = SPI bus clock is idle high.
[2]	**TXNEG**	**Transmit on Negative Edge** 0 = Transmitted data output signal is changed on the rising edge of SPI bus clock. 1 = Transmitted data output signal is changed on the falling edge of SPI bus clock.
[1]	**RXNEG**	**Receive on Negative Edge** 0 = Received data input signal is latched on the rising edge of SPI bus clock. 1 = Received data input signal is latched on the falling edge of SPI bus clock.
[0]	**SPIEN**	**SPI Transfer Control Enable Bit** In Master mode, the transfer will start when there is data in the FIFO buffer after this bit is set to 1. In Slave mode, this device is ready to receive data when this bit is set to 1. 0 = Transfer control Disabled. 1 = Transfer control Enabled. **Note:** Before changing the configurations of SPIx_CTL, SPIx_CLKDIV, SPIx_SSCTL and SPIx_FIFOCTL, registers, user shall clear the SPIEN (SPIx_CTL[0]) and confirm the SPIENSTS (SPIx_STATUS[15]) is 0.

SPI Clock Divider Register (SPIx_CLKDIV)

Register	Offset	R/W	Description	Reset Value
SPIx_CLKDIV	SPIx_BA+0x04	R/W	SPI Clock Divider Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							DIVIDER
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DIVIDER							

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8:0]	**DIVIDER**	**Clock Divider** The value in this field is the frequency divider for generating the peripheral clock, f_{spi_eclk}, and the SPI bus clock of SPI Master. The frequency is obtained according to the following equation. $f_{spi_eclk} = \frac{f_{spi_clock_src}}{(DIVIDER+1)}$ where $f_{spi_clock_src}$ is the peripheral clock source, which is defined in the clock control register, CLK_CLKSEL4. **Note 1:** Not supported in I^2S mode. **Note 2:** The time interval must be larger than or equal 5 peripheral clock cycles between releasing SPI IP software reset and setting this clock divider register. **Note 3:** The setting values of this clock divider "DIVIDER" are only 1, 3, 5, 7,…, 2*N+1, which variable N equals to 0,1,2,3,4,…

Note: DIVIDER should be set carefully because the peripheral clock frequency must be slower than or equal to system frequency.

SPI Slave Select Control Register (SPIx_SSCTL)

Register	Offset	R/W	Description	Reset Value
SPIx_SSCTL	SPIx_BA+0x08	R/W	SPI Slave Select Control Register	0x0000_0000

Note: Not supported in I^2S mode.

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		SSINAIEN	SSACTIEN	Reserved		SLVURIEN	SLVBEIEN
7	6	5	4	3	2	1	0
Reserved			SLV3WIRE	AUTOSS	SSACTPOL	SS1	SS0

Bits	Description	
[31:14]	**Reserved**	Reserved.
[13]	**SSINAIEN**	**Slave Select Inactive Interrupt Enable Bit** 0 = Slave select inactive interrupt Disabled. 1 = Slave select inactive interrupt Enabled.
[12]	**SSACTIEN**	**Slave Select Active Interrupt Enable Bit** 0 = Slave select active interrupt Disabled. 1 = Slave select active interrupt Enabled.
[11:10]	**Reserved**	Reserved.
[9]	**SLVURIEN**	**Slave Mode TX Under Run Interrupt Enable Bit** 0 = Slave mode TX under run interrupt Disabled. 1 = Slave mode TX under run interrupt Enabled.
[8]	**SLVBEIEN**	**Slave Mode Bit Count Error Interrupt Enable Bit** 0 = Slave mode bit count error interrupt Disabled. 1 = Slave mode bit count error interrupt Enabled.
[7:5]	**Reserved**	Reserved.
[4]	**SLV3WIRE**	**Slave 3-wire Mode Enable Bit for SPI Slave Mode Only** In Slave 3-wire mode, the SPI controller can work with 3-wire interface including SPIx_CLK, SPIx_MISO and SPIx_MOSI pins. 0 = 4-wire bi-direction interface Disabled. 1 = 3-wire bi-direction interface Enabled. **Note:** The value of this register equals to control register SLAVE (SPIx_I2SCTL[8]) when I^2S mode is enabled.
[3]	**AUTOSS**	**Automatic Slave Selection Function Enable Bit for Master Only** 0 = Automatic slave selection function Disabled. Slave selection signal will be asserted/de-asserted according to SS0 (SPIx_SSCTL[0]) or SS1 (SPIx_SSCTL[1]). 1 = Automatic slave selection function Enabled.

[2]	**SSACTPOL**	**Slave Selection Active Polarity** This bit defines the active polarity of slave selection signal (SPIx_SS0 and SPIx_SS1). 0 = The slave selection signal SPIx_SS0/ SPIx_SS1 is active low. 1 = The slave selection signal SPIx_SS0/ SPIx_SS1 is active high.
[1]	**SS1**	**Slave Selection Control 1 for Master Only** If AUTOSS bit is cleared to 0, 0 = Set the SPIx_SS1 line to inactive state. 1 = Set the SPIx_SS1 line to active state. If the AUTOSS bit is set to 1, 0 = Keep the SPIx_SS1 line at inactive state. 1 = SPIx_SS1 line will be automatically driven to active state for the duration of data transfer, and will be driven to inactive state for the rest of the time. The active state of SPIx_SS1 is specified in SSACTPOL (SPIx_SSCTL[2]).
[0]	**SS0**	**Slave Selection Control 0 for Master Only** If AUTOSS bit is cleared to 0, 0 = Set the SPIx_SS0 line to inactive state. 1 = Set the SPIx_SS0 line to active state. If the AUTOSS bit is set to 1, 0 = Keep the SPIx_SS0 line at inactive state. 1 = SPIx_SS0 line will be automatically driven to active state for the duration of data transfer, and will be driven to inactive state for the rest of the time. The active state of SPIx_SS0 is specified in SSACTPOL (SPIx_SSCTL[2]).

SPI PDMA Control Register (SPIx_PDMACTL)

Register	Offset	R/W	Description	Reset Value
SPIx_PDMACTL	SPIx_BA+0x0C	R/W	SPI PDMA Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					PDMARST	RXPDMAEN	TXPDMAEN

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**PDMARST**	**PDMA Reset** 0 = No effect. 1 = Reset the PDMA control logic of the SPI controller. This bit will be automatically cleared to 0.
[1]	**RXPDMAEN**	**Receive PDMA Enable Bit** 0 = Receive PDMA function Disabled. 1 = Receive PDMA function Enabled.
[0]	**TXPDMAEN**	**Transmit PDMA Enable Bit** 0 = Transmit PDMA function Disabled. 1 = Transmit PDMA function Enabled. **Note 1:** In SPI Master mode with full duplex transfer, if both TX and RX PDMA functions are enabled, RX PDMA function cannot be enabled prior to TX PDMA function. User can enable TX PDMA function firstly or enable both functions simultaneously. **Note 2:** In SPI Master mode with full duplex transfer, if both TX and RX PDMA functions are enabled, TX PDMA function cannot be disabled prior to RX PDMA function. User can disable RX PDMA function firstly or disable both functions simultaneously.

SPI FIFO Control Register (SPIx_FIFOCTL)

Register	Offset	R/W	Description	Reset Value
SPIx_FIFOCTL	SPIx_BA+0x10	R/W	SPI FIFO Control Register	0x4400_0000

31	30	29	28	27	26	25	24
TXTH				RXTH			
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved					SLVBERX	TXFBCLR	RXFBCLR
7	6	5	4	3	2	1	0
TXUFIEN	TXUFPOL	RXOVIEN	RXTOIEN	TXTHIEN	RXTHIEN	TXRST	RXRST

Bits	Description	
[31:28]	**TXTH**	**Transmit FIFO Threshold** If the valid data count of the transmit FIFO buffer is less than or equal to the TXTH setting, the TXTHIF bit will be set to 1, else the TXTHIF bit will be cleared to 0. The MSB of this bit field is only meaningful while SPI mode 8~16 bits of data length.
[27:24]	**RXTH**	**Receive FIFO Threshold** If the valid data count of the receive FIFO buffer is larger than the RXTH setting, the RXTHIF bit will be set to 1, else the RXTHIF bit will be cleared to 0. The MSB of this bit field is only meaningful while SPI mode 8~16 bits of data length.
[23:11]	**Reserved**	Reserved.
[10]	**SLVBERX**	**RX FIFO Write Data Enable Bit When Slave Mode Bit Count Error for SPI Slave Mode Only** 0 = Uncompleted RX data will be dropped from RX FIFO when bit count error event happen in SPI slave mode. 1 = Uncompleted RX data will be written into RX FIFO when bit count error event happen in SPI slave mode. User can read SLVBENUM (SPIx_STATUS2[29:24]) to know that the effective bit number of uncompleted RX data when SPI slave bit count error happened.
[9]	**TXFBCLR**	**Transmit FIFO Buffer Clear** 0 = No effect. 1 = Clear transmit FIFO pointer. The TXFULL bit will be cleared to 0 and the TXEMPTY bit will be set to 1. This bit will be cleared to 0 by hardware about 1 system clock after it is set to 1. **Note:** The TX shift register will not be cleared.
[8]	**RXFBCLR**	**Receive FIFO Buffer Clear** 0 = No effect. 1 = Clear receive FIFO pointer. The RXFULL bit will be cleared to 0 and the RXEMPTY bit will be set to 1. This bit will be cleared to 0 by hardware about 1 system clock after it is set to 1. **Note:** The RX shift register will not be cleared.
[7]	**TXUFIEN**	**TX Underflow Interrupt Enable Bit** When TX underflow event occurs in Slave mode, TXUFIF (SPIx_STATUS[19]) will be set to 1. This bit is used to enable the TX underflow interrupt. 0 = Slave TX underflow interrupt Disabled.

		1 = Slave TX underflow interrupt Enabled.
[6]	**TXUFPOL**	**TX Underflow Data Polarity** 0 = The SPI data out is keep 0 if there is TX underflow event in Slave mode. 1 = The SPI data out is keep 1 if there is TX underflow event in Slave mode. **Note 1:** The TX underflow event occurs if there is no any data in TX FIFO when the slave selection signal is active. **Note 2:** This bit should be set as 0 in I^2S mode. **Note 3:** When TX underflow event occurs, SPIx_MISO pin state will be determined by this setting even though TX FIFO is not empty afterward. Data stored in TX FIFO will be sent through SPIx_MISO pin in the next transfer frame.
[5]	**RXOVIEN**	**Receive FIFO Overrun Interrupt Enable Bit** 0 = Receive FIFO overrun interrupt Disabled. 1 = Receive FIFO overrun interrupt Enabled.
[4]	**RXTOIEN**	**Slave Receive Time-out Interrupt Enable Bit** 0 = Receive time-out interrupt Disabled. 1 = Receive time-out interrupt Enabled.
[3]	**TXTHIEN**	**Transmit FIFO Threshold Interrupt Enable Bit** 0 = TX FIFO threshold interrupt Disabled. 1 = TX FIFO threshold interrupt Enabled.
[2]	**RXTHIEN**	**Receive FIFO Threshold Interrupt Enable Bit** 0 = RX FIFO threshold interrupt Disabled. 1 = RX FIFO threshold interrupt Enabled.
[1]	**TXRST**	**Transmit Reset** 0 = No effect. 1 = Reset transmit FIFO pointer and transmit circuit. The TXFULL bit will be cleared to 0 and the TXEMPTY bit will be set to 1. This bit will be cleared to 0 by hardware about 3 system clock cycles + 2 peripheral clock cycles after it is set to 1. User can read TXRXRST (SPIx_STATUS[23]) to check if reset is accomplished or not. **Note:** If TX underflow event occurs in SPI Slave mode, this bit can be used to make SPI return to idle state.
[0]	**RXRST**	**Receive Reset** 0 = No effect. 1 = Reset receive FIFO pointer and receive circuit. The RXFULL bit will be cleared to 0 and the RXEMPTY bit will be set to 1. This bit will be cleared to 0 by hardware about 3 system clock cycles + 2 peripheral clock cycles after it is set to 1. User can read TXRXRST (SPIx_STATUS[23]) to check if reset is accomplished or not.

SPI Status Register (SPIx_STATUS)

Register	Offset	R/W	Description	Reset Value
SPIx_STATUS	SPIx_BA+0x14	R/W	SPI Status Register	0x0005_0110

Note: Not supported in I^2S mode.

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
TXRXRST	FIFOCLR	Reserved		TXUFIF	TXTHIF	TXFULL	TXEMPTY
15	14	13	12	11	10	9	8
SPIENSTS	Reserved		RXTOIF	RXOVIF	RXTHIF	RXFULL	RXEMPTY
7	6	5	4	3	2	1	0
SLVURIF	SLVBEIF	Reserved	SSLINE	SSINAIF	SSACTIF	UNITIF	BUSY

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23]	**TXRXRST**	**TX or RX Reset Status (Read Only)** 0 = The reset function of TXRST or RXRST is done. 1 = Doing the reset function of TXRST or RXRST. **Note:** Both the reset operations of TXRST and RXRST need 3 system clock cycles + 2 peripheral clock cycles. User can check the status of this bit to monitor the reset function is doing or done.
[22]	**FIFOCLR**	**TXFIFO or RXFIFO Clear Status (Read Only)** User can check the status of this bit to monitor the clear function of TXFIFO or RXFIFO is doing or done. 0 = The clear function of TXFBCLR or RXFBCLR is done. 1 = Doing the clear function of TXFBCLR or RXFBCLR.
[21:20]	**Reserved**	Reserved.
[19]	**TXUFIF**	**TX Underflow Interrupt Flag** When the TX underflow event occurs, this bit will be set to 1, the state of data output pin depends on the setting of TXUFPOL. 0 = No effect. 1 = No data in Transmit FIFO and TX shift register when the slave selection signal is active. **Note 1:** This bit will be cleared by writing 1 to it. **Note 2:** If reset slave's transmission circuit when slave selection signal is active, this flag will be set to 1 after 2 peripheral clock cycles + 3 system clock cycles since the reset operation is done.
[18]	**TXTHIF**	**Transmit FIFO Threshold Interrupt Flag (Read Only)** 0 = The valid data count within the transmit FIFO buffer is larger than the setting value of TXTH. 1 = The valid data count within the transmit FIFO buffer is less than or equal to the setting value of TXTH.
[17]	**TXFULL**	**Transmit FIFO Buffer Full Indicator (Read Only)** 0 = Transmit FIFO buffer is not full. 1 = Transmit FIFO buffer is full. **Note:** TXFULL flag does not go high when TXPDMAEN is set to 1.
[16]	**TXEMPTY**	**Transmit FIFO Buffer Empty Indicator (Read Only)**

		0 = Transmit FIFO buffer is not empty. 1 = Transmit FIFO buffer is empty.
[15]	**SPIENSTS**	**SPI Enable Status (Read Only)** 0 = SPI controller Disabled. 1 = SPI controller Enabled. **Note:** The SPI peripheral clock is asynchronous with the system clock. In order to make sure the SPI control logic is disabled, this bit indicates the real status of SPI controller.
[14:13]	**Reserved**	Reserved.
[12]	**RXTOIF**	**Receive Time-out Interrupt Flag** 0 = No receive FIFO time-out event. 1 = Receive FIFO buffer is not empty and no read operation on receive FIFO buffer over 64 SPI peripheral clock periods in Master mode or over 576 SPI peripheral clock periods in Slave mode. When the received FIFO buffer is read by software, the time-out status will be cleared automatically. **Note:** This bit will be cleared by writing 1 to it.
[11]	**RXOVIF**	**Receive FIFO Overrun Interrupt Flag** When the receive FIFO buffer is full, the follow-up data will be dropped and this bit will be set to 1. 0 = No FIFO is overrun. 1 = Receive FIFO is overrun. **Note:** This bit will be cleared by writing 1 to it.
[10]	**RXTHIF**	**Receive FIFO Threshold Interrupt Flag (Read Only)** 0 = The valid data count within the receive FIFO buffer is smaller than or equal to the setting value of RXTH. 1 = The valid data count within the receive FIFO buffer is larger than the setting value of RXTH.
[9]	**RXFULL**	**Receive FIFO Buffer Full Indicator (Read Only)** 0 = Receive FIFO buffer is not full. 1 = Receive FIFO buffer is full.
[8]	**RXEMPTY**	**Receive FIFO Buffer Empty Indicator (Read Only)** 0 = Receive FIFO buffer is not empty. 1 = Receive FIFO buffer is empty.
[7]	**SLVURIF**	**Slave Mode TX Under Run Interrupt Flag** In Slave mode, if TX underflow event occurs and the slave select line goes to inactive state, this interrupt flag will be set to 1. 0 = No Slave TX under run event. 1 = Slave TX under run event occurred. **Note:** This bit will be cleared by writing 1 to it.
[6]	**SLVBEIF**	**Slave Mode Bit Count Error Interrupt Flag** In Slave mode, when the slave select line goes to inactive state, if bit counter is mismatched with DWIDTH, this interrupt flag will be set to 1. 0 = No Slave mode bit count error event. 1 = Slave mode bit count error event occurred. **Note:** If the slave select active but there is no any bus clock input, the SLVBEIF also active when the slave select goes to inactive state. This bit will be cleared by writing 1 to it.
[5]	**Reserved**	Reserved.
[4]	**SSLINE**	**Slave Select Line Bus Status (Read Only)** 0 = The slave select line status is 0. 1 = The slave select line status is 1. **Note:** This bit is only available in Slave mode. If SSACTPOL (SPIx_SSCTL[2]) is set 0, and the SSLINE is

		1, the SPI slave select is in inactive status.
[3]	**SSINAIF**	**Slave Select Inactive Interrupt Flag** 0 = Slave select inactive interrupt was cleared or not occurred. 1 = Slave select inactive interrupt event occurred. **Note:** Only available in Slave mode. This bit will be cleared by writing 1 to it.
[2]	**SSACTIF**	**Slave Select Active Interrupt Flag** 0 = Slave select active interrupt was cleared or not occurred. 1 = Slave select active interrupt event occurred. **Note:** Only available in Slave mode. This bit will be cleared by writing 1 to it.
[1]	**UNITIF**	**Unit Transfer Interrupt Flag** 0 = No transaction has been finished since this bit was cleared to 0. 1 = SPI controller has finished one unit transfer. **Note:** This bit will be cleared by writing 1 to it.
[0]	**BUSY**	**Busy Status (Read Only)** 0 = SPI controller is in idle state. 1 = SPI controller is in busy state. The following lists the bus busy conditions: a. SPIEN (SPIx_CTL[0]) = 1 and TXEMPTY = 0. b. For SPI Master mode, SPIEN (SPIx_CTL[0]) = 1 and TXEMPTY = 1 but the current transaction is not finished yet. c. For SPI Master mode, SPIEN (SPIx_CTL[0]) = 1 and RXONLY = 1. d. For SPI Slave mode, SPIEN (SPIx_CTL[0]) = 1 and there is serial clock input into the SPI core logic when slave select is active. e. For SPI Slave mode, SPIEN (SPIx_CTL[0]) = 1 and the transmit buffer or transmit shift register is not empty even if the slave select is inactive. **Note:** By applications, this SPI bus flag should be used with other status registers in SPIx_STATUS such as TXCNT, RXCNT, TXTHIF, TXFULL, TXEMPTY, RXTHIF, RXFULL, RXEMPTY, and UNITIF. Therefore the SPI transfer done events of TX/RX operations can be obtained at correct timing point.

SPI Status2 Register (SPIx_STATUS2)

Register	Offset	R/W	Description	Reset Value
SPIx_STATUS2	SPIx_BA+0x18	R	SPI Status2 Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		SLVBENUM					
23	22	21	20	19	18	17	16
Reserved		TXCNT					
15	14	13	12	11	10	9	8
Reserved		RXCNT					
7	6	5	4	3	2	1	0
Reserved							

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29:24]	**SLVBENUM**	**Effective Bit Number of Uncompleted RX Data for SPI Slave Mode Only** This status register indicates that effective bit number of uncompleted RX data when SLVBERX (SPIx_FIFOCTL[10]) is enabled and RX bit count error event happen in SPI slave mode. This status register will be fixed to 0x0 when SLVBERX (SPIx_FIFOCTL[10]) is disabled. **Note:** This register will be cleared to 0x0 when user write 0x1 to SLVBEIF (SPIx_STATUS[6]).
[23:22]	**Reserved**	Reserved.
[21:16]	**TXCNT**	**Transmit FIFO Data Count (Read Only)** This bit field indicates the valid data count of transmit FIFO buffer.
[15:14]	**Reserved**	Reserved.
[13:8]	**RXCNT**	**Receive FIFO Data Count (Read Only)** This bit field indicates the valid data count of receive FIFO buffer.
[7:0]	**Reserved**	Reserved.

SPI FIFO2 Control Register (SPIx_FIFOCTL2)

Register	Offset	R/W	Description	Reset Value
SPIx_FIFOCTL2	SPIx_BA+0x1C	R/W	SPI FIFO2 Control Register	0x0404_0000

31	30	29	28	27	26	25	24
Reserved			TXTHPDMA				
23	22	21	20	19	18	17	16
Reserved			RXTHPDMA				
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							

Bits	Description	
[31:29]	**Reserved**	Reserved.
[28:24]	**TXTHPDMA**	**Transmit FIFO Threshold When TX PDMA Enable Only** If the valid data count of the transmit FIFO buffer is less than or equal to the TXTH setting, the TXTHIF bit will be set to 1, else the TXTHIF bit will be cleared to 0.
[23:21]	**Reserved**	Reserved.
[20:16]	**RXTHPDMA**	**Receive FIFO Threshold When RX PDMA Enable Only** If the valid data count of the receive FIFO buffer is larger than the RXTH setting, the RXTHIF bit will be set to 1, else the RXTHIF bit will be cleared to 0.
[15:0]	**Reserved**	Reserved.

SPI Data Transmit Register (SPIx_TX)

Register	Offset	R/W	Description	Reset Value
SPIx_TX	SPIx_BA+0x20	W	SPI Data Transmit Register	0x0000_0000

31	30	29	28	27	26	25	24
TX							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TX							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TX							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TX							

Bits	Description	
[31:0]	**TX**	**Data Transmit Register** The data transmit registers pass through the transmitted data into the 8-level transmit FIFO buffers. The number of valid bits depends on the setting of DWIDTH (SPIx_CTL[12:8]) in SPI mode or WDWIDTH (SPIx_I2SCTL[5:4]) in I^2S mode. In SPI mode, if DWIDTH is set to 0x08, the bits TX[7:0] will be transmitted. If DWIDTH is set to 0x00, the SPI controller will perform a 32-bit transfer. In I^2S mode, if WDWIDTH (SPIx_I2SCTL[5:4]) is set to 0x2, the data width of audio channel is 24-bit and corresponding to TX[23:0]. If WDWIDTH is set as 0x0, 0x1, or 0x3, all bits of this field are valid and referred to the data arrangement in I^2S mode FIFO operation section **Note:** In Master mode, the SPI controller will start to transfer the SPI bus clock after 1 APB clock and 6 peripheral clock cycles after user writes to this register.

SPI Data Receive Register (SPIx_RX)

Register	Offset	R/W	Description	Reset Value
SPIx_RX	SPIx_BA+0x30	R	SPI Data Receive Register	0x0000_0000

31	30	29	28	27	26	25	24
RX							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RX							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RX							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RX							

Bits	Description	
[31:0]	**RX**	**Data Receive Register (Read Only)** There are 8-level FIFO buffers in this controller. The data receive register holds the data received from SPI data input pin. If the RXEMPTY (SPIx_STATUS[8] or SPIx_I2SSTS[8]) is not set to 1, the receive FIFO buffers can be accessed through software by reading this register.

I²S Control Register (SPIx_I2SCTL)

Register	Offset	R/W	Description	Reset Value
SPIx_I2SCTL	SPIx_BA+0x60	R/W	I²S Control Register	0x0000_0000

Note: Not supported in SPI mode.

31	30	29	28	27	26	25	24
SLVERRIEN	Reserved	FORMAT		Reserved		LZCIEN	RZCIEN
23	22	21	20	19	18	17	16
RXLCH	Reserved					LZCEN	RZCEN
15	14	13	12	11	10	9	8
MCLKEN	Reserved						SLAVE
7	6	5	4	3	2	1	0
ORDER	MONO	WDWIDTH		MUTE	RXEN	TXEN	I2SEN

Bits	Description	
[31]	**SLVERRIEN**	**Bit Number Error Interrupt Enable Bit for Slave Mode** Interrupt occurs if this bit is set to 1 and bit number error event occurs. 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[30]	**Reserved**	Reserved.
[29:28]	**FORMAT**	**Data Format Selection** 00 = I²S data format. 01 = MSB justified data format. 10 = PCM mode A. 11 = PCM mode B.
[27:26]	**Reserved**	Reserved.
[25]	**LZCIEN**	**Left Channel Zero Cross Interrupt Enable Bit** Interrupt occurs if this bit is set to 1 and left channel zero cross event occurs. 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[24]	**RZCIEN**	**Right Channel Zero Cross Interrupt Enable Bit** Interrupt occurs if this bit is set to 1 and right channel zero cross event occurs. 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[23]	**RXLCH**	**Receive Left Channel Enable Bit** When monaural format is selected (MONO = 1), I²S controller will receive right channel data if RXLCH is set to 0, and receive left channel data if RXLCH is set to 1. 0 = Receive right channel data in Mono mode. 1 = Receive left channel data in Mono mode.
[22:18]	**Reserved**	Reserved.

[17]	**LZCEN**	**Left Channel Zero Cross Detection Enable Bit** If this bit is set to 1, when left channel data sign bit changes or next shift data bits are all 0 then LZCIF flag in SPIx_I2SSTS register is set to 1. This function is only available in transmit operation. 0 = Left channel zero cross detection Disabled. 1 = Left channel zero cross detection Enabled.
[16]	**RZCEN**	**Right Channel Zero Cross Detection Enable Bit** If this bit is set to 1, when right channel data sign bit change or next shift data bits are all 0 then RZCIF flag in SPIx_I2SSTS register is set to 1. This function is only available in transmit operation. 0 = Right channel zero cross detection Disabled. 1 = Right channel zero cross detection Enabled.
[15]	**MCLKEN**	**Master Clock Enable Bit** If MCLKEN is set to 1, I^2S controller will generate master clock on SPIx_I2SMCLK pin for external audio devices. 0 = Master clock Disabled. 1 = Master clock Enabled.
[14:9]	**Reserved**	Reserved.
[8]	**SLAVE**	**Slave Mode** I^2S can operate as master or slave. For Master mode, I2Sx_BCLK and I2Sx_LRCLK pins are output mode and send bit clock from this chip to audio CODEC chip. In Slave mode, I2Sx_BCLK and I2Sx_LRCLK pins are input mode and I2Sx_BCLK and I2Sx_LRCLK signals are received from outer audio CODEC chip. 0 = Master mode. 1 = Slave mode.
[7]	**ORDER**	**Stereo Data Order in FIFO** 0 = Left channel data at high byte. 1 = Left channel data at low byte.
[6]	**MONO**	**Monaural Data** 0 = Data is stereo format. 1 = Data is monaural format.
[5:4]	**WDWIDTH**	**Word Width** 00 = data size is 8-bit. 01 = data size is 16-bit. 10 = data size is 24-bit. 11 = data size is 32-bit.
[3]	**MUTE**	**Transmit Mute Enable Bit** 0 = Transmit data is shifted from buffer. 1 = Transmit channel zero.
[2]	**RXEN**	**Receive Enable Bit** 0 = Data receive Disabled. 1 = Data receive Enabled.
[1]	**TXEN**	**Transmit Enable Bit** 0 = Data transmit Disabled. 1 = Data transmit Enabled.

[0]	**I2SEN**	**I^2S Controller Enable Bit** 0 = I^2S mode Disabled. 1 = I^2S mode Enabled. **Note 1:** If enabling this bit, I2Sx_BCLK will start to output in Master mode. **Note 2:** Before changing the configurations of SPIx_I2SCTL, SPIx_I2SCLK, SPIx_FIFOCTL, registers, user shall clear the I2SEN (SPIx_I2SCTL[0]) and confirm the I2SENSTS (SPIx_I2SSTS[15]) is 0.

I²S Clock Divider Control Register (SPIx_I2SCLK)

Register	Offset	R/W	Description	Reset Value
SPIx_I2SCLK	SPIx_BA+0x64	R/W	I²S Clock Divider Control Register	0x0000_0000

Note: Not supported in SPI mode.

31	30	29	28	27	26	25	24
Reserved						I2SSLAVE	I2SMODE
23	22	21	20	19	18	17	16
Reserved						BCLKDIV	
15	14	13	12	11	10	9	8
BCLKDIV							
7	6	5	4	3	2	1	0
Reserved	MCLKDIV						

Bits	Description	
[31:26]	**Reserved**	Reserved.
[25]	**I2SSLAVE**	**I²S Clock Divider Number Selection for I²S Slave Mode and I²S Master Mode** User sets I2SSLAVE to set frequency of peripheral clock of I²S master mode and I²S slave mode when BCLKDIV (SPIx_I2SCLK[17:8]) is set. User needs to set I2SSLAVE before I2SEN (SPIx_I2SCTL[0]) is enabled. 0 = The frequency of peripheral clock set to I²S master mode. 1 = The frequency of peripheral clock set to I²S slave mode.
[24]	**I2SMODE**	**I²S Clock Divider Number Selection for I²S Mode and SPI Mode** User sets I2SMODE to set frequency of peripheral clock of I²S mode or SPI mode when BCLKDIV (SPIx_I2SCLK[17:8]) or DIVIDER (SPIx_CLKDIV[8:0]) are set. User needs to set I2SMODE before I2SEN (SPIx_I2SCTL[0]) or SPIEN (SPIx_CTL[0]) is enabled. 0 = The frequency of peripheral clock set to SPI mode. 1 = The frequency of peripheral clock set to I²S mode.
[23:18]	**Reserved**	Reserved.

[17:8]	**BCLKDIV**	**Bit Clock Divider** The I²S controller will generate bit clock in Master mode. The clock frequency of bit clock, f_{BCLK}, is determined by the following expression: $f_{BCLK} = \frac{f_{i2s_clock_src}}{2 \times (\mathrm{BCLKDIV} + 1)}$ where $f_{i2s_clock_src}$ is the frequency of I²S peripheral clock source, which is defined in the clock control register CLK_CLKSEL4. In I²S Slave mode, this field is used to define the frequency of peripheral clock and it's determined by $f_{i2s_clock_src} \div \left(\frac{\mathrm{BCLKDIV}}{2} + 1\right)$. The peripheral clock frequency in I²S Slave mode must be equal to or faster than 6 times of input bit clock. **Note:** The time interval must be larger than or equal 5 peripheral clock cycles between releasing SPI IP software reset and setting this clock divider register.
[7]	**Reserved**	Reserved.
[6:0]	**MCLKDIV**	**Master Clock Divider** If MCLKEN is set to 1, I²S controller will generate master clock for external audio devices. The frequency of master clock, f_{MCLK}, is determined by the following expressions: If MCLKDIV >= 1,. $f_{MCLK} = \frac{f_{i2s_clock_src}}{2 \times \mathrm{MCLKDIV}}$ If MCLKDIV = 0,. $f_{MCLK} = f_{i2s_clock_src}$ where $f_{i2s_clock_src}$ is the frequency of I²S peripheral clock source, which is defined in the clock control register CLK_CLKSEL4. In general, the master clock rate is 256 times sampling clock rate.

Note: BCLKDIV should be set carefully because the peripheral clock frequency must be slower than or equal to system frequency.

I²S Status Register (SPIx_I2SSTS)

Register	Offset	R/W	Description	Reset Value
SPIx_I2SSTS	SPIx_BA+0x68	R/W	I²S Status Register	0x0005_0100

Note: Not supported in SPI mode.

31	30	29	28	27	26	25	24
TXCNT				RXCNT			
23	22	21	20	19	18	17	16
TXRXRST	SLVERRIF	LZCIF	RZCIF	TXUFIF	TXTHIF	TXFULL	TXEMPTY
15	14	13	12	11	10	9	8
I2SENSTS	Reserved		RXTOIF	RXOVIF	RXTHIF	RXFULL	RXEMPTY
7	6	5	4	3	2	1	0
Reserved			RIGHT	Reserved			

Bits	Description	
[31:28]	**TXCNT**	**Transmit FIFO Data Count (Read Only)** This bit field indicates the valid data count of transmit FIFO buffer.
[27:24]	**RXCNT**	**Receive FIFO Data Count (Read Only)** This bit field indicates the valid data count of receive FIFO buffer.
[23]	**TXRXRST**	**TX or RX Reset Status (Read Only)** 0 = The reset function of TXRST or RXRST is done. 1 = Doing the reset function of TXRST or RXRST. **Note:** Both the reset operations of TXRST and RXRST need 3 system clock cycles + 2 peripheral clock cycles. User can check the status of this bit to monitor the reset function is doing or done.
[22]	**SLVERRIF**	**Bit Number Error Interrupt Flag for Slave Mode** 0 = No bit number error event occurred. 1 = Bit number error event occurred. **Note:** This bit will be cleared by writing 1 to it.
[21]	**LZCIF**	**Left Channel Zero Cross Interrupt Flag** 0 = No zero cross event occurred on left channel. 1 = Zero cross event occurred on left channel.
[20]	**RZCIF**	**Right Channel Zero Cross Interrupt Flag** 0 = No zero cross event occurred on right channel. 1 = Zero cross event occurred on right channel.
[19]	**TXUFIF**	**Transmit FIFO Underflow Interrupt Flag** When the transmit FIFO buffer is empty and there is no datum written into the FIFO buffer, if there is more bus clock input, this bit will be set to 1. **Note:** This bit will be cleared by writing 1 to it.
[18]	**TXTHIF**	**Transmit FIFO Threshold Interrupt Flag (Read Only)** 0 = The valid data count within the transmit FIFO buffer is larger than the setting value of TXTH. 1 = The valid data count within the transmit FIFO buffer is less than or equal to the setting value of TXTH. **Note:** If TXTHIEN = 1 and TXTHIF = 1, the SPI/I²S controller will generate a SPI interrupt request.

[17]	**TXFULL**	**Transmit FIFO Buffer Full Indicator (Read Only)** 0 = Transmit FIFO buffer is not full. 1 = Transmit FIFO buffer is full.
[16]	**TXEMPTY**	**Transmit FIFO Buffer Empty Indicator (Read Only)** 0 = Transmit FIFO buffer is not empty. 1 = Transmit FIFO buffer is empty.
[15]	**I2SENSTS**	**I^2S Enable Status (Read Only)** 0 = The SPI/I^2S control logic Disabled. 1 = The SPI/I^2S control logic Enabled. **Note:** The SPI peripheral clock is asynchronous with the system clock. In order to make sure the SPI/I^2S control logic is disabled, this bit indicates the real status of SPI/I^2S control logic for user.
[14:13]	**Reserved**	Reserved.
[12]	**RXTOIF**	**Receive Time-out Interrupt Flag** 0 = No receive FIFO time-out event. 1 = Receive FIFO buffer is not empty and no read operation on receive FIFO buffer over 64 SPI peripheral clock period in Master mode or over 576 SPI peripheral clock period in Slave mode. When the received FIFO buffer is read by software, the time-out status will be cleared automatically. **Note:** This bit will be cleared by writing 1 to it.
[11]	**RXOVIF**	**Receive FIFO Overrun Interrupt Flag** When the receive FIFO buffer is full, the follow-up data will be dropped and this bit will be set to 1. **Note:** This bit will be cleared by writing 1 to it.
[10]	**RXTHIF**	**Receive FIFO Threshold Interrupt Flag (Read Only)** 0 = The valid data count within the receive FIFO buffer is smaller than or equal to the setting value of RXTH. 1 = The valid data count within the receive FIFO buffer is larger than the setting value of RXTH. **Note:** If RXTHIEN = 1 and RXTHIF = 1, the SPI/I^2S controller will generate a SPI interrupt request.
[9]	**RXFULL**	**Receive FIFO Buffer Full Indicator (Read Only)** 0 = Receive FIFO buffer is not full. 1 = Receive FIFO buffer is full.
[8]	**RXEMPTY**	**Receive FIFO Buffer Empty Indicator (Read Only)** 0 = Receive FIFO buffer is not empty. 1 = Receive FIFO buffer is empty.
[7:5]	**Reserved**	Reserved.
[4]	**RIGHT**	**Right Channel (Read Only)** This bit indicates the current transmit data is belong to which channel. 0 = Left channel. 1 = Right channel.
[3:0]	**Reserved**	Reserved.

6.28 Quad Serial Peripheral Interface (QSPI)

6.28.1 Overview

The Quad Serial Peripheral Interface (QSPI) applies to synchronous serial data communication and allows full duplex transfer. Devices communicate in Master/Slave mode with the 4-wire bi-direction interface. The chip contains two QSPI controllers performing a serial-to-parallel conversion on data received from a peripheral device, and a parallel-to-serial conversion on data transmitted to a peripheral device.

The QSPI controller supports 2-bit transfer mode to perform full-duplex 2-bit data transfer and also supports Dual and Quad I/O transfer mode and the controller supports the PDMA function to access the data buffer.

6.28.2 Features

- Up to two sets of QSPI controllers
- Supports Master or Slave mode operation
- Supports 2-bit transfer mode
- Supports Dual and Quad I/O transfer mode
- Configurable bit length of a transaction word from 8 to 32-bit
- For SPI PDMA function disable, provides separate 8-level of 32-bit transmit and receive FIFO buffers
- For SPI PDMA function enable, provides separate 8-level of 32-bit (or 16-level of 16-bit or 32-level of 8-bit) transmit and receive FIFO buffers which depended on SPI setting of data width
- Supports MSB first or LSB first transfer sequence
- Supports Byte Reorder function
- Supports Byte or Word Suspend mode
- Supports PDMA transfer
- Supports 3-Wire, no slave selection signal, bi-direction interface
- Supports one data channel half-duplex transfer
- Supports Double Transfer Rate Mode (DTR mode) for SPI master mode
- Supports receive-only mode

6.28.3 Block Diagram

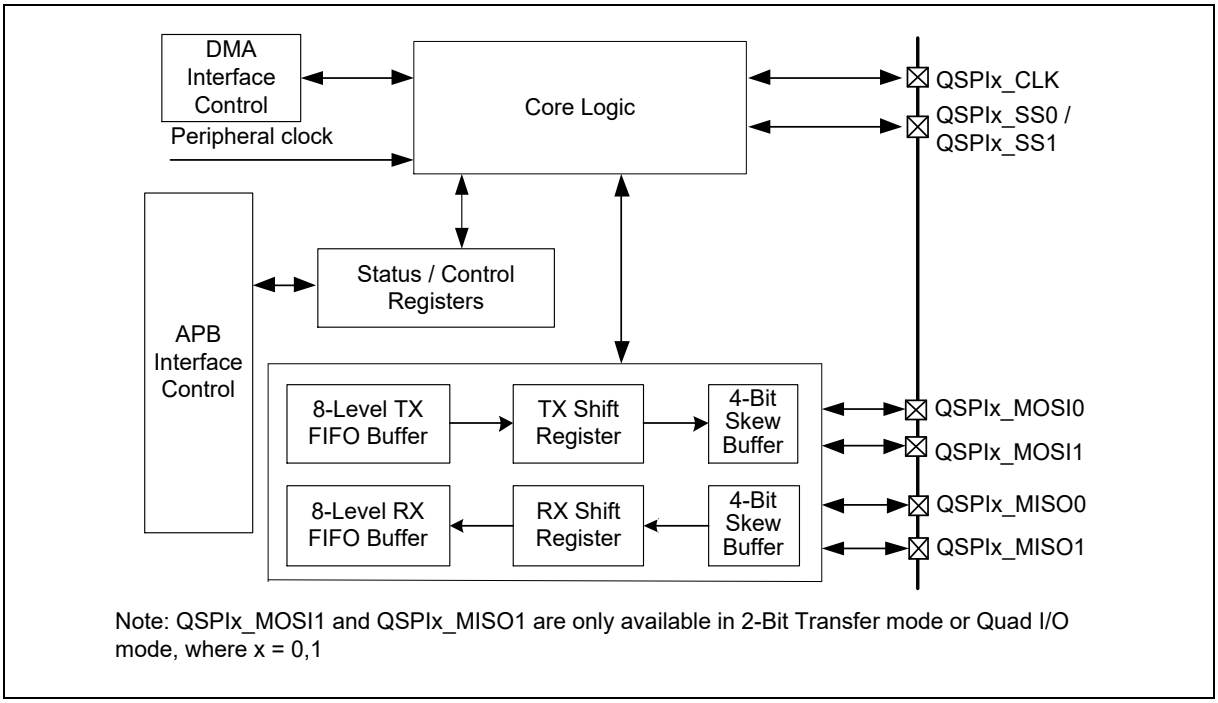


Figure 6.28-1 QSPI Block Diagram

TX FIFO Buffer:

The transmit FIFO buffer is a 8-level depth, 32-bit wide, first-in, first-out register buffer. The data can be written to the transmit FIFO buffer in advance through software by writing the QSPIx_TX register.

RX FIFO Buffer:

The receive FIFO buffer is also a 8-level depth, 32-bit wide, first-in, first-out register buffer. The receive control logic will store the receive data to this buffer. The FIFO buffer data can be read from QSPIx_RX register by software.

TX Shift Register:

The transmit shift register is a 32-bit wide register buffer. The transmit data is loaded from the TX FIFO buffer and shifted out bit-by-bit to the skew buffer.

RX Shift Register:

The receive shift register is also a 32-bit wide register buffer. The receive data is shift in bit-by-bit from the skew buffer and is loaded into RX FIFO buffer when a transaction done.

Skew Buffer:

The skew buffer is a 4-level 1-bit buffer. There are two skew buffers in transmitting and received side. In received side, it is used to shift bits into RX shift register from QSPI bus. In transmitting side, it is used to shift bits into QSPI bus from TX shift register.

6.28.4 Basic Configuration

6.28.4.1 QSPI0 Basic Configuration

- Clock Source Configuration
  - Select the source of QSPI0 peripheral clock on QSPI0SEL (CLK_CLKSEL4[9:8]).

- Enable QSPI0 peripheral clock in QSPI0CKEN (CLK_APBCLK1[6]).

- Reset Configuration
  - Reset QSPI0 controller in QSPI0RST (SYS_IPRST1[12]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
QSPI0	QSPI0_CLK	PD.1	MFP5
	QSPI0_MISO0	PD.3	MFP5
	QSPI0_MISO1	PD.5	MFP5
	QSPI0_MOSI0	PD.2	MFP5
	QSPI0_MOSI1	PD.4	MFP5
	QSPI0_SS0	PD.0	MFP5
	QSPI0_SS1	N/A	N/A

6.28.4.2 QSPI1 Basic Configuration

- Clock Source Configuration
  - Select the source of QSPI1 peripheral clock on QSPI1SEL (CLK_CLKSEL4[11:10]).
  - Enable QSPI1 peripheral clock in QSPI1CKEN (CLK_APBCLK1[7]).
- Reset Configuration
  - Reset QSPI1 controller in QSPI1RST (SYS_IPRST2[4]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
QSPI1	QSPI1_CLK	PD.9	MFP5
		PL.3	MFP6
	QSPI1_MISO0	PD.11	MFP5
		PL.5	MFP6
	QSPI1_MISO1	PD.7	MFP5
		PL.1	MFP6
	QSPI1_MOSI0	PD.10	MFP5
		PL.4	MFP6
	QSPI1_MOSI1	PD.6	MFP5
		PL.0, PL.6	MFP6
	QSPI1_SS0	PD.8	MFP5
	QSPI1_SS1	PL.2	MFP6

6.28.5 Functional Description

6.28.5.1 Terminology

QSPI Peripheral Clock and QSPI Bus Clock

The QSPI controller needs the peripheral clock to drive the QSPI logic unit to perform the data transfer. The peripheral clock rate is determined by the settings of clock divisor (QSPIx_CLKDIV) and the clock source which can be PCLK or APLL. QSPIxSEL of CLK_CLKSEL4 register determines the clock source of the peripheral clock. The DIVIDER (QSPIx_CLKDIV[8:0]) setting determines the divisor of the clock rate calculation.

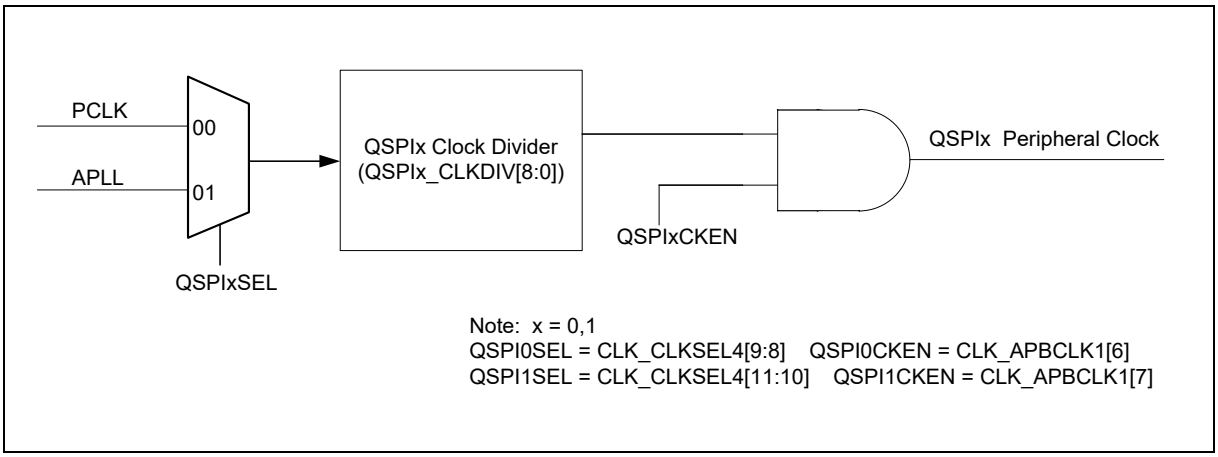


Figure 6.28-2 QSPI Peripheral Clock

In Master mode, the frequency of the QSPI bus clock is equal to the peripheral clock rate. In general, the QSPI bus clock is denoted as QSPI clock. In Slave mode, the QSPI bus clock is provided by a master device. The frequency of QSPI peripheral clock cannot be faster than the system clock rate regardless of Master or Slave mode. If the clock source of peripheral clock is not system clock, the frequency of QSPI peripheral clock shall be slower than the system clock frequency regardless of Master or Slave mode.

Master/Slave mode

The QSPI controllers can be set as Master or Slave mode by setting the SLAVE (QSPIx_CTL[18]) to communicate with the off-chip SPI slave or master device. The HALFDPX (QSPIx_CTL[14]) can be used to select the full-duplex or half-duplex in QSPI transmission. The application block diagrams in Master and Slave mode are shown below.

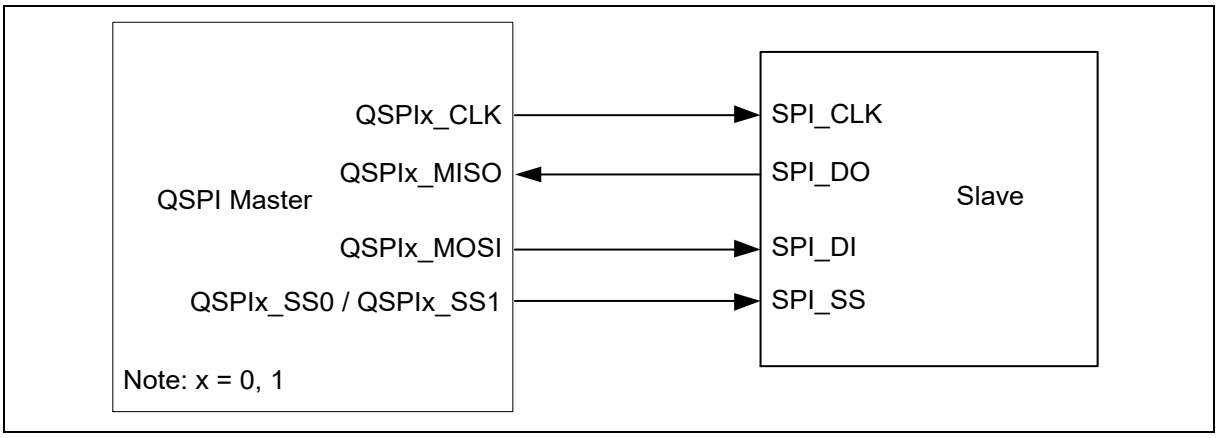


Figure 6.28-3 QSPI Full-Duplex Master Mode Application Block Diagram

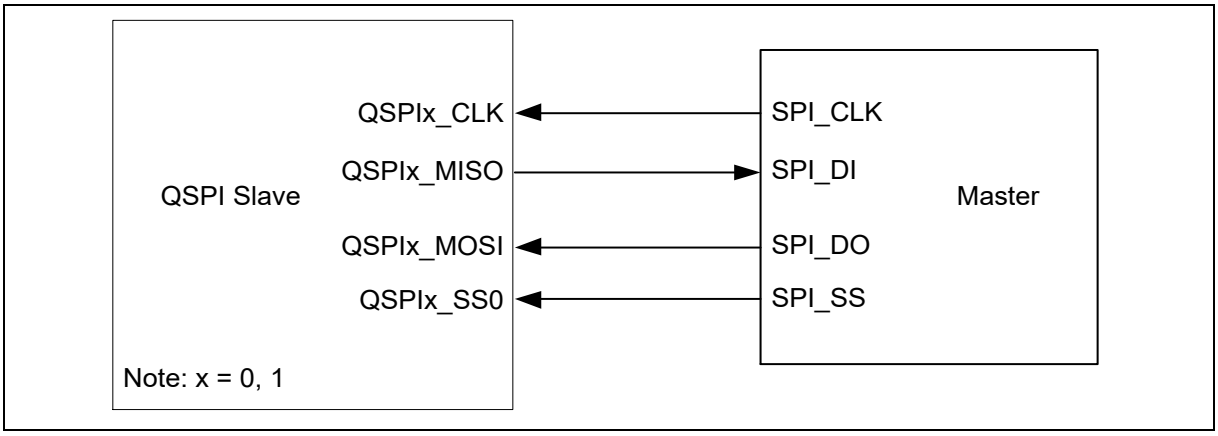


Figure 6.28-4 QSPI Full-Duplex Slave Mode Application Block Diagram

Slave Selection

In Master mode, the QSPI controller can drive off-chip slave device through the slave select output pin QSPIx_SS0 or QSPIx_SS1. In Slave mode, the off-chip master device drives the slave selection signal from the QSPIx_SS0 input port to this QSPI controller. The duration between the slave select active edge and the first QSPI clock input shall over 3 QSPI peripheral clock cycles of slave.

In Master/Slave mode, the active state of slave selection signal can be programmed to low or high active in SSACTPOL (QSPIx_SSCTL[2]). The selection of slave select conditions depends on what type of device is connected. In Slave mode, to recognize the inactive state of the slave selection signal, the inactive period of the slave selection signal must be larger than or equal to 3 peripheral clock cycles between two successive transactions.

In Slave mode, the QSPIx_MISO signal type is controlled by the QSPIx_SS0 pin. If the QSPIx_SS0 pin is inactive state, the QSPIx_MISO is set to tri-state to avoid interference with other Slave devices. If the QSPIx_SS0 pin is active state, the QSPIx_MISO is set to output mode for data transfer.

Timing Condition

The CLKPOL (QSPIx_CTL[3]) defines the QSPI clock idle state. If CLKPOL = 1, the output QSPI clock is idle at high state; if CLKPOL = 0, it is idle at low state.

TXNEG (QSPIx_CTL[2]) defines the data transmitted out either on negative edge or on positive edge of QSPI clock. RXNEG (QSPIx_CTL[1]) defines the data received either on negative edge or on positive edge of QSPI clock.

Note: The settings of TXNEG and RXNEG are mutual exclusive. In other words, do not transmit and receive data at the same clock edge.

Transmit/Receive Bit Length

The bit length of a transaction word is defined in DWIDTH (QSPIx_CTL[12:8]) and can be configured up to 32-bit length in a transaction word for transmitting and receiving.

When QSPI controller finishes a transaction, i.e. receives or transmits a specific count of bits defined in DWIDTH (QSPIx_CTL[12:8]), the unit transfer interrupt flag UNITIF (QSPIx_STATUS[1]) will be set to 1.

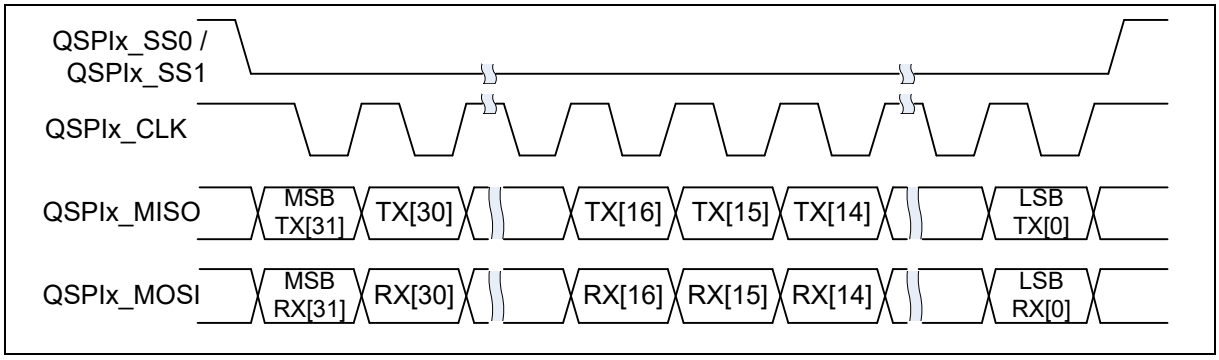


Figure 6.28-5 32-bit in One Transaction

LSB/MSB First

LSB (QSPIx_CTL[13]) defines the bit transfer sequence in a transaction. If the LSB (QSPIx_CTL[13]) is set to 1, the transfer sequence is LSB first. The bit 0 will be transferred firstly. If the LSB (QSPIx_CTL[13]) is cleared to 0, the transfer sequence is MSB first.

Suspend Interval

SUSPITV (QSPIx_CTL[7:4]) provides a configurable suspend interval, 0.5 ~ 15.5 QSPI clock periods, between two successive transaction words in Master mode. The definition of the suspend interval is the interval between the last clock edge of the preceding transaction word and the first clock edge of the following transaction word. The default value of SUSPITV is 0x3 (3.5 QSPI clock cycles).

6.28.5.2 Automatic Slave Selection

In Master mode, if AUTOSS (QSPIx_SSCTL[3]) is set, the slave selection signal will be generated automatically and output to the QSPIx_SS0 or QSPIx_SS1 pin according to whether SS0 (QSPIx_SSCTL[0]) or SS1 (QSPIx_SSCTL[1]) is enabled or not. The slave selection signal will be set to active state by the SPI controller when the QSPI data transfer is started by writing to FIFO. It will be set to inactive state when QSPI bus is idle. If QSPI bus is not idle, i.e. TX FIFO, TX shift register or TX skew buffer is not empty, the slave selection signal will be set to inactive state between transactions if the value of SUSPITV (QSPIx_CTL[7:4]) is greater than or equal to 3.

In Master mode, if the value of SUSPITV is less than 3 and the AUTOSS is set as 1, the slave selection signal will be kept at active state between two successive transactions.

If the AUTOSS bit is cleared, the slave selection output signal will be determined by the SS0 or SS1 setting. The active state of the slave selection output signal is specified in SSACTPOL (QSPIx_SSCTL[2]).

The duration between the slave selection signal active edge and the first QSPI bus clock edge is 1 QSPI bus clock cycle and the duration between the last QSPI bus clock and the slave selection signal inactive edge is 1.5 QSPI bus clock cycle.

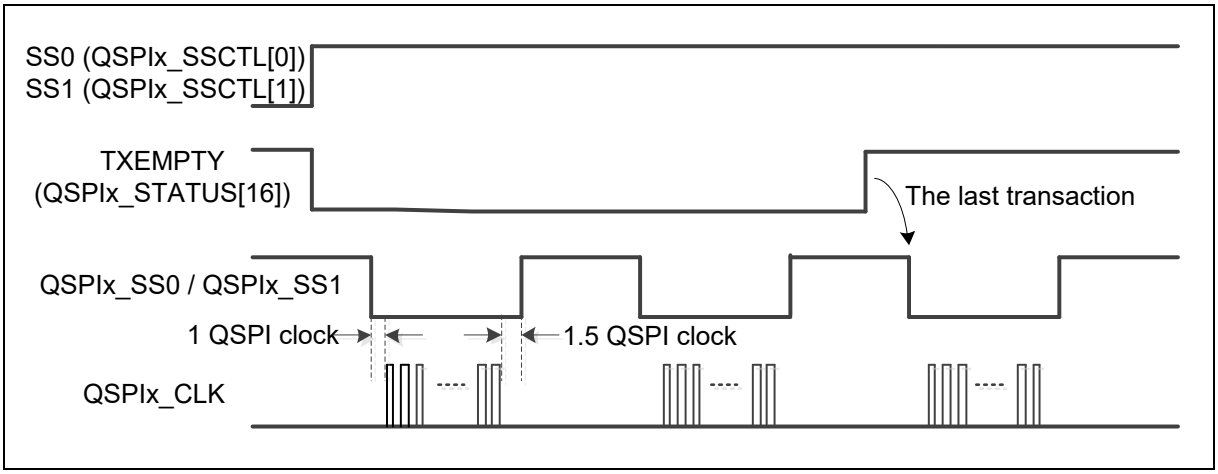


Figure 6.28-6 Automatic Slave Selection (SSACTPOL = 0, SUSPITV > 0x2)

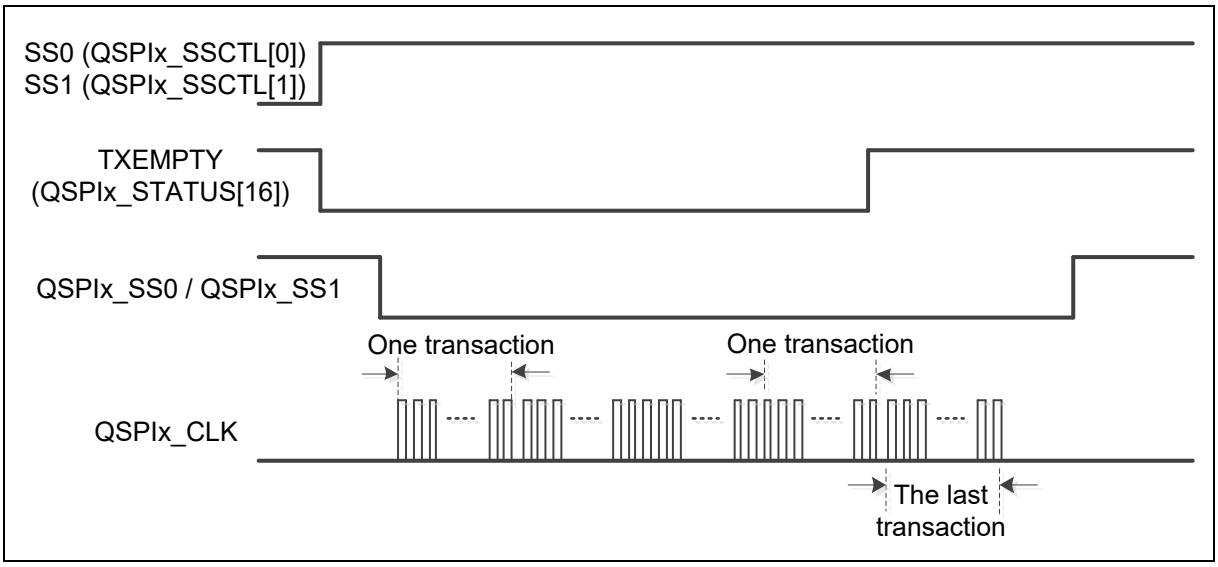


Figure 6.28-7 Automatic Slave Selection (SSACTPOL = 0, SUSPITV < 0x3)

6.28.5.3 Byte Reorder and Suspend Function

When the transfer is set as MSB first (LSB = 0) and the REORDER (QSPIx_CTL[19]) is set to 1, the data stored in the TX buffer and RX buffer will be rearranged in the order as [Byte0, Byte1, Byte2, Byte3] in 32-bit transfer (DWIDTH = 0). The sequence of transmitted/received data will be Byte0, Byte1, Byte2, and then Byte3. If the DWIDTH is set as 24-bit transfer mode, the data in TX buffer and RX buffer will be rearranged as [unknown byte, Byte0, Byte1, Byte2]. The QSPI controller will transmit/receive data with the sequence of Byte0, Byte1 and then Byte2. Each byte will be transmitted/received with MSB first. The rule of 16-bit mode is the same as above. Byte Reorder function is only available when DWIDTH is configured as 16, 24, and 32 bits.

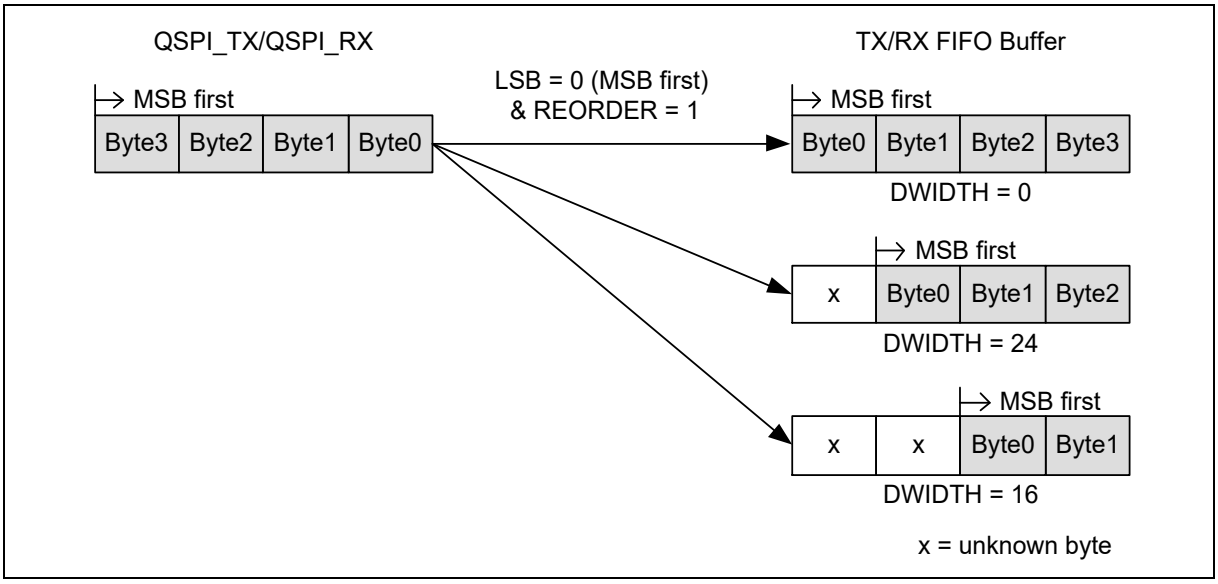


Figure 6.28-8 Byte Reorder Function

In Master mode, if REORDER (QSPIx_CTL[19]) is set to 1, a suspend interval of 0.5 ~ 15.5 QSPI clock periods will be inserted by hardware between two successive bytes in a transaction word. The suspend interval is configured in SUSPITV (QSPIx_CTL[7:4]).

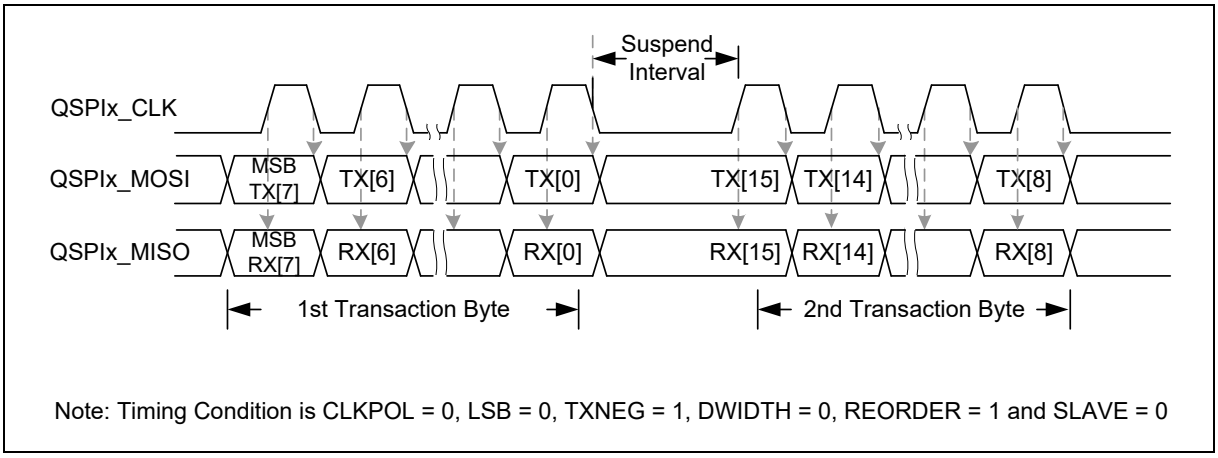


Figure 6.28-9 Timing Waveform for Byte Suspend

6.28.5.4 Half-Duplex Communication

The QSPI controller can communicate in half-duplex mode by setting HALFDPX (QSPIx_CTL[14]) bit. In half-duplex mode, there is only one data line for receiving or transmitting data direction which is defined by DATDIR (QSPIx_CTL[20]). In half-duplex configuration, the QSPIx_MISO pin is free for other applications and it can be configured as GPIO. Enabling or disabling the control bit HALFDPX (QSPIx_CTL[14]) will produce TXFBCLR (QSPIx_FIFOCTL[9]) and RXFBCLR (QSPIx_FIFOCTL[8]) at the same time automatically.

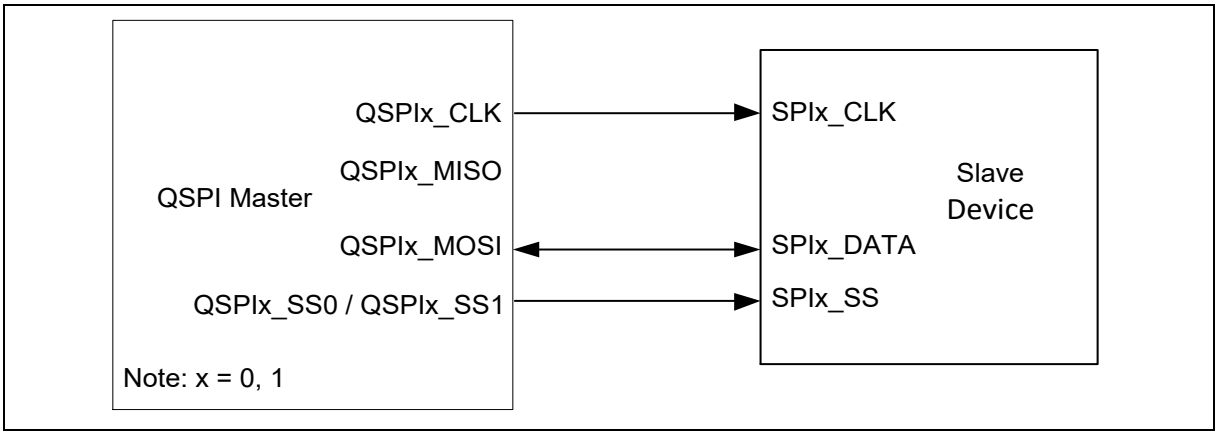


Figure 6.28-10 QSPI Half-Duplex Master Mode Application Block Diagram

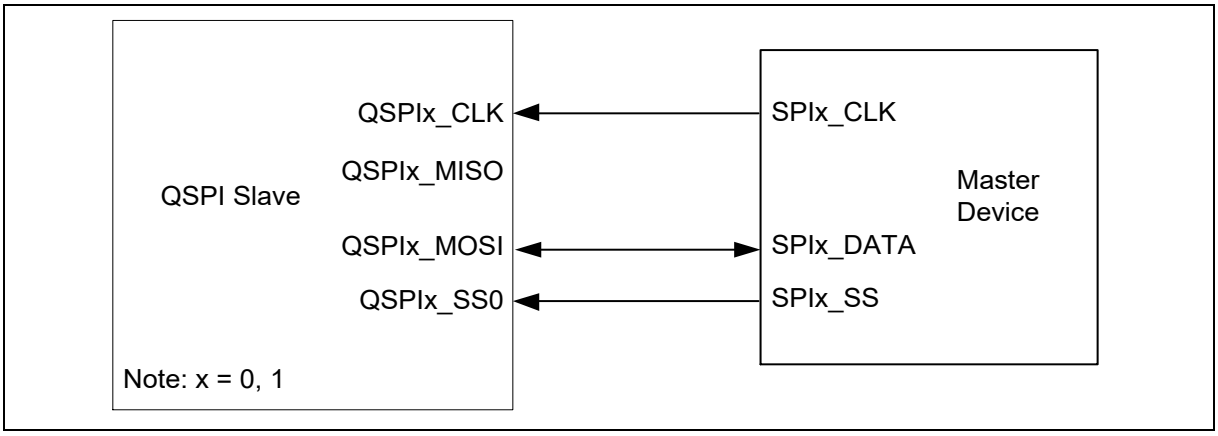


Figure 6.28-11 QSPI Half-Duplex Slave Mode Application Block Diagram

6.28.5.5 Receive-Only Mode

The QSPI Master device can communicate in receive-only mode by setting RXONLY (QSPIx_CTL[15]). In this configuration, the QSPI Master device will generate QSPI bus clock continuously as long as the receive-only mode is enabled for receiving data bit from SPI slave device. If AUTOSS (QSPIx_SSCTL[3]) is enabled in receive-only mode, QSPI Master will keep activating the slave select signal.

The remaining QSPIx_MOSI pin of QSPI Master device is not used for communication and can be configured as GPIO. The status BUSY (QSPIx_STATUS[0]) will be asserted in receive-only mode due to the generation of QSPI bus clock. Entering this mode will produce the TXFBCLR (QSPIx_FIFOCTL[9]) and RXFBCLR (QSPIx_FIFOCTL[8]) at the same time automatically. When user enables this mode, the output QSPI bus clock will be sent out after 6 peripheral clock cycles. In this mode, the data which has been written into transmit FIFO will be loaded into transmit shift register and sent out.

When user sets RXONLY (QSPIx_CTL[15]) enable, QSPI RX data with data bit width of DWIDTH (QSPIx_CTL[12:8]) will be received into RX FIFO and QSPI clock will be sent to SPI slave device until RX FIFO is full.

For two-bit transfer mode TWOBIT(QSPIx_CTL[16]) enable, the QSPI master will send QSPI output clock to SPI slave and receive RX data until RX FIFO is full. After user reads RX data from RX FIFO, QSPI master will send QSPI output clock to SPI slave and receive RX data again when RX FIFO counter RXCNT is less than or equal to 4.

6.28.5.6 Slave 3-Wire Mode

When SLV3WIRE (QSPIx_SSCTL[4]) is set by software to enable the Slave 3-Wire mode, the QSPI controller can work with no slave selection signal in Slave mode. The SLV3WIRE (QSPIx_SSCTL[4]) only takes effect in Slave mode. Only three pins, QSPIx_CLK, QSPIx_MISO, and QSPIx_MOSI, are required to communicate with a SPI master. The QSPIx_SS0 and QSPIx_SS1 pin can be configured as a GPIO. When the SLV3WIRE (QSPIx_SSCTL[4]) is set to 1, the QSPI slave will be ready to transmit/receive data after the SPIEN (QSPIx_CTL[0]) is set to 1.

6.28.5.7 PDMA Transfer Function

The QSPI controller supports PDMA transfer function.

When TXPDMAEN (QSPIx_PDMACTL[0]) is set to 1, the controller will issue request to PDMA controller to start the PDMA transmission process automatically. Data width DWIDTH (QSPIx_CTL[12:8]) only supports 8-bits, 16-bits, and 32-bits when TXPDMAEN (QSPIx_PDMACTL[0]) is set to 1 or RXPDMAEN (QSPIx_PDMACTL[1]) is set to 1.

The TX FIFO will be configured to 8x32 bits automatically when TXPDMAEN (QSPIx_PDMACTL[0]) is set to 1 and QSPI data width DWIDTH (QSPIx_CTL[12:8]) is 32 bits. The TX FIFO will be configured to 16x16 bits automatically when TXPDMAEN (QSPIx_PDMACTL[0]) is set to 1 and QSPI data width DWIDTH (QSPIx_CTL[12:8]) is 16 bits. The TX FIFO will be configured to 32x8 bits automatically when TXPDMAEN (QSPIx_PDMACTL[0]) is set to 1 and QSPI data width DWIDTH (QSPIx_CTL[12:8]) is 8 bits.

When RXPDMAEN (QSPIx_PDMACTL[1]) is set to 1, the controller will start the PDMA reception process. QSPI controller will issue request to PDMA controller automatically when there is data in the RX FIFO buffer.

The RX FIFO will be configured to 8x32 bits automatically when RXPDMAEN (QSPIx_PDMACTL[1]) is set to 1 and QSPI data width DWIDTH (QSPIx_CTL[12:8]) is 32 bits. The RX FIFO will be configured to 16x16 bits automatically when RXPDMAEN (QSPIx_PDMACTL[1]) is set to 1 and QSPI data width DWIDTH (QSPIx_CTL[12:8]) is 16 bits. The RX FIFO will be configured to 32x8 bits automatically when RXPDMAEN (QSPIx_PDMACTL[1]) is set to 1 and QSPI data width DWIDTH (QSPIx_CTL[12:8]) is 8 bits.

Note: The QSPI supports single request PDMA (Read/Write) only, and burst request PDMA is not supported.

6.28.5.8 Two-bit Transfer Mode

The QSPI controller also supports 2-bit transfer mode when setting TWOBIT (QSPIx_CTL[16]) to 1. In 2-bit transfer mode, the QSPI controller performs full duplex data transfer. In other words, the two serial data bits can be transmitted and received simultaneously.

For example, in Master mode, the even data (TX Data (n)) stored in the TX FIFO of QSPIx will be transmitted through the QSPIx_MOSI0 pin and the odd data (TX Data (n+1)) stored in the TX FIFO of QSPIx will be transmitted through the QSPIx_MOSI1 pin respectively. In the meanwhile, the even data received from QSPIx_MISO0 pin will be written to RX FIFO prior to the odd data received from QSPIx_MISO1 pin.

In Slave mode, the even and odd data stored in the TX FIFO of QSPIx will be transmitted through the QSPIx_MISO0 pin and QSPIx_MISO1 pin respectively. In the meanwhile, the RX FIFO of QSPIx will store the even data received from the QSPIx_MOSI0 pin and the odd data from QSPIx_MOSI1 pin respectively. The data sequence of FIFO buffers is the same as the Master mode.

Note: QSPI_SS0/QSPI_SS1 signals for Slave 0 and Slave 1 are in active state for two-bit transfer mode simultaneously when data is transmitted and received.

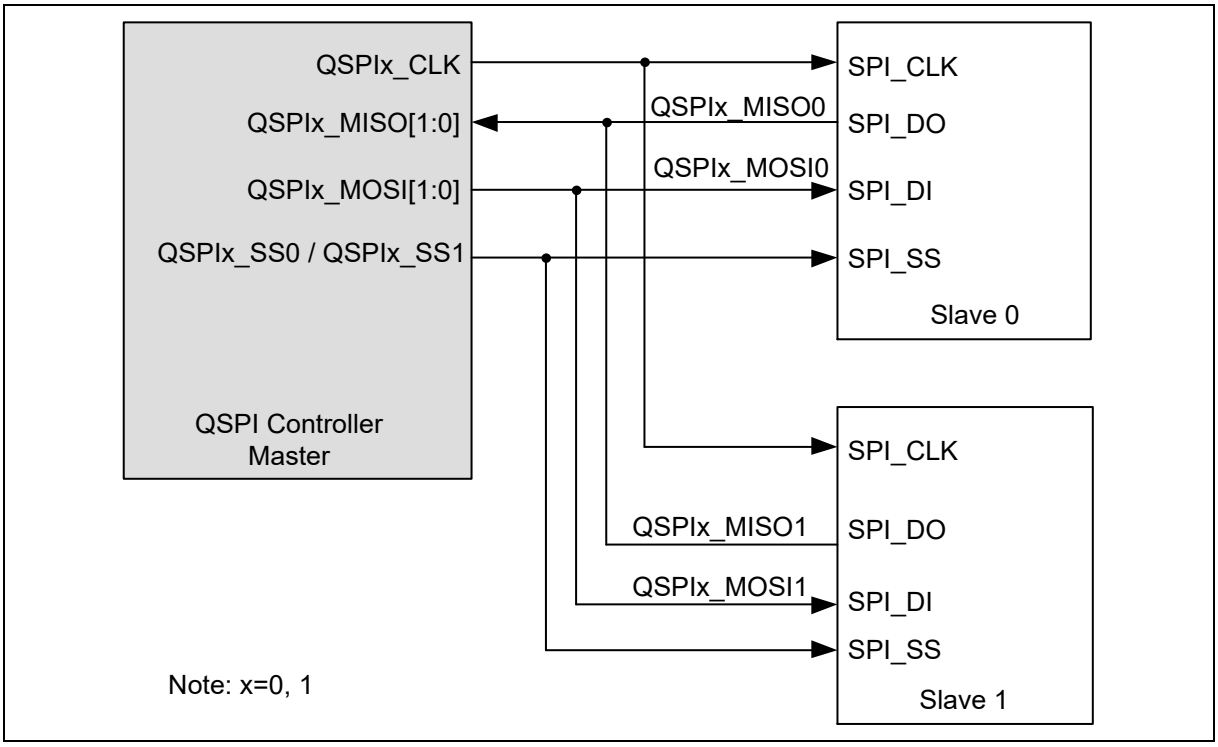


Figure 6.28-12 Two-bit Transfer Mode System Architecture

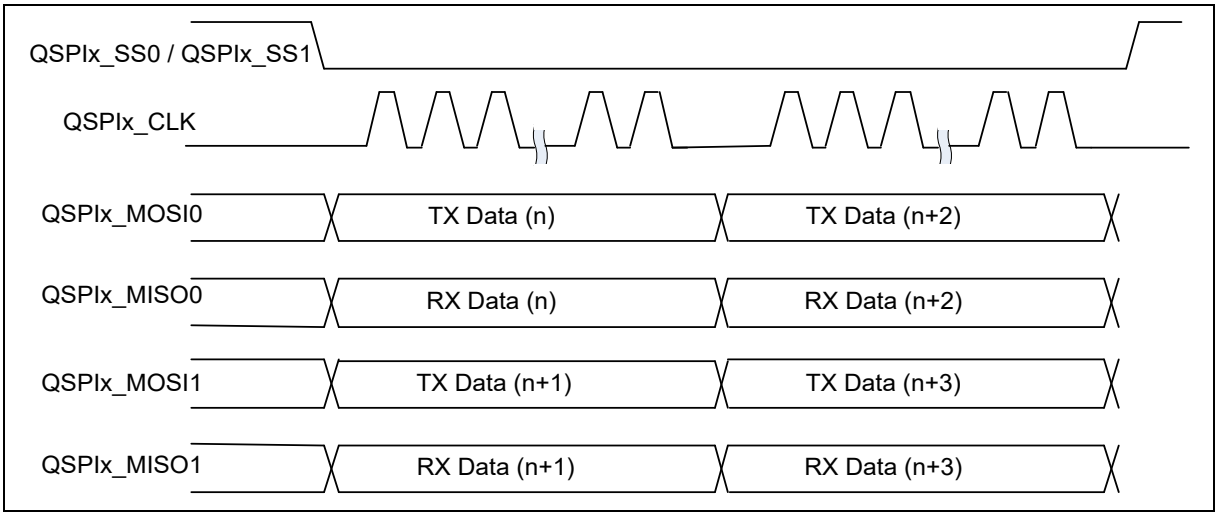


Figure 6.28-13 Two-bit Transfer Mode Timing (Master Mode)

6.28.5.9 Dual I/O Mode

The QSPI controller also supports Dual I/O transfer when setting the DUALIOEN ((QSPIx_CTL[21]) to 1. Many general SPI Flashes support Dual I/O transfer. The DATDIR (QSPIx_CTL[20]) is used to define the direction of the transfer data. When the DATDIR bit is set to 1, the controller will send the data to external device. When the DATDIR bit is set to 0, the controller will read the data from the external device. This function supports 8, 16, 24, and 32 bits of length. The Dual I/O mode is not supported when the Slave 3-Wire mode is enabled.

For Dual I/O mode, if both the DUALIOEN (QSPIx_CTL[21]) and DATDIR (QSPIx_CTL[20]) are set as 1, the QSPIx_MOSI0 is the even bit data output and the QSPIx_MISO0 will be set as the odd bit data output. If the DUALIOEN (QSPIx_CTL[21]) is set as 1 and DATDIR (QSPIx_CTL[20]) is set as 0, both

the QSPIx_MISO0 and QSPIx_MOSI0 will be set as data input ports.

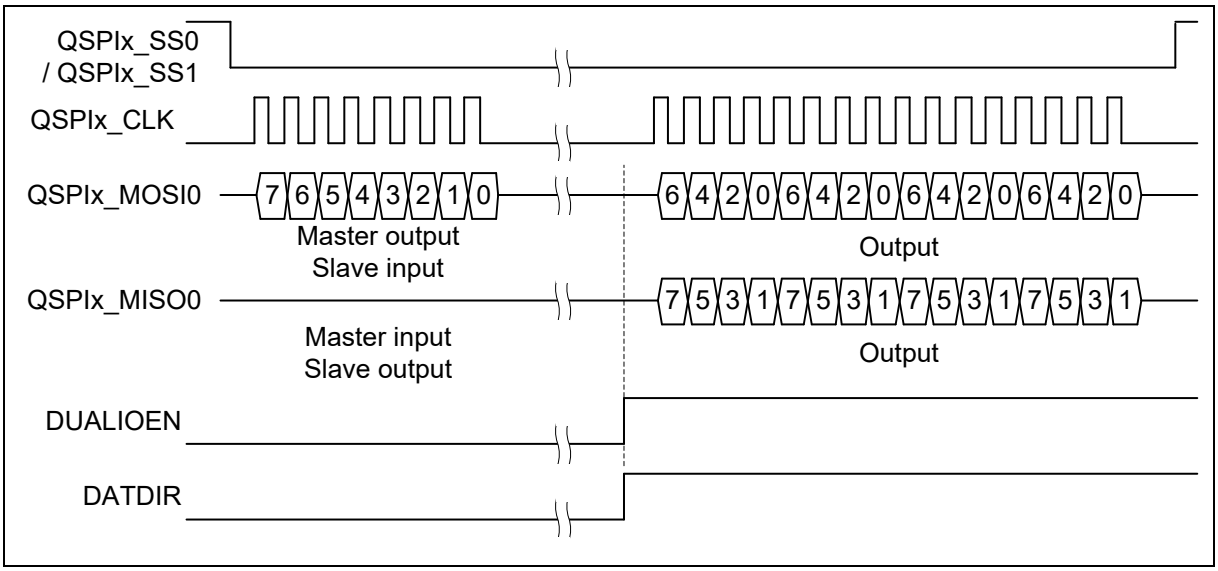


Figure 6.28-14 Bit Sequence of Dual Output Mode

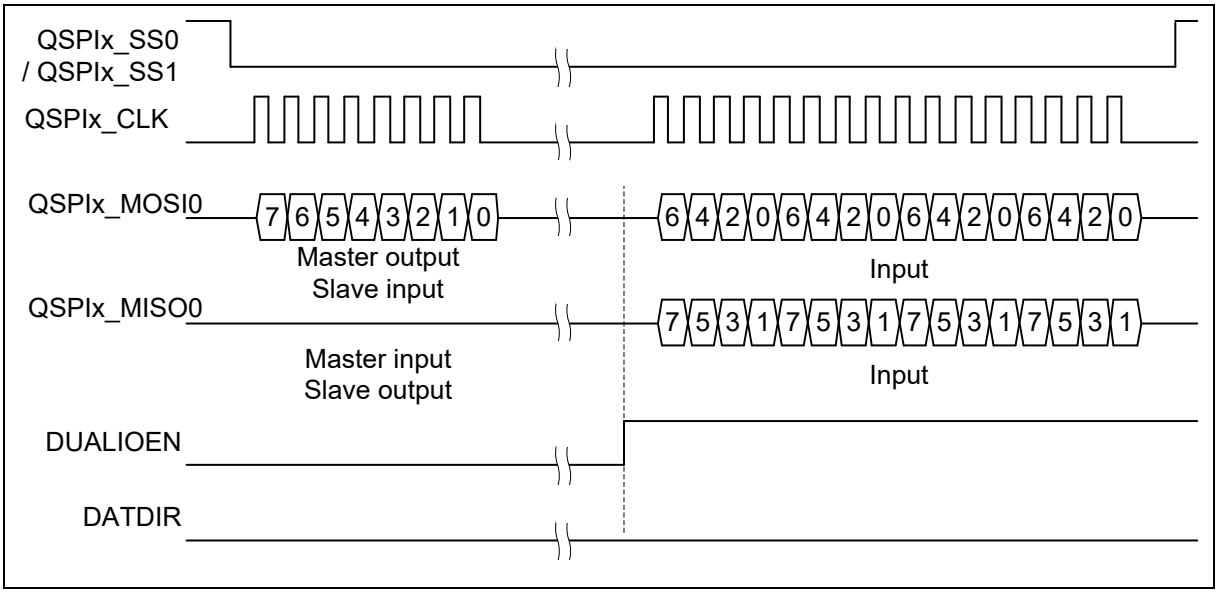


Figure 6.28-15 Bit Sequence of Dual Input Mode

6.28.5.10 Quad I/O Mode

The QSPI controller also supports Quad I/O transfer when setting the QUADIOEN (QSPIx_CTL[22]) to 1. Many general SPI Flashes support Quad I/O transfer. The DATDIR bit (QSPIx_CTL[20]) is used to define the direction of the transfer data. When the DATDIR (QSPIx_CTL[20]) is set to 1, the controller will send the data to external device. When the DATDIR (QSPIx_CTL[20]) is set to 0, the controller will read the data from the external device. This function supports 8, 16, 24, and 32 bits of length.

The Quad I/O mode is not supported when the Slave 3-Wire mode is enabled. The DUALIOEN (QSPIx_CTL[21]) and QUADIOEN (QSPIx_CTL[22]) shall not be set to 1 simultaneously.

For Quad I/O mode, if both the QUADIOEN (QSPIx_CTL[22]) and DATDIR (QSPIx_CTL[20]) are set as 1, the QSPIx_MOSI0 and QSPIx_MOSI1 are the even bit data output and the QSPIx_MISO0 and QSPIx_MISO1 will be set as the odd bit data output. If the QUADIOEN (QSPIx_CTL[22]) is set as 1 and

DATDIR (QSPIx_CTL[20]) is set as 0, all the QSPIx_MISO0, QSPIx_MISO1, QSPIx_MOSI0 and QSPIx_MOSI1 pins will be set as data input ports.

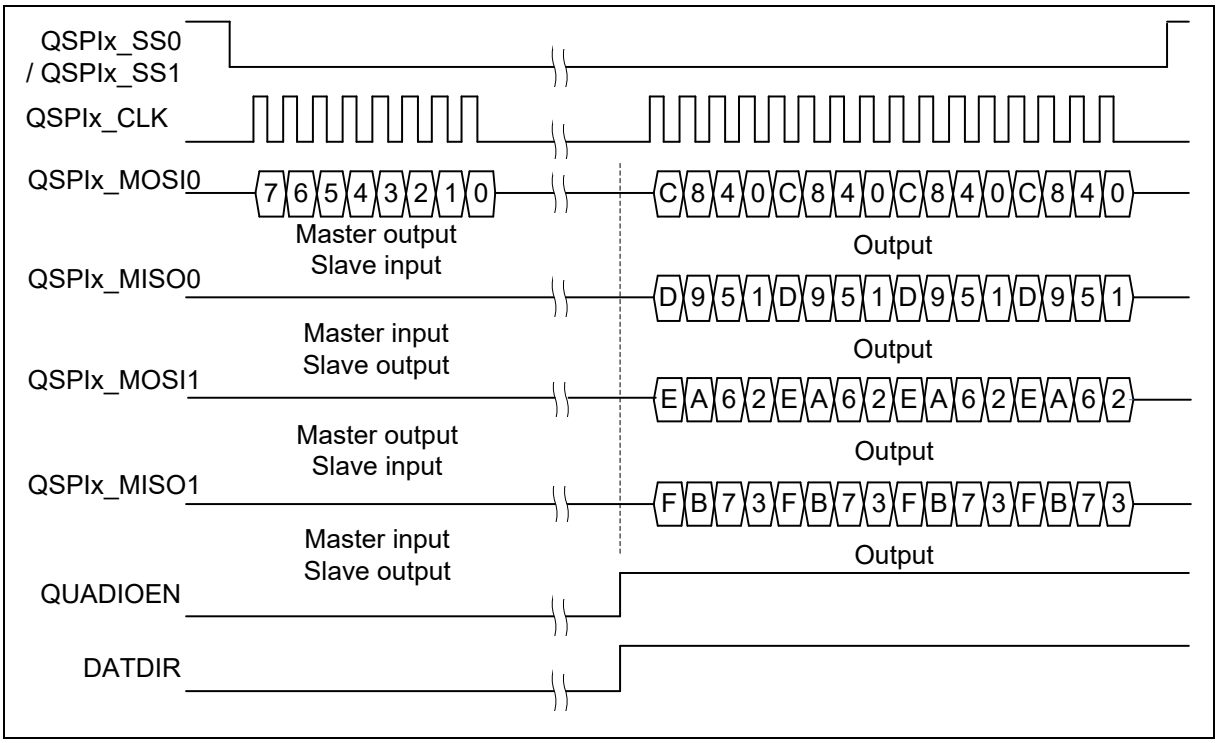


Figure 6.28-16 Bit Sequence of Quad Output Mode

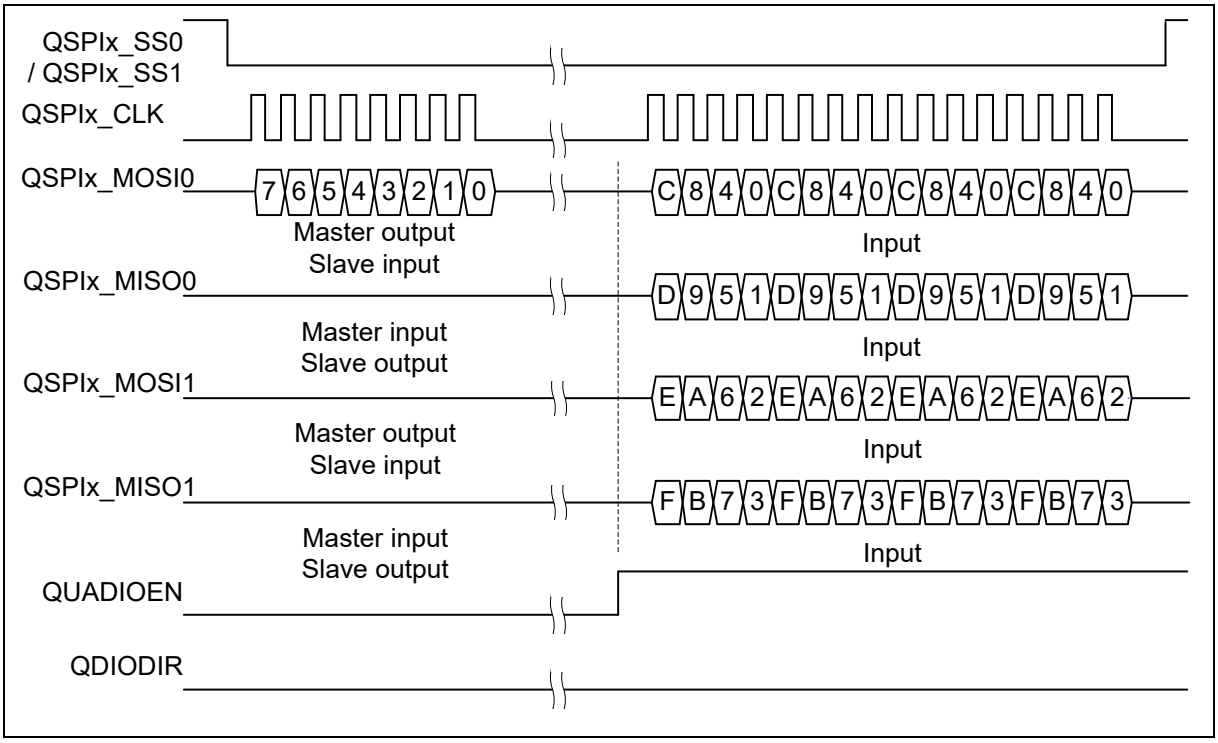


Figure 6.28-17 Bit Sequence of Quad Input Mode

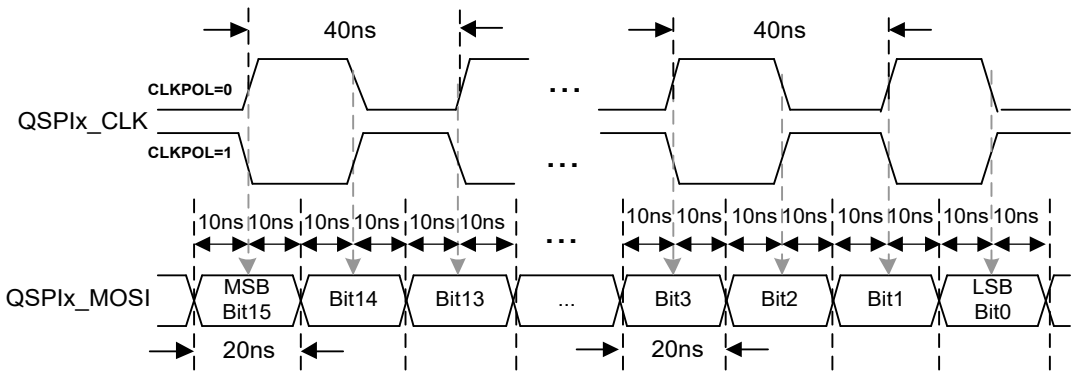
6.28.5.11 Double Transfer Rate Mode

The QSPI master mode supports DTR mode (Double Transfer Rate Mode) when user sets control register DTREN (QSPIx_CTL[23]) to 1. The timing diagram of DTR mode for SPI output data, SPI input data, and SPI output bus clock are shown in Figure 6.28-18 and Figure 6.28-19. In generally, QSPI DTR function of master mode is usually used for application of SPI Flash controller, where the SPI Flash supports DTR commands such as 0xED, 0xBD, and 0x0D.

In DTR mode, the frequency of QSPI master output bus clock (QSPIx_CLK) equals to the half frequency of QSPI peripheral clock (QSPI_ECLK). TX output data of QSPI master in DTR mode will be sent near rising edge and falling edge of QSPIx_CLK, where the rising edge and falling edge of QSPIx_CLK are the half period of QSPI peripheral clock (QSPI_ECLK) behind data driving edge of QSPI output data QSPIx_MOSI.

User can use control register CLKPOL (QSPI_CTL[3]) to select idle polarity of QSPIx_CLK in DTR mode. When user sets CLKPOL to 0, the idle polarity of QSPIx_CLK is 0. When user sets CLKPOL to 1, the idle polarity of QSPIx_CLK is 1.

Note: QSPI master mode supports DTR mode, and QSPI slave mode does not support this mode.



For example : QSPI_ECLK = 50MHz, QSPIx_CLK = 25MHz

Figure 6.28-18 Timing Diagram of QSPI Output Data and Output Bus clock for Double Transfer Rate Mode

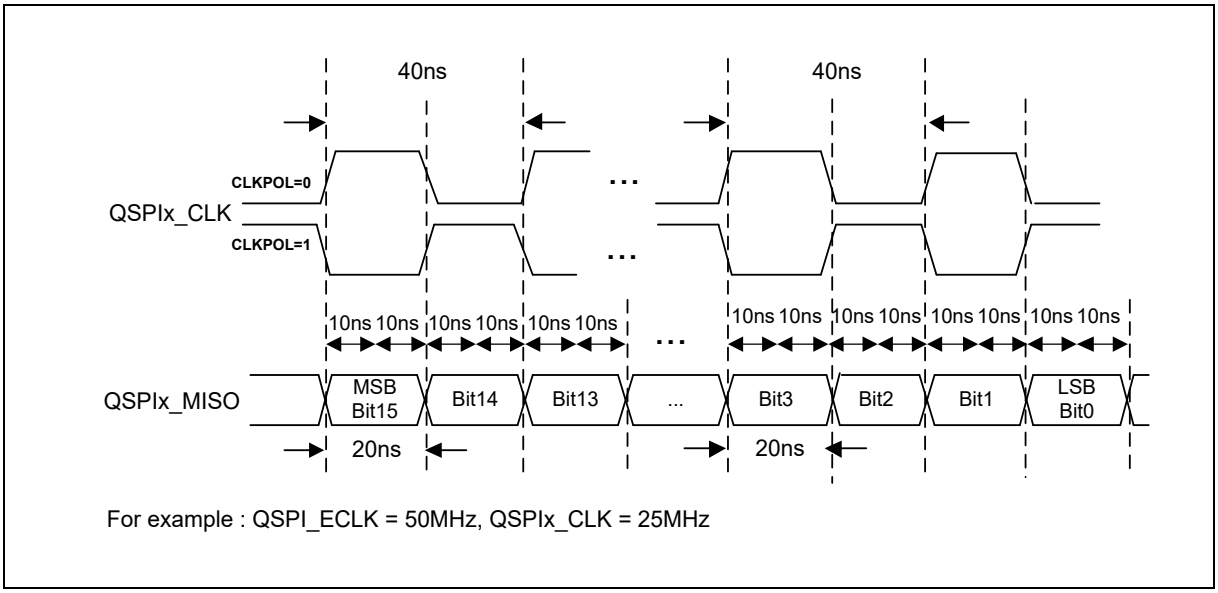


Figure 6.28-19 Timing Diagram of QSPI Input Data and Output Bus clock for Double Transfer Rate Mode

6.28.5.12 FIFO Buffer Operation

The QSPI controller is equipped with eight 32-bit wide transmit and receive FIFO buffers. The data stored in the transmit FIFO buffer will be read and sent out by the transmission control logic. If the transmit FIFO buffer is full, the TXFULL (QSPIx_STATUS[17]) will be set to 1. When the QSPI transmission logic unit draws out the last datum of the transmit FIFO buffer, so that the transmit FIFO buffer is empty, the TXEMPTY (QSPIx_STATUS[16]) will be set to 1. Note that the TXEMPTY (QSPIx_STATUS[16]) flag is set to 1 while the last transaction is still in progress. In Master mode, the BUSY (QSPIx_STATUS[0]) is set to 1 when the FIFO buffer is written any data or there is any transaction on the QSPI bus. (e.g. the slave selection signal is active and the QSPI controller is receiving data in Slave mode). It will be set to 0 when the transmit FIFO is empty and the current transaction has done. Thus, the status of BUSY (QSPIx_STATUS[0]) should be checked by software to make sure whether the QSPI is in idle or not.

The receive control logic will store the QSPI input data into the receive FIFO buffer. There are FIFO related status bits, like RXEMPTY (QSPIx_STATUS[8]) and RXFULL (QSPIx_STATUS[9]), to indicate the current status of RX FIFO buffer.

The transmitting and receiving threshold can be configured by setting TXTH (QSPIx_FIFOCTL[30:28]) and RXTH (QSPIx_FIFOCTL[26:24]). When the count of valid data stored in transmit FIFO buffer is less than or equal to TXTH (QSPIx_FIFOCTL[30:28]) setting, TXTHIF (QSPIx_STATUS[18]) will be set to 1. When the count of valid data stored in receive FIFO buffer is larger than RXTH (QSPIx_FIFOCTL[26:24]) setting, RXTHIF (QSPIx_STATUS[10]) will be set to 1.

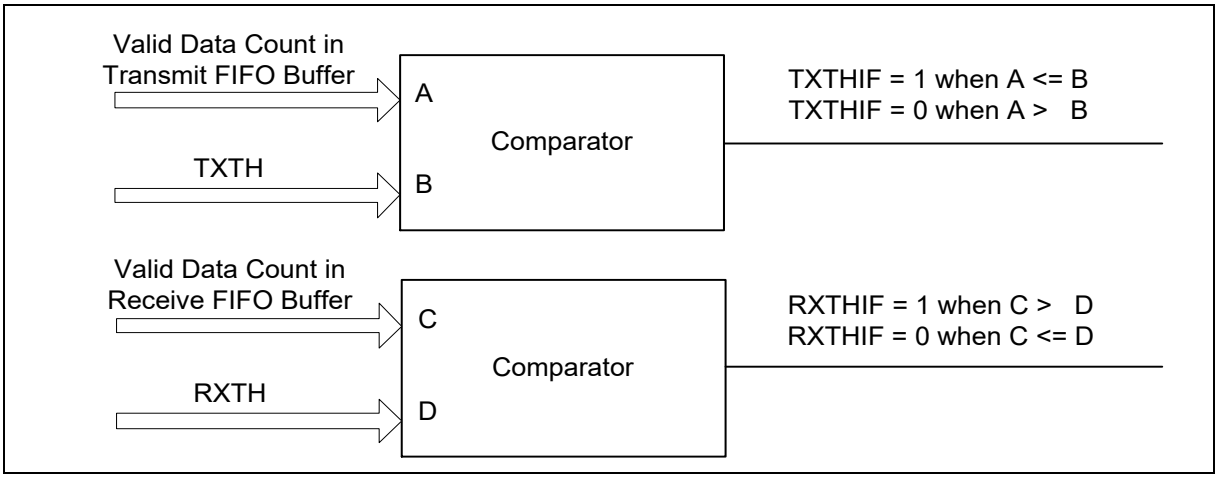


Figure 6.28-20 FIFO Threshold Comparator

In Master mode, when the first datum is written to the QSPIx_TX register, the TXEMPTY flag (QSPIx_STATUS[16]) will be cleared to 0. The transmission will start after 1 PCLK clock cycles and 6 peripheral clock cycles. User can write the next data into QSPIx_TX register immediately. The QSPI controller will insert a suspend interval between two successive transactions. The period of suspend interval is decided by the setting of SUSPITV (QSPIx_CTL[7:4]). If the SUSPITV (QSPIx_CTL[7:4]) equals 0, QSPI controller can perform continuous transfer. User can write data into QSPIx_TX register as long as the TXFULL (QSPIx_STATUS[17]) is 0.

In the Example 1 of Figure 6.28-21, it indicates the updated condition of TXEMPTY (QSPIx_STATUS[16]) and the relationship among the FIFO buffer, shift register and the skew buffer. The TXEMPTY (QSPIx_STATUS[16]) is set to 0 when the Data0 is written into the FIFO buffer. The Data0 will be loaded into the shift register by the core logic and the TXEMPTY (QSPIx_STATUS[16]) will be to 1. The Data0 in shift register will be shift into skew buffer by bit for transmission until the transfer is done.

In the Example 2 of Figure 6.28-21, it indicates the updated condition of TXFULL (QSPIx_STATUS[17]) when there are 8 data in the FIFO buffer and the next data of Data9 does not be written into the FIFO buffer when the TXFULL = 1.

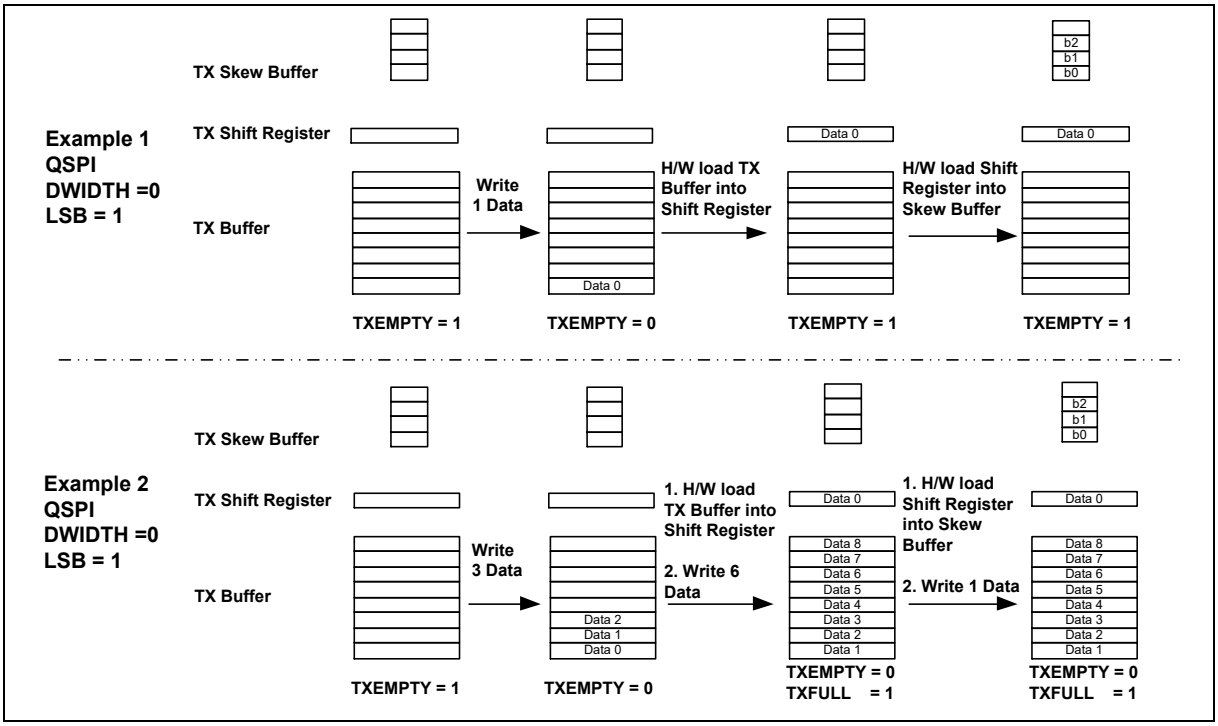


Figure 6.28-21 Transmit FIFO Buffer Example

The subsequent transactions will be triggered automatically if the transmitted data are updated in time. If the QSPIx_TX register does not be updated after all data transfer are done, the transfer will stop.In Master mode, during receiving operation, the serial data are received from QSPIx_MISO pin and stored to receive FIFO buffer.

The received data (Data0's b0, b1, …b31) is stored into skew buffer first according the serial clock (QSPIx_CLK) and then it is shift into the shift register bit by bit. The core logic will load the data in shift register into FIFO buffer when the received data bit count reach the value of DWIDTH (QSPIx_CTL[12:8]). The RXEMPTY (QSPIx_STATUS[8]) will be cleared to 0 while the receive FIFO buffer contains unread data (see the Example 1 of Receive FIFO Buffer Example in Figure 6.28-22). The received data can be read by software from QSPIx_RX register as long as the RXEMPTY (QSPIx_STATUS[8]) is 0. If the receive FIFO buffer contains 8 unread data, the RXFULL (QSPIx_STATUS[9]) will be set to 1 (see the Example 2 of Receive FIFO Buffer Example in Figure 6.28-22).

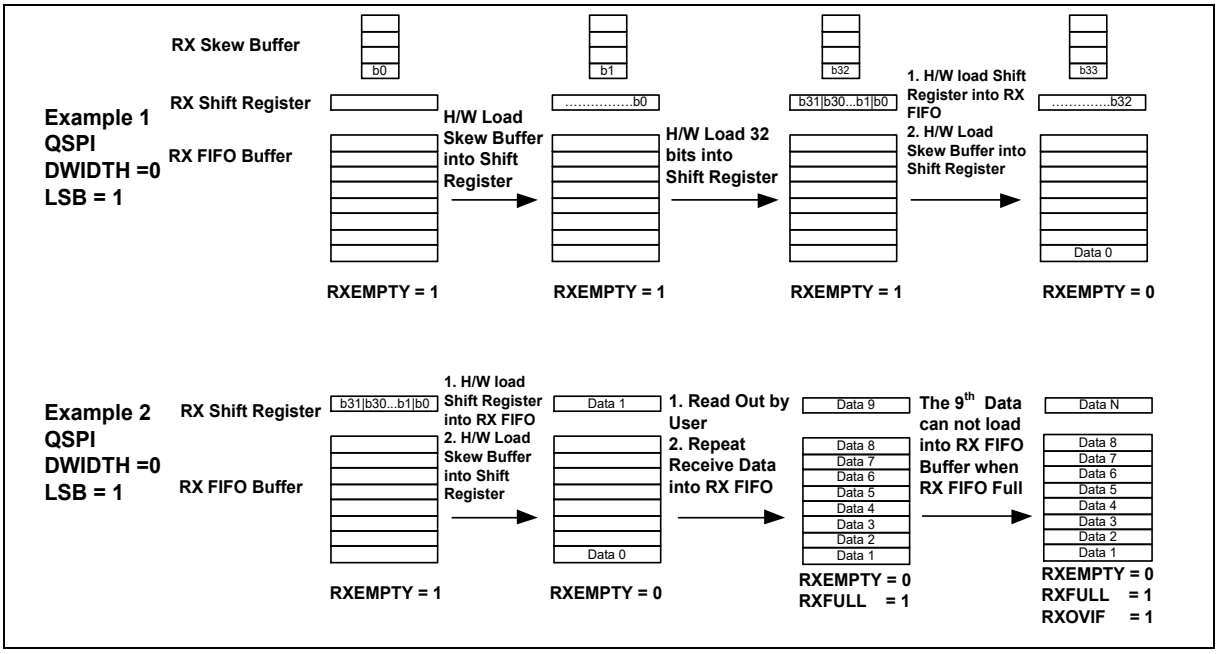


Figure 6.28-22 Receive FIFO Buffer Example

In Slave mode, during transmission operation, when data is written to the QSPIx_TX register by software, the data will be loaded into transmit FIFO buffer and the TXEMPTY (QSPIx_STATUS[16]) will be set to 0. The transmission will start when the slave device receives clock signal from master. Data can be written to QSPIx_TX register as long as the TXFULL (QSPIx_STATUS[17]) is 0. After all data have been drawn out by the QSPI transmission logic unit and the QSPIx_TX register is not updated by software, the TXEMPTY (QSPIx_STATUS[16]) will be set to 1.

If there is no any data written to the QSPIx_TX register, the transmit underflow interrupt flag, TXUFIF (QSPIx_STATUS[19]) will be set to 1 when the slave selection signal is active. The output data will be held by TXUFPOL (QSPIx_FIFOCTL[6]) setting during this transfer until the slave selection signal goes to inactive state. When the transmit underflow event occurs, the slave under run interrupt flag, SLVURIF (QSPIx_STATUS[7]), will be set to 1 as QSPIx_SS0 goes to inactive state.

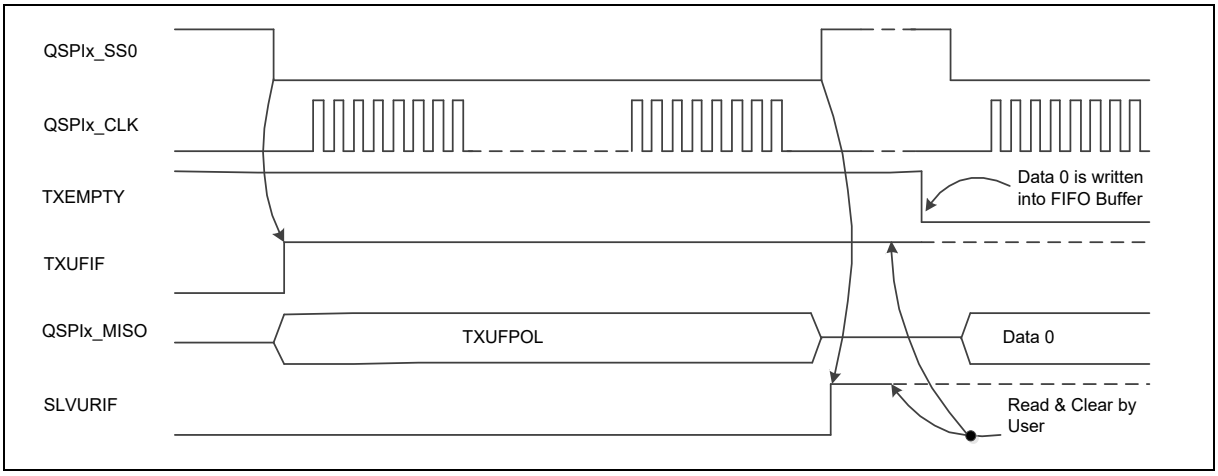


Figure 6.28-23 TX Underflow Event and Slave Under Run Event

In 2-bit transfer mode, the transmit data is loaded into shift register after 2 datum have been written into the TX FIFO buffer. It uses two shift registers and two 4-level skew buffers concurrently. The detail timing of 2-bit transfer mode, please refer to the section of Two-bit Transfer Mode.

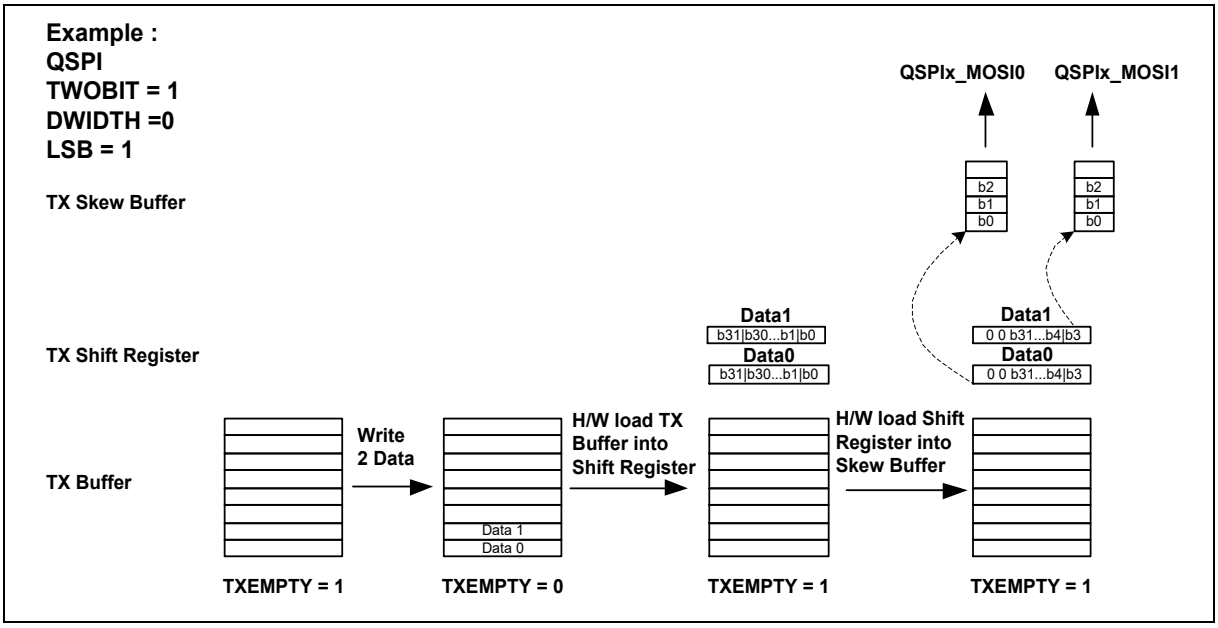


Figure 6.28-24 Two-bit Transfer Mode FIFO Buffer Example

In QSPI Slave 3-Wire mode, the first 2-bit data is un-predicted (keep on the level of last bit in previously transfer) if the data is written into TX FIFO among 3 peripheral clock cycles before the QSPI bus clock is presented. The other bits are held by TXUFPOL (QSPIx_FIFOCTL[6]) because there is TX underflow event. The written data will be transmitted in the next transfer.

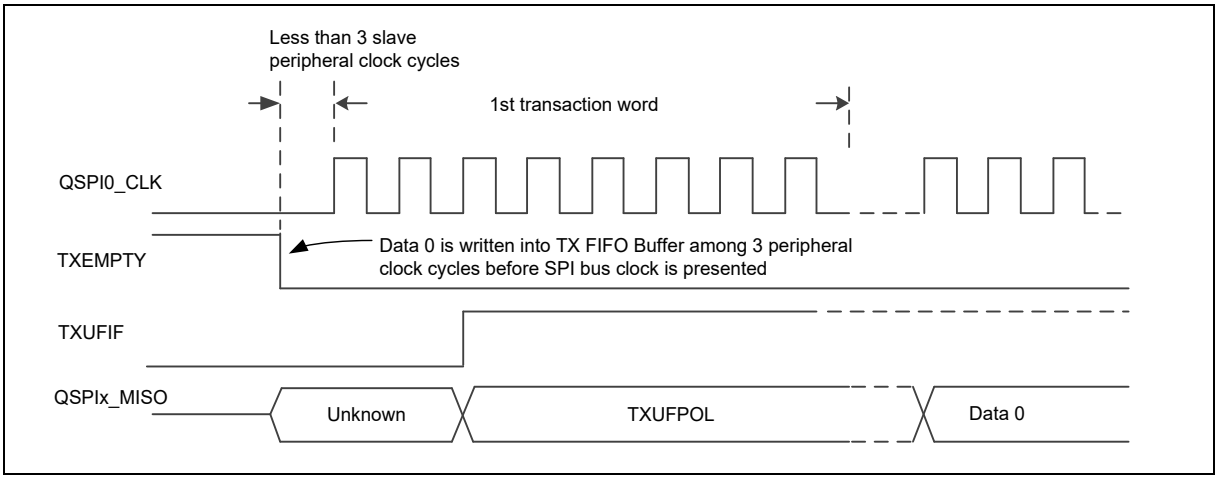


Figure 6.28-25 TX Underflow Event (QSPI0 Slave 3-Wire Mode Enabled)

In Slave mode, during receiving operation, the serial data is received from QSPIx_MOSI pin and stored to QSPIx_RX register. The reception mechanism is similar to Master mode reception operation. If the receive FIFO buffer contains 8 unread data, the RXFULL (QSPIx_STATUS[9]) will be set to 1 and the RXOVIF (QSPIx_STATUS[11]) will be set to 1 if there is more serial data received from QSPIx_MOSI and follow-up data will be dropped (refer to the Receive FIFO Buffer Example in Figure 6.28-22).

When the receive bit count is mismatched with the DWIDTH (QSPIx_CTL[12:8]) and the slave selection line goes to inactive state, the SLVBEIF (QSPIx_STATUS[6]) is set to 1. RX data will be written into RX FIFO and SLVBENUM (QSPIx_STATUS2[29:24]) will be updated when SLVBERX (QSPIx_FIFOCTL[10]) is enabled and error event of SPI slave bit count happened. The RX data will be dropped and SLVBENUM (QSPIx_STATUS2[29:24]) will be fixed to 0x0 when the control register SLVBERX (QSPIx_FIFOCTL[10]) is disabled and error event of SPI slave bit count happened. The

status register SLVBENUM (QSPIx_STATUS2[29:24]) indicates that effective bit number of uncompleted RX data when error event of SPI slave bit count happened. In Figure 6.28-26, the timing diagram of SPI slave bit count error and SLVBENUM is shown. SLVBENUM will be updated to 0x8 when SPI slave device receives 8 bits data and DWIDTH (QSPIx_CTL[12:8]) are set to 16 bits. The flag of SPI slave bit count error (SLVBEIF) generates pulse of RX FIFO write operation to write RX data from RX shift register to RX FIFO, and RX FIFO count (RXCNT) updates to 1.

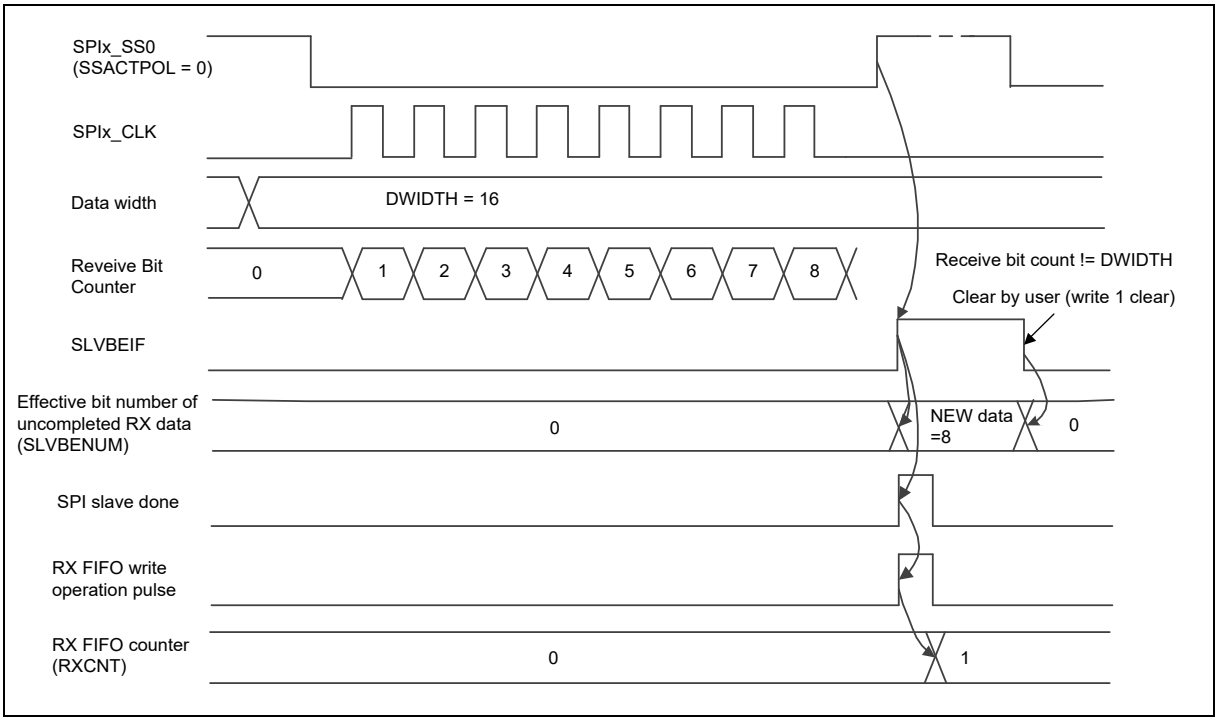


Figure 6.28-26 Slave Mode Bit Count Error and Effective Bit Number of Uncompleted RX Data

When the Slave select signal is active and the value of SLVTOCNT (QSPIx_SSCTL[31:16]) is not 0, the Slave time-out counter in the QSPI controller logic will start after the serial clock input. This counter will be cleared after one transaction done or the SLVTOCNT is set to 0. If the value of the time-out counter is equal to the value of SLVTOCNT before one transaction done, the slave time-out event occurs and the SLVTOIF (QSPIx_STATUS[5]) will be set to 1.

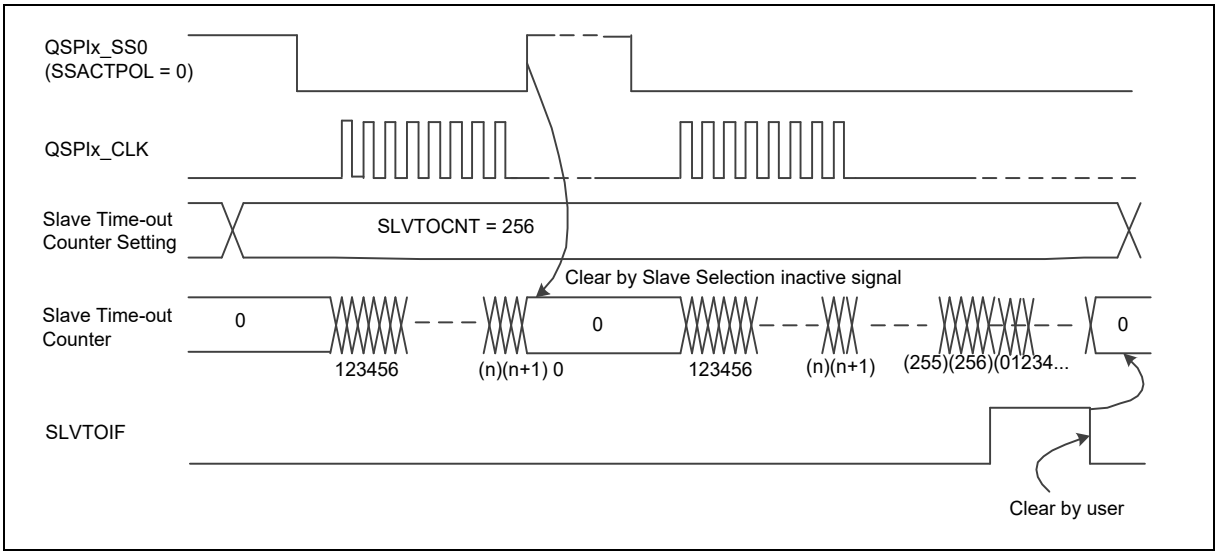


Figure 6.28-27 Slave Time-out Event

A receive time-out function is built-in in this controller. When the receive FIFO is not empty and no read operation in receive FIFO over 64 QSPI peripheral clock periods in Master mode or over 576 QSPI peripheral clock periods in Slave mode, the receive time-out occurs and the RXTOIF (QSPIx_STATUS[12]) will be set to 1. When the receive FIFO is read by user, the time-out status will be cleared automatically.

6.28.5.13 Interrupt

- QSPI unit transfer interrupt

  As the QSPI controller finishes a unit transfer, the unit transfer interrupt flag UNITIF (QSPIx_STATUS[1]) will be set to 1. The unit transfer interrupt event will generate an interrupt to CPU if the unit transfer interrupt enable bit UNITIEN (QSPIx_CTL[17]) is set. The unit transfer interrupt flag can be cleared only by writing 1 to it.

- QSPI slave selection active/inactive interrupt

  In Slave mode, the slave selection active/inactive interrupt flag, SSACTIF (QSPIx_STATUS[2]) and SSINAIF (QSPIx_STATUS[3]), will be set to 1 when the SPIEN (QSPIx_CTL[0]) and SLAVE (QSPIx_CTL[18]) are set to 1 and the slave selection signal goes to active/inactive state. The QSPI controller will issue an interrupt if the SSINAIEN (QSPIx_SSCTL[13]) or SSACTIEN (QSPIx_SSCTL[12]), are set to 1.

- Slave time-out interrupt

  In Slave mode, there is slave time-out function for user to know that there is serial clock input but one transaction is not finished over the period of SLVTOCNT (QSPIx_SSCTL[31:16]) basing on Slave peripheral clock.

  When the slave selection signal is active and the value of SLVTOCNT (QSPIx_SSCTL[31:16]) is not 0, the slave time-out counter in the QSPI controller logic will start after the serial clock input. This counter will be cleared after one transaction done or the SLVTOCNT (QSPIx_SSCTL[31:16]) is set to 0. If the value of the time-out counter is greater than or equal to the value of SLVTOCNT (QSPIx_SSCTL[31:16]) before one transaction done, the slave time-out event occurs and the SLVTOIF (QSPIx_STATUS[5]) will be set to 1. The QSPI controller will issue an interrupt if the SLVTOIEN (QSPIx_SSCTL[5]) is set to 1.

- Slave bit count error interrupt

  In Slave mode, if the transmit/receive bit count mismatch with the DWIDTH (QSPIx_CTL[12:8]) when the slave selection line goes to inactive state, the SLVBEIF (QSPIx_STATUS[6]) will be set to 1. The uncompleted transaction will be dropped from TX shift register. When the control register SLVBERX (QSPIx_FIFOCTL[10]) is disabled and SPI slave bit count error event happened, the uncompleted transaction data will be dropped from RX shift registers. When control register SLVBERX (QSPIx_FIFOCTL[10]) is enabled and SPI slave bit count error event happened, the uncompleted transaction data will be written from RX shift registers into RX FIFO. The QSPI controller will issue an interrupt if the SLVBEIEN (QSPIx_SSCTL[8]) is set to 1.

  Note: If the slave selection signal is active but there is no any serial clock input, the SLVBEIF (QSPIx_STATUS[6]) will be set to 1 when the slave selection signal goes to inactive state.

- TX underflow interrupt

  In QSPI Slave mode, if there is no any data is written to the QSPIx_TX register, the TXUFIF (QSPIx_STATUS[19]) will be set to 1 when the slave selection signal is active. The QSPI controller will issue a TX underflow interrupt if the TXUFIEN (QSPIx_FIFOCTL[7]) is set to 1.

Note: If underflow event occurs in QSPI Slave mode, there are two conditions which make QSPI Slave mode return to idle state and then goes for next transfer: (1) set TXRST to 1 (2) slave select signal is changed to inactive state while SLV3WIRE=0.

- Slave TX under run interrupt

  If the TX underflow event occurs, the SLVURIF (QSPIx_STATUS[7]) will be set to 1 when QSPIx_SS goes to inactive state. The QSPI controller will issue a TX under run interrupt if the SLVURIEN (QSPIx_SSCTL[9]) is set to 1.

  Note: In Slave 3-Wire mode, the slave selection signal is considered active all the time so that user shall poll the TXUFIF (QSPIx_STATUS[19]) to know if there is TX underflow event or not.

- Receive Overrun interrupt

  In Slave mode, if the receive FIFO buffer contains 8 unread data, the RXFULL (QSPIx_STATUS[9]) will be set to 1 and the RXOVIF (QSPIx_STATUS[11]) will be set to 1 if there is more serial data received from QSPI bus and follow-up data will be dropped. The QSPI controller will issue an interrupt if the RXOVIEN (QSPIx_FIFOCTL[5]) is set to 1.

- Receive FIFO time-out interrupt

  If there is a received data in the FIFO buffer and it is not read by software over 64 QSPI peripheral clock periods in Master mode or over 576 QSPI peripheral clock periods in Slave mode, it will send a RX time-out interrupt to the system if the RX time-out interrupt enable bit, RXTOIEN (QSPIx_FIFOCTL[4]), is set to 1.

- Transmit FIFO interrupt

  In FIFO mode, if the valid data count of the transmit FIFO buffer is less than or equal to the setting value of TXTH (QSPIx_FIFOCTL[30:28]), the transmit FIFO interrupt flag TXTHIF (QSPIx_STATUS[18]) will be set to 1. The QSPI controller will generate a transmit FIFO interrupt to the system if the transmit FIFO interrupt enable bit, TXTHIEN (QSPIx_FIFOCTL[3]), is set to 1.

- Receive FIFO interrupt

  In FIFO mode, if the valid data count of the receive FIFO buffer is larger than the setting value of RXTH (QSPIx_FIFOCTL[26:24]), the receive FIFO interrupt flag RXTHIF (QSPIx_STATUS[10]) will be set to 1. The QSPI controller will generate a receive FIFO interrupt to the system if the receive FIFO interrupt enable bit, RXTHIEN (QSPIx_FIFOCTL[2]), is set to 1.

6.28.6 Timing Diagram

The active state of slave selection signal can be defined by setting the SSACTPOL (QSPIx_SSCTL[2]). The QSPI clock which is in idle state can be configured as high or low state by setting the CLKPOL (QSPIx_CTL[3]). It also provides the bit length of a transaction word in DWIDTH (QSPIx_CTL[12:8]), and transmitting/receiving data from MSB or LSB first in LSB (QSPIx_CTL[13]). User can also select which edge of QSPI clock to transmit/receive data in TXNEG/RXNEG (QSPIx_CTL[2:1]). Four QSPI timing diagrams for master/slave operations and the related settings are shown below.

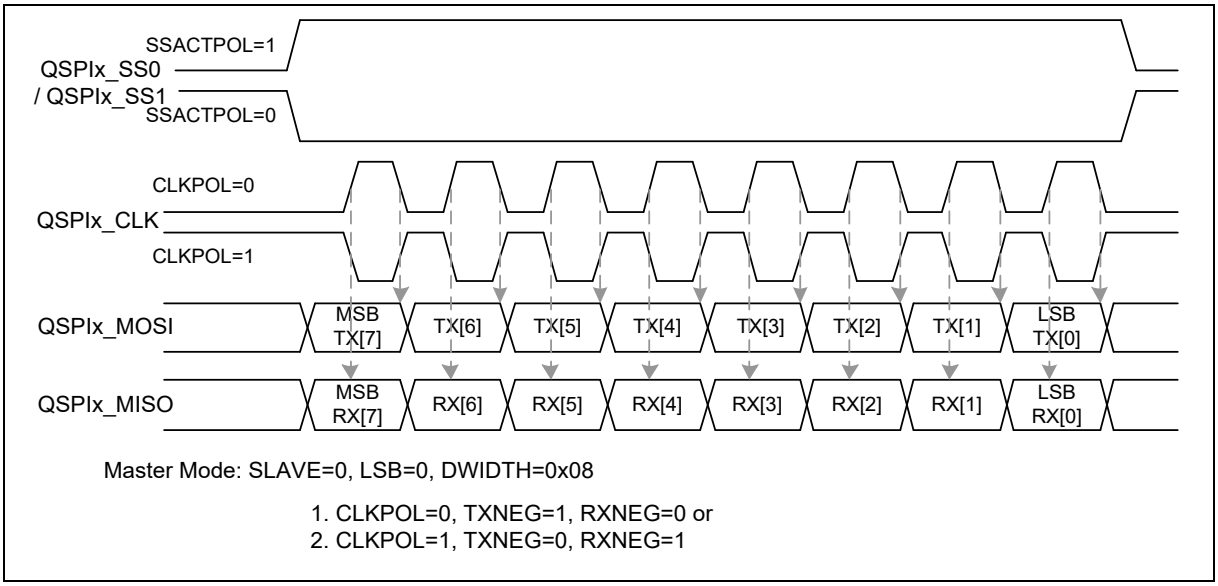


Figure 6.28-28 QSPI Timing in Master Mode

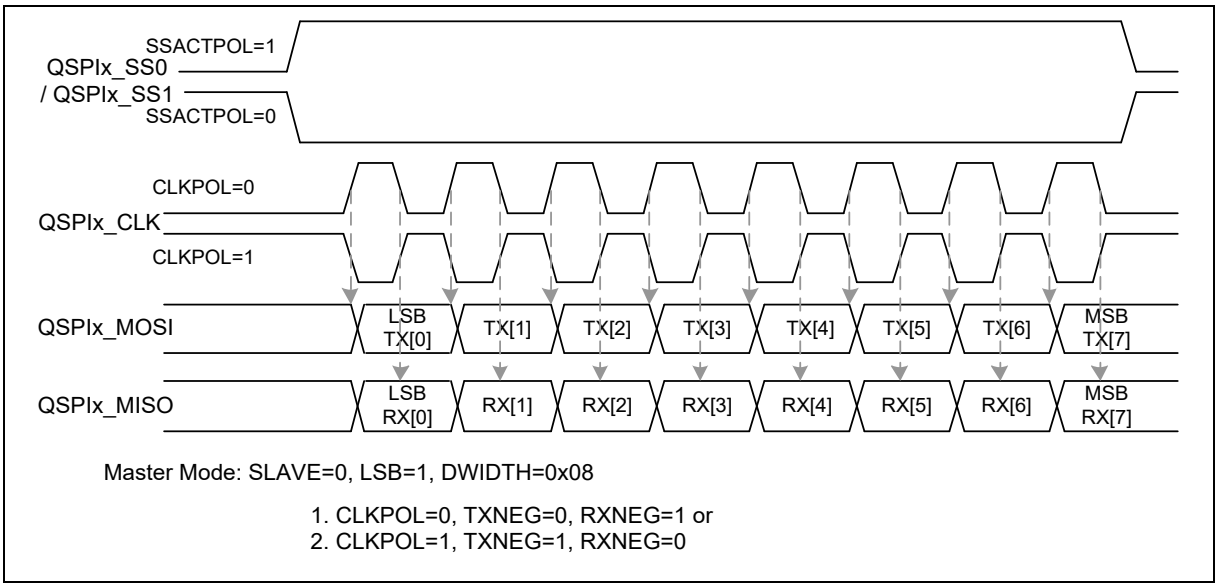


Figure 6.28-29 QSPI Timing in Master Mode (Alternate Phase of QSPIx_CLK)

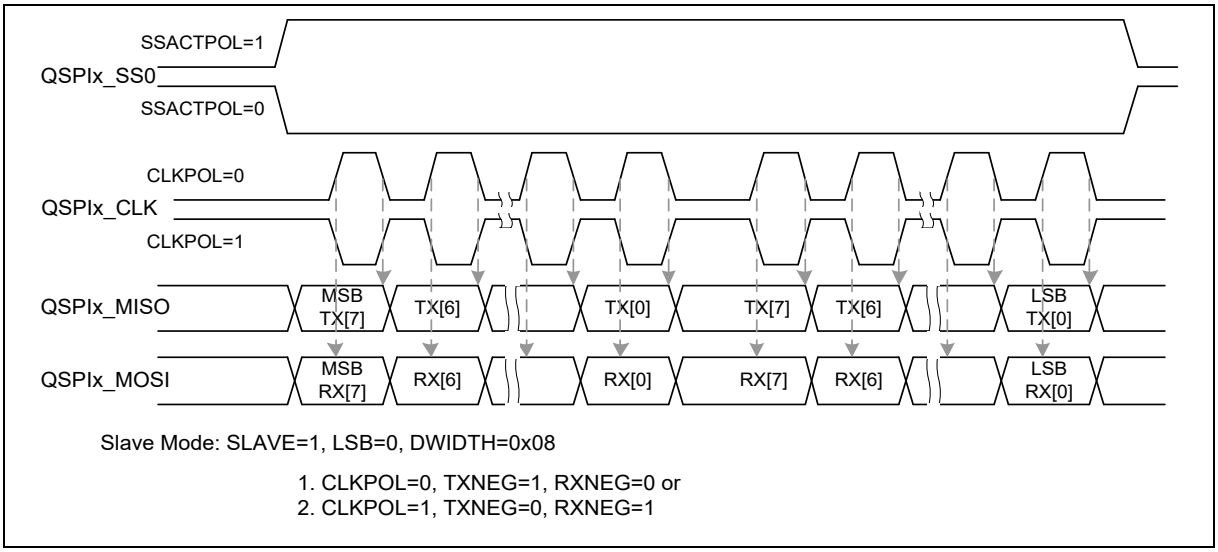


Figure 6.28-30 QSPI Timing in Slave Mode

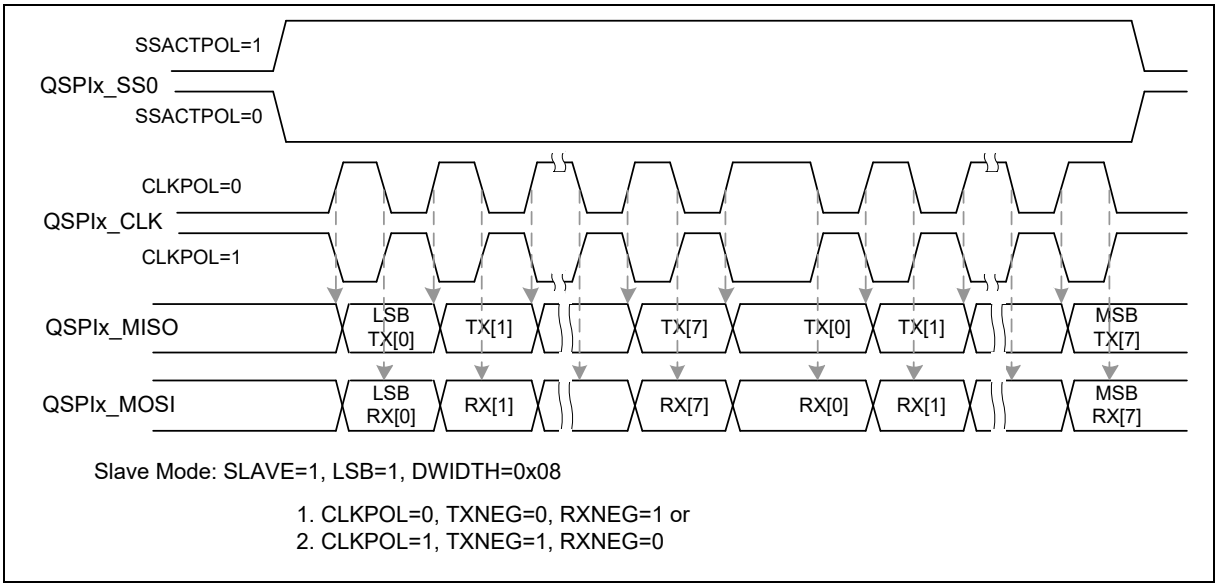


Figure 6.28-31 QSPI Timing in Slave Mode (Alternate Phase of QSPIx_CLK)

6.28.7 Programming Examples

Example 1:

The QSPI controller is set as a full-duplex master to access an off-chip slave device with the following specifications:

- Data bit is latched on positive edge of QSPI bus clock.
- Data bit is driven on negative edge of QSPI bus clock.
- Data is transferred from MSB first.
- QSPI bus clock is idle at low state.
- Only one byte of data to be transmitted/received in a transaction.

- Uses the first QSPI slave select pin to connect with an off-chip slave device. The slave selection signal is active low.

The operation flow is as follows:

1. Set DIVIDER (QSPIx_CLKDIV [8:0]) to determine the output frequency of QSPI clock.
2. Write the QSPIx_SSCTL register a proper value for the related settings of Master mode:
   1) Clear AUTOSS (QSPIx_SSCTL[3]) to 0 to disable the Automatic Slave Selection function.
   2) Configure slave selection signal as active low by clearing SSACTPOL (QSPIx_SSCTL[2]) to 0.
   3) Enable slave selection signal by setting SS0 (QSPIx_SSCTL[0]) or SS1 (QSPIx_SSCTL[1]) to 1 to activate the off-chip slave device.
3. Write the related settings into the QSPIx_CTL register to control the QSPI master actions.
   1) Configure this QSPI controller as master device by setting SLAVE (QSPIx_CTL[18]) to 0.
   2) Force the QSPI clock idle state at low by clearing CLKPOL (QSPIx_CTL[3]) to 0.
   3) Select data transmitted on negative edge of QSPI bus clock by setting TXNEG (QSPIx_CTL[2]) to 1.
   4) Select data latched on positive edge of QSPI bus clock by clearing RXNEG (QSPIx_CTL[1]) to 0.
   5) Set the bit length of a transaction as 8-bit in DWIDTH bit field (QSPIx_CTL[12:8] = 0x08).
   6) Set MSB transfer first by clearing LSB (QSPIx_CTL[13]) to 0.
4. Set SPIEN (QSPIx_CTL[0]) to 1 to enable the data transfer with the QSPI interface.
5. If this QSPI master attempts to transmit (write) one byte data to the off-chip slave device, write the byte data that will be transmitted into the QSPIx_TX register.
6. Waiting for QSPI interrupt if the UNITIEN (QSPIx_CTL[17]) is set to 1, or just polling the unit transfer interrupt flag UNITIF (QSPIx_STATUS[1]).
7. Read out the received one byte data from QSPIx_RX register.
8. Go to 5) to continue another data transfer or set SS0 (QSPIx_SSCTL[0]) or SS1 (QSPIx_SSCTL[1]) to 0 to inactivate the off-chip slave device.

Example 2:

The QSPI controller is set as a full-duplex slave device and connects with an off-chip master device. The off-chip master device communicates with the on-chip QSPI slave controller through the QSPI interface with the following specifications:

- Data bit is latched on positive edge of QSPI bus clock.
- Data bit is driven on negative edge of QSPI bus clock.
- Data is transferred from LSB first.
- QSPI bus clock is idle at high state.
- Only one byte of data to be transmitted/received in a transaction.
- Slave selection signal is active high.

The operation flow is as follows:

1. Write the QSPIx_SSCTL register a proper value for the related settings of Slave mode.
2. Select high level for the input of slave selection signal by setting SSACTPOL (QSPIx_SSCTL[2]) to 1.

3. Write the related settings into the QSPIx_CTL register to control this QSPI slave actions
   1) Set the QSPI controller as slave device by setting SLAVE (QSPIx_CTL[18]) to 1.
   2) Select the QSPI clock idle state at high by setting CLKPOL (QSPIx_CTL[3]) to 1.
   3) Select data transmitted on negative edge of QSPI bus clock by setting TXNEG (QSPIx_CTL[2]) to 1.
   4) Select data latched on positive edge of QSPI bus clock by clearing RXNEG (QSPIx_CTL[1]) to 0.
   5) Set the bit length of a transaction as 8-bit in DWIDTH bit field (QSPIx_CTL[12:8] = 0x08).
4. Set LSB transfer first by setting LSB (QSPIx_CTL[13]) to 1.
5. Set the SPIEN (QSPIx_CTL[0]) to 1. Wait for the slave select trigger input and QSPI clock input from the off-chip master device to start the data transfer.
6. If this QSPI slave attempts to transmit (be read) one byte data to the off-chip master device, write the byte data that will be transmitted into the QSPIx_TX register.
7. If this QSPI slave just only attempts to receive (be written) one byte data from the off-chip master device and does not care what data will be transmitted, the QSPIx_TX register does not need to be updated by software.
8. Waiting for QSPI interrupt if the UNITIEN (QSPIx_CTL[17]) is set to 1, or just polling the unit transfer interrupt flag UNITIF (QSPIx_STATUS[1]).
9. Read out the received one byte data from QSPIx_RX register.

Go to 7 to continue another data transfer or stop data transfer.

6.28.8 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
QSPI Base Address: **QSPIx_BA = 0x4068_0000 + (0x0001_0000 * x)** **x=0, 1**				
QSPIx_CTL	QSPIx_BA+0x00	R/W	QSPI Control Register	0x0000_0034
QSPIx_CLKDIV	QSPIx_BA+0x04	R/W	QSPI Clock Divider Register	0x0000_0000
QSPIx_SSCTL	QSPIx_BA+0x08	R/W	QSPI Slave Select Control Register	0x0000_0000
QSPIx_PDMACTL	QSPIx_BA+0x0C	R/W	QSPI PDMA Control Register	0x0000_0000
QSPIx_FIFOCTL	QSPIx_BA+0x10	R/W	QSPI FIFO Control Register	0x4400_0000
QSPIx_STATUS	QSPIx_BA+0x14	R/W	QSPI Status Register	0x0005_0110
QSPIx_STATUS2	QSPIx_BA+0x18	R	QSPI Status2 Register	0x0000_0000
QSPIx_FIFOCTL2	QSPIx_BA+0x1C	R/W	QSPI FIFO2 Control Register	0x0404_0000
QSPIx_TX	QSPIx_BA+0x20	W	QSPI Data Transmit Register	0x0000_0000
QSPIx_RX	QSPIx_BA+0x30	R	QSPI Data Receive Register	0x0000_0000
QSPIx_INTERNAL	QSPIx_BA+0x48	R/W	QSPI Internal Control Register	0x0000_0000

6.28.9 Register Description

QSPI Control Register (QSPIx_CTL)

Register	Offset	R/W	Description	Reset Value
QSPIx_CTL	QSPIx_BA+0x00	R/W	QSPI Control Register	0x0000_0034

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DTREN	QUADIOEN	DUALIOEN	DATDIR	REORDER	SLAVE	UNITIEN	TWOBIT
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RXONLY	HALFDPX	LSB	DWIDTH				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SUSPITV				CLKPOL	TXNEG	RXNEG	SPIEN

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23]	**DTREN**	**Transmit Double Transfer Rate Mode Enable Bit** In master mode, QSPI DTR function usually uses for SPI Flash controller, which this SPI Flash supports DTR commands as 0xED, 0xBD, and 0x0D. 0 = DTR mode Disabled. 1 = DTR mode Enabled. **Note:** QSPI master mode supports DTR mode, and QSPI slave mode does not support this mode.
[22]	**QUADIOEN**	**Quad I/O Mode Enable Bit** 0 = Quad I/O mode Disabled. 1 = Quad I/O mode Enabled.
[21]	**DUALIOEN**	**Dual I/O Mode Enable Bit** 0 = Dual I/O mode Disabled. 1 = Dual I/O mode Enabled.
[20]	**DATDIR**	**Data Port Direction Control** This bit is used to select the data input/output direction in half-duplex transfer and Dual/Quad transfer 0 = QSPI data is input direction. 1 = QSPI data is output direction.
[19]	**REORDER**	**Byte Reorder Function Enable Bit** 0 = Byte Reorder function Disabled. 1 = Byte Reorder function Enabled. A byte suspend interval will be inserted among each byte. The period of the byte suspend interval depends on the setting of SUSPITV. **Note:** Byte Reorder function is only available if DWIDTH is defined as 16, 24, and 32 bits.
[18]	**SLAVE**	**Slave Mode Control** 0 = Master mode. 1 = Slave mode.

[17]	**UNITIEN**	**Unit Transfer Interrupt Enable Bit** 0 = QSPI unit transfer interrupt Disabled. 1 = QSPI unit transfer interrupt Enabled.
[16]	**TWOBIT**	**2-bit Transfer Mode Enable Bit** 0 = 2-bit transfer mode Disabled. 1 = 2-bit transfer mode Enabled. **Note:** When 2-bit transfer mode is enabled, the first serial transmitted bit data is from the first FIFO buffer data, and the 2nd serial transmitted bit data is from the second FIFO buffer data. As the same as transmitted function, the first received bit data is stored into the first FIFO buffer and the 2nd received bit data is stored into the second FIFO buffer at the same time.
[15]	**RXONLY**	**Receive-only Mode Enable Bit for Master Only** This bit field is only available in Master mode. In receive-only mode, QSPI Master will generate QSPI bus clock continuously for receiving data bit from SPI slave device and assert the BUSY status. 0 = Receive-only mode Disabled. 1 = Receive-only mode Enabled.
[14]	**HALFDPX**	**QSPI Half-duplex Transfer Enable Bit** This bit is used to select full-duplex or half-duplex for QSPI transfer. The bit field DATDIR (QSPIx_CTL[20]) can be used to set the data direction in half-duplex transfer. 0 = QSPI operates in full-duplex transfer Disabled. 1 = QSPI operates in half-duplex transfer Enabled.
[13]	**LSB**	**Send LSB First** 0 = The MSB, which bit of transmit/receive register depends on the setting of DWIDTH, is transmitted/received first. 1 = The LSB, bit 0 of the QSPIx TX register, is sent first to the QSPI data output pin, and the first bit received from the QSPI data input pin will be put in the LSB position of the RX register (bit 0 of QSPIx_RX).
[12:8]	**DWIDTH**	**Data Width** This field specifies how many bits can be transmitted / received in one transaction. The minimum bit length is 8 bits and can up to 32 bits. DWIDTH = 0x08 …. 8 bits. DWIDTH = 0x09 …. 9 bits. …… DWIDTH = 0x1F …. 31 bits. DWIDTH = 0x00 …. 32 bits. **Note 1:** This bit field will decide the depth of TX/RX FIFO configuration in SPI mode**.** Therefore, changing this bit field will clear TX/RX FIFO by hardware automatically. **Note 2:** When user changes data width, the clear functions of TXFIFO and RXFIFO will be triggered by hardware automatically. User can read status register FIFOCLR (QSPI_STATUS[22]) to check status of clear function of TXFIFO and RXFIFO. After status register FIFOCLR is 0, user can write TXFIFO or read RXFIFO correctly.
[7:4]	**SUSPITV**	**Suspend Interval for Master Only** The four bits provide configurable suspend interval between two successive transmit/receive transaction in a transfer. The definition of the suspend interval is the interval between the last clock edge of the preceding transaction word and the first clock edge of the following transaction word. The default value is 0x3. The period of the suspend interval is obtained according to the following equation. (SUSPITV[3:0] + 0.5) * period of QSPI_CLK clock cycle Example: SUSPITV = 0x0 …. 0.5 QSPI_CLK clock cycle. SUSPITV = 0x1 …. 1.5 QSPI_CLK clock cycle. …… SUSPITV = 0xE …. 14.5 QSPI_CLK clock cycle.

		SUSPITV = 0xF …. 15.5 QSPI_CLK clock cycle. **Note:** In TX DTR mode, SUSPITV equals to 0x0.
[3]	**CLKPOL**	**Clock Polarity** 0 = QSPI bus clock is idle low. 1 = QSPI bus clock is idle high.
[2]	**TXNEG**	**Transmit on Negative Edge** 0 = Transmitted data output signal is changed on the rising edge of QSPI bus clock. 1 = Transmitted data output signal is changed on the falling edge of QSPI bus clock. **Note:** In TX DTR mode, TXNEG equals to CLKPOL (QSPI_CTL[3]).
[1]	**RXNEG**	**Receive on Negative Edge** 0 = Received data input signal is latched on the rising edge of QSPI bus clock. 1 = Received data input signal is latched on the falling edge of QSPI bus clock.
[0]	**SPIEN**	**QSPI Transfer Control Enable Bit** In Master mode, the transfer will start when there is data in the FIFO buffer after this bit is set to 1. In Slave mode, this device is ready to receive data when this bit is set to 1. 0 = Transfer control Disabled. 1 = Transfer control Enabled. **Note:** Before changing the configurations of QSPIx_CTL, QSPIx_CLKDIV, QSPIx_SSCTL and QSPIx_FIFOCTL, registers, user shall clear the SPIEN (QSPIx_CTL[0]) and confirm the SPIENSTS (QSPIx_STATUS[15]) is 0.

QSPI Clock Divider Register (QSPIx_CLKDIV)

Register	Offset	R/W	Description	Reset Value
QSPIx_CLKDIV	QSPIx_BA+0x04	R/W	QSPI Clock Divider Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							DIVIDER
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DIVIDER							

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8:0]	**DIVIDER**	**Clock Divider** The value in this field is the frequency divider for generating the peripheral clock, f_{spi_eclk}, and the QSPI bus clock of QSPI Master. The frequency is obtained according to the following equation. $f_{spi_eclk} = \frac{f_{spi_clock_src}}{(DIVIDER+1)}$ where $f_{spi_clock_src}$ is the peripheral clock source, which is defined in the clock control register, CLK_CLKSEL4. **Note 1:** The time interval must be larger than or equal 5 peripheral clock cycles between releasing QSPI IP software reset and setting this clock divider register. **Note 2:** The setting values of this clock divider "DIVIDER" are only 1, 3, 5, 7,…, 2*N+1, where the variable N is equal to 0,1,2,3,4,…

Note: DIVIDER should be set carefully because the peripheral clock frequency must be slower than or equal to system frequency.

QSPI Slave Select Control Register (QSPIx_SSCTL)

Register	Offset	R/W	Description	Reset Value
QSPIx_SSCTL	QSPIx_BA+0x08	R/W	QSPI Slave Select Control Register	0x0000_0000

31	30	29	28	27	26	25	24
SLVTOCNT							
23	22	21	20	19	18	17	16
SLVTOCNT							
15	14	13	12	11	10	9	8
Reserved		SSINAIEN	SSACTIEN	Reserved		SLVURIEN	SLVBEIEN
7	6	5	4	3	2	1	0
Reserved	SLVTORST	SLVTOIEN	SLV3WIRE	AUTOSS	SSACTPOL	SS1	SS0

Bits	Description	
[31:16]	**SLVTOCNT**	**Slave Mode Time-out Period** In Slave mode, these bits indicate the time-out period when there is bus clock input during slave select active. The clock source of the time-out counter is Slave peripheral clock. If the value is 0, it indicates the slave mode time-out function is disabled.
[15:14]	**Reserved**	Reserved.
[13]	**SSINAIEN**	**Slave Select Inactive Interrupt Enable Bit** 0 = Slave select inactive interrupt Disabled. 1 = Slave select inactive interrupt Enabled.
[12]	**SSACTIEN**	**Slave Select Active Interrupt Enable Bit** 0 = Slave select active interrupt Disabled. 1 = Slave select active interrupt Enabled.
[11:10]	**Reserved**	Reserved.
[9]	**SLVURIEN**	**Slave Mode TX Under Run Interrupt Enable Bit** 0 = Slave mode TX under run interrupt Disabled. 1 = Slave mode TX under run interrupt Enabled.
[8]	**SLVBEIEN**	**Slave Mode Bit Count Error Interrupt Enable Bit** 0 = Slave mode bit count error interrupt Disabled. 1 = Slave mode bit count error interrupt Enabled.
[7]	**Reserved**	Reserved.
[6]	**SLVTORST**	**Slave Mode Time-out Reset Control** 0 = When Slave mode time-out event occurs, the TX and RX control circuit will not be reset. 1 = When Slave mode time-out event occurs, the TX and RX control circuit will be reset by hardware.
[5]	**SLVTOIEN**	**Slave Mode Time-out Interrupt Enable Bit** 0 = Slave mode time-out interrupt Disabled. 1 = Slave mode time-out interrupt Enabled.
[4]	**SLV3WIRE**	**Slave 3-wire Mode Enable Bit**

		In Slave 3-wire mode, the QSPI controller can work with 3-wire interface including QSPIx_CLK, QSPIx_MISO and QSPIx_MOSI pins. 0 = 4-wire bi-direction interface Disabled. 1 = 3-wire bi-direction interface Enabled.
[3]	**AUTOSS**	**Automatic Slave Selection Function Enable Bit for Master Only** 0 = Automatic slave selection function Disabled. Slave selection signal will be asserted/de-asserted according to SS0 (QSPIx_SSCTL[0]) or SS1 (QSPIx_SSCTL[1]). 1 = Automatic slave selection function Enabled.
[2]	**SSACTPOL**	**Slave Selection Active Polarity** This bit defines the active polarity of slave selection signal (QSPIx_SS0 and QSPIx_SS1). 0 = The slave selection signal QSPIx_SS0/ QSPIx_SS1 is active low. 1 = The slave selection signal QSPIx_SS0/ QSPIx_SS1 is active high.
[1]	**SS1**	**Slave Selection 1 Control for Master Only** If AUTOSS bit is cleared to 0, 0 = Set the QSPIx_SS1 line to inactive state. 1 = Set the QSPIx_SS1 line to active state. If the AUTOSS bit is set to 1, 0 = Keep the QSPIx_SS1 line at inactive state. 1 = QSPIx_SS1 line will be automatically driven to active state for the duration of data transfer, and will be driven to inactive state for the rest of the time. The active state of QSPIx_SS1 is specified in SSACTPOL (QSPIx_SSCTL[2]).
[0]	**SS0**	**Slave Selection 0 Control for Master Only** If AUTOSS bit is cleared to 0, 0 = set the QSPIx_SS0 line to inactive state. 1 = set the QSPIx_SS0 line to active state. If the AUTOSS bit is set to 1, 0 = Keep the QSPIx_SS0 line at inactive state. 1 = QSPIx_SS0 line will be automatically driven to active state for the duration of data transfer, and will be driven to inactive state for the rest of the time. The active state of QSPIx_SS0 is specified in SSACTPOL (QSPIx_SSCTL[2]).

QSPI PDMA Control Register (QSPIx_PDMACTL)

Register	Offset	R/W	Description	Reset Value
QSPIx_PDMACTL	QSPIx_BA+0x0C	R/W	QSPI PDMA Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					PDMARST	RXPDMAEN	TXPDMAEN

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**PDMARST**	**PDMA Reset** 0 = No effect. 1 = Reset the PDMA control logic of the QSPI controller. This bit will be automatically cleared to 0.
[1]	**RXPDMAEN**	**Receive PDMA Enable Bit** 0 = Receive PDMA function Disabled. 1 = Receive PDMA function Enabled.
[0]	**TXPDMAEN**	**Transmit PDMA Enable Bit** 0 = Transmit PDMA function Disabled. 1 = Transmit PDMA function Enabled. **Note 1:** In QSPI Master mode with full duplex transfer, if both TX and RX PDMA functions are enabled, RX PDMA function cannot be enabled prior to TX PDMA function. User can enable TX PDMA function firstly or enable both functions simultaneously. **Note 2:** In QSPI Master mode with full duplex transfer, if both TX and RX PDMA functions are enabled, TX PDMA function cannot be disabled prior to RX PDMA function. User can disable RX PDMA function firstly or disable both functions simultaneously.

QSPI FIFO Control Register (QSPIx_FIFOCTL)

Register	Offset	R/W	Description	Reset Value
QSPIx_FIFOCTL	QSPIx_BA+0x10	R/W	QSPI FIFO Control Register	0x4400_0000

31	30	29	28	27	26	25	24
Reserved	TXTH			Reserved	RXTH		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					SLVBERX	TXFBCLR	RXFBCLR
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TXUFIEN	TXUFPOL	RXOVIEN	RXTOIEN	TXTHIEN	RXTHIEN	TXRST	RXRST

Bits	Description	
[31]	**Reserved**	Reserved.
[30:28]	**TXTH**	**Transmit FIFO Threshold** If the valid data count of the transmit FIFO buffer is less than or equal to the TXTH setting, the TXTHIF bit will be set to 1, else the TXTHIF bit will be cleared to 0.
[27]	**Reserved**	Reserved.
[26:24]	**RXTH**	**Receive FIFO Threshold** If the valid data count of the receive FIFO buffer is larger than the RXTH setting, the RXTHIF bit will be set to 1, else the RXTHIF bit will be cleared to 0.
[23:11]	**Reserved**	Reserved.
[10]	**SLVBERX**	**RX FIFO Write Data Enable Bit When Slave Mode Bit Count Error** 0 = Uncompleted RX data will be dropped from RX FIFO when bit count error event happen in SPI slave mode. 1 = Uncompleted RX data will be written into RX FIFO when bit count error event happen in SPI slave mode. User can read SLVBENUM (QSPIx_STATUS2[29:24]) to know that the effective bit number of uncompleted RX data when SPI slave bit count error happened.
[9]	**TXFBCLR**	**Transmit FIFO Buffer Clear** 0 = No effect. 1 = Clear transmit FIFO pointer. The TXFULL bit will be cleared to 0 and the TXEMPTY bit will be set to 1. This bit will be cleared to 0 by hardware about 1 system clock after it is set to 1. **Note:** The TX shift register will not be cleared.
[8]	**RXFBCLR**	**Receive FIFO Buffer Clear** 0 = No effect. 1 = Clear receive FIFO pointer. The RXFULL bit will be cleared to 0 and the RXEMPTY bit will be set to 1. This bit will be cleared to 0 by hardware about 1 system clock after it is set to 1. **Note:** The RX shift register will not be cleared.
[7]	**TXUFIEN**	**TX Underflow Interrupt Enable Bit** When TX underflow event occurs in Slave mode, TXUFIF (QSPIx_STATUS[19]) will be set to 1. This bit is used to enable the TX underflow interrupt.

		0 = Slave TX underflow interrupt Disabled. 1 = Slave TX underflow interrupt Enabled.
[6]	**TXUFPOL**	**TX Underflow Data Polarity** 0 = The QSPI data out is keep 0 if there is TX underflow event in Slave mode. 1 = The QSPI data out is keep 1 if there is TX underflow event in Slave mode. **Note 1:** The TX underflow event occurs if there is no any data in TX FIFO when the slave selection signal is active. **Note 2:** When TX underflow event occurs, QSPIx_MISO pin state will be determined by this setting even though TX FIFO is not empty afterward. Data stored in TX FIFO will be sent through QSPIx_MISO pin in the next transfer frame.
[5]	**RXOVIEN**	**Receive FIFO Overrun Interrupt Enable Bit** 0 = Receive FIFO overrun interrupt Disabled. 1 = Receive FIFO overrun interrupt Enabled.
[4]	**RXTOIEN**	**Receive Time-out Interrupt Enable Bit** 0 = Receive time-out interrupt Disabled. 1 = Receive time-out interrupt Enabled.
[3]	**TXTHIEN**	**Transmit FIFO Threshold Interrupt Enable Bit** 0 = TX FIFO threshold interrupt Disabled. 1 = TX FIFO threshold interrupt Enabled.
[2]	**RXTHIEN**	**Receive FIFO Threshold Interrupt Enable Bit** 0 = RX FIFO threshold interrupt Disabled. 1 = RX FIFO threshold interrupt Enabled.
[1]	**TXRST**	**Transmit Reset** 0 = No effect. 1 = Reset transmit FIFO pointer and transmit circuit. The TXFULL bit will be cleared to 0 and the TXEMPTY bit will be set to 1. This bit will be cleared to 0 by hardware about 3 system clock cycles + 2 peripheral clock cycles after it is set to 1. User can read TXRXRST (QSPIx_STATUS[23]) to check if reset is accomplished or not. **Note:** If TX underflow event occurs in QSPI Slave mode, this bit can be used to make QSPI return to idle state.
[0]	**RXRST**	**Receive Reset** 0 = No effect. 1 = Reset receive FIFO pointer and receive circuit. The RXFULL bit will be cleared to 0 and the RXEMPTY bit will be set to 1. This bit will be cleared to 0 by hardware about 3 system clock cycles + 2 peripheral clock cycles after it is set to 1. User can read TXRXRST (QSPIx_STATUS[23]) to check if reset is accomplished or not.

QSPI Status Register (QSPIx_STATUS)

Register	Offset	R/W	Description	Reset Value
QSPIx_STATUS	QSPIx_BA+0x14	R/W	QSPI Status Register	0x0005_0110

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
TXRXRST	FIFOCLR	Reserved		TXUFIF	TXTHIF	TXFULL	TXEMPTY
15	14	13	12	11	10	9	8
SPIENSTS	Reserved		RXTOIF	RXOVIF	RXTHIF	RXFULL	RXEMPTY
7	6	5	4	3	2	1	0
SLVURIF	SLVBEIF	SLVTOIF	SSLINE	SSINAIF	SSACTIF	UNITIF	BUSY

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23]	**TXRXRST**	**TX or RX Reset Status (Read Only)** 0 = The reset function of TXRST or RXRST is done. 1 = Doing the reset function of TXRST or RXRST. **Note:** Both the reset operations of TXRST and RXRST need 3 system clock cycles + 2 peripheral clock cycles. User can check the status of this bit to monitor the reset function is doing or done.
[22]	**FIFOCLR**	**TXFIFO or RXFIFO Clear Status (Read Only)** User can check the status of this bit to monitor the clear function of TXFIFO or RXFIFO is doing or done. 0 = The clear function of TXFBCLR or RXFBCLR is done. 1 = Doing the clear function of TXFBCLR or RXFBCLR.
[21:20]	**Reserved**	Reserved.
[19]	**TXUFIF**	**TX Underflow Interrupt Flag** When the TX underflow event occurs, this bit will be set to 1, the state of data output pin depends on the setting of TXUFPOL. 0 = No effect. 1 = No data in Transmit FIFO and TX shift register when the slave selection signal is active. **Note 1:** This bit will be cleared by writing 1 to it. **Note 2:** If reset slave's transmission circuit when slave selection signal is active, this flag will be set to 1 after 2 peripheral clock cycles + 3 system clock cycles since the reset operation is done.
[18]	**TXTHIF**	**Transmit FIFO Threshold Interrupt Flag (Read Only)** 0 = The valid data count within the transmit FIFO buffer is larger than the setting value of TXTH. 1 = The valid data count within the transmit FIFO buffer is less than or equal to the setting value of TXTH.
[17]	**TXFULL**	**Transmit FIFO Buffer Full Indicator (Read Only)** 0 = Transmit FIFO buffer is not full. 1 = Transmit FIFO buffer is full. **Note:** TXFULL flag does not go high when TXPDMAEN is set to 1.
[16]	**TXEMPTY**	**Transmit FIFO Buffer Empty Indicator (Read Only)**

		0 = Transmit FIFO buffer is not empty. 1 = Transmit FIFO buffer is empty.
[15]	**SPIENSTS**	**QSPI Enable Status (Read Only)** 0 = QSPI controller Disabled. 1 = QSPI controller Enabled. **Note:** The QSPI peripheral clock is asynchronous with the system clock. In order to make sure the QSPI control logic is disabled, this bit indicates the real status of QSPI controller.
[14:13]	**Reserved**	Reserved.
[12]	**RXTOIF**	**Receive Time-out Interrupt Flag** 0 = No receive FIFO time-out event. 1 = Receive FIFO buffer is not empty and no read operation on receive FIFO buffer over 64 QSPI peripheral clock periods in Master mode or over 576 QSPI peripheral clock periods in Slave mode. When the received FIFO buffer is read by software, the time-out status will be cleared automatically. **Note:** This bit will be cleared by writing 1 to it.
[11]	**RXOVIF**	**Receive FIFO Overrun Interrupt Flag** When the receive FIFO buffer is full, the follow-up data will be dropped and this bit will be set to 1. 0 = No FIFO is overrun. 1 = Receive FIFO is overrun. **Note:** This bit will be cleared by writing 1 to it.
[10]	**RXTHIF**	**Receive FIFO Threshold Interrupt Flag (Read Only)** 0 = The valid data count within the receive FIFO buffer is smaller than or equal to the setting value of RXTH. 1 = The valid data count within the receive FIFO buffer is larger than the setting value of RXTH.
[9]	**RXFULL**	**Receive FIFO Buffer Full Indicator (Read Only)** 0 = Receive FIFO buffer is not full. 1 = Receive FIFO buffer is full.
[8]	**RXEMPTY**	**Receive FIFO Buffer Empty Indicator (Read Only)** 0 = Receive FIFO buffer is not empty. 1 = Receive FIFO buffer is empty.
[7]	**SLVURIF**	**Slave Mode TX Under Run Interrupt Flag** In Slave mode, if TX underflow event occurs and the slave select line goes to inactive state, this interrupt flag will be set to 1. 0 = No Slave TX under run event. 1 = Slave TX under run event occurred. **Note:** This bit will be cleared by writing 1 to it.
[6]	**SLVBEIF**	**Slave Mode Bit Count Error Interrupt Flag** In Slave mode, when the slave select line goes to inactive state, if bit counter is mismatch with DWIDTH, this interrupt flag will be set to 1. 0 = No Slave mode bit count error event. 1 = Slave mode bit count error event occurred. **Note:** If the slave select active but there is no any bus clock input, the SLVBEIF also active when the slave select goes to inactive state. This bit will be cleared by writing 1 to it.
[5]	**SLVTOIF**	**Slave Time-out Interrupt Flag** When the slave select is active and the value of SLVTOCNT is not 0, if the bus clock is detected, the slave time-out counter in QSPI controller logic will be started. When the value of time-out counter is greater than or equal to the value of SLVTOCNT (QSPIx_SSCTL[31:16]) before one transaction is done, the slave time-out interrupt event will be asserted. 0 = Slave time-out is not active.

		1 = Slave time-out is active. **Note:** This bit will be cleared by writing 1 to it.
[4]	**SSLINE**	**Slave Select Line Bus Status (Read Only)** 0 = The slave select line status is 0. 1 = The slave select line status is 1. **Note:** This bit is only available in Slave mode. If SSACTPOL (QSPIx_SSCTL[2]) is set 0, and the SSLINE is 1, the QSPI slave select is in inactive status.
[3]	**SSINAIF**	**Slave Select Inactive Interrupt Flag** 0 = Slave select inactive interrupt was cleared or not occurred. 1 = Slave select inactive interrupt event occurred. **Note:** Only available in Slave mode. This bit will be cleared by writing 1 to it.
[2]	**SSACTIF**	**Slave Select Active Interrupt Flag** 0 = Slave select active interrupt was cleared or not occurred. 1 = Slave select active interrupt event occurred. **Note:** Only available in Slave mode. This bit will be cleared by writing 1 to it.
[1]	**UNITIF**	**Unit Transfer Interrupt Flag** 0 = No transaction has been finished since this bit was cleared to 0. 1 = QSPI controller has finished one unit transfer. **Note:** This bit will be cleared by writing 1 to it.
[0]	**BUSY**	**Busy Status (Read Only)** 0 = QSPI controller is in idle state. 1 = QSPI controller is in busy state. The following lists the bus busy conditions: • SPIEN (QSPIx_CTL[0]) = 1 and TXEMPTY = 0. • For QSPI Master mode, SPIEN (QSPIx_CTL[0]) = 1 and TXEMPTY = 1 but the current transaction is not finished yet. • For QSPI Master mode, SPIEN (QSPIx_CTL[0]) = 1 and RXONLY = 1. • For QSPI Slave mode, SPIEN (QSPIx_CTL[0]) = 1 and there is serial clock input into the QSPI core logic when slave select is active. • For QSPI Slave mode, SPIEN (QSPIx_CTL[0]) = 1 and the transmit buffer or transmit shift register is not empty even if the slave select is inactive. **Note:** By applications, this SPI bus flag should be used with other status registers in SPIx_STATUS such as TXCNT, RXCNT, TXTHIF, TXFULL, TXEMPTY, RXTHIF, RXFULL, RXEMPTY, and UNITIF. Therefore, the SPI transfer done events of TX/RX operations can be obtained at the correct timing point.

QSPI Status2 Register (QSPIx_STATUS2)

Register	Offset	R/W	Description	Reset Value
QSPIx_STATUS2	QSPIx_BA+0x18	R	QSPI Status2 Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		SLVBENUM					
23	22	21	20	19	18	17	16
Reserved		TXCNT					
15	14	13	12	11	10	9	8
Reserved		RXCNT					
7	6	5	4	3	2	1	0
Reserved							

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29:24]	**SLVBENUM**	**Effective Bit Number of Uncompleted RX Data** In slave mode, this status register indicates that effective bit number of uncompleted RX data when SLVBERX (QSPIx_FIFOCTL[10]) is enabled and RX bit count error event happen in SPI slave mode. This status register will be fixed to 0x0 when SLVBERX (QSPIx_FIFOCTL[10]) is disabled. **Note:** This register will be cleared to 0x0 when user writes 0x1 to SLVBEIF (QSPIx_STATUS[6]).
[23:22]	**Reserved**	Reserved.
[21:16]	**TXCNT**	**Transmit FIFO Data Count (Read Only)** This bit field indicates the valid data count of transmit FIFO buffer.
[15:14]	**Reserved**	Reserved.
[13:8]	**RXCNT**	**Receive FIFO Data Count (Read Only)** This bit field indicates the valid data count of receive FIFO buffer.
[7:0]	**Reserved**	Reserved.

QSPI FIFO2 Control Register (QSPIx_FIFOCTL2)

Register	Offset	R/W	Description	Reset Value
QSPIx_FIFOCTL2	QSPIx_BA+0x1C	R/W	QSPI FIFO2 Control Register	0x0404_0000

31	30	29	28	27	26	25	24
Reserved			TXTHPDMA				
23	22	21	20	19	18	17	16
Reserved			RXTHPDMA				
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							

Bits	Description	
[31:29]	**Reserved**	Reserved.
[28:24]	**TXTHPDMA**	**Transmit FIFO Threshold When TX PDMA Enable Only** If the valid data count of the transmit FIFO buffer is less than or equal to the TXTH setting, the TXTHIF bit will be set to 1, else the TXTHIF bit will be cleared to 0.
[23:21]	**Reserved**	Reserved.
[20:16]	**RXTHPDMA**	**Receive FIFO Threshold When RX PDMA Enable Only** If the valid data count of the receive FIFO buffer is larger than the RXTH setting, the RXTHIF bit will be set to 1, else the RXTHIF bit will be cleared to 0.
[15:0]	**Reserved**	Reserved.

QSPI Data Transmit Register (QSPIx_TX)

Register	Offset	R/W	Description	Reset Value
QSPIx_TX	QSPIx_BA+0x20	W	QSPI Data Transmit Register	0x0000_0000

31	30	29	28	27	26	25	24
TX							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TX							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TX							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TX							

Bits	Description	
[31:0]	**TX**	**Data Transmit Register** The data transmit registers pass through the transmitted data into the 8-level transmit FIFO buffers. The number of valid bits depends on the setting of DWIDTH (QSPIx_CTL[12:8]). If DWIDTH is set to 0x08, the bits TX[7:0] will be transmitted. If DWIDTH is set to 0x00, the QSPI controller will perform a 32-bit transfer. **Note:** In Master mode, the QSPI controller will start to transfer the QSPI bus clock after 1 APB clock and 6 peripheral clock cycles after user writes to this register.

QSPI Data Receive Register (QSPIx_RX)

Register	Offset	R/W	Description	Reset Value
QSPIx_RX	QSPIx_BA+0x30	R	QSPI Data Receive Register	0x0000_0000

31	30	29	28	27	26	25	24
RX							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RX							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RX							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RX							

Bits	Description	
[31:0]	**RX**	**Data Receive Register (Read Only)** There are 8-level FIFO buffers in this controller. The data receive register holds the data received from QSPI data input pin. If the RXEMPTY (QSPIx_STATUS[8) is not set to 1, the receive FIFO buffers can be accessed through software by reading this register.

QSPI Internal Control Register (QSPIx_INTERNAL)

Register	Offset	R/W	Description	Reset Value
QSPIx_INTERNAL	QSPIx_BA+0x48	R/W	QSPI Internal Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
MRXPHASE				Reserved			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:12]	**MRXPHASE**	**QSPI Master Receive Phase Selection** It is used to adjust the timing of receive data sampled into RX skew buffer while QSPI Master mode due to the PAD I/O latency (send QSPI_CLK to Slave device through the output pin and then wait the receive data from Slave device through the input pin) ● For DTR mode (QSPIx_CTL[23]) disable : 0x0: Sample RX data by the Master feedback QSPI_CLK (Chip PAD-out loopback to PAD-in) 0x1: Sample RX data latter the 0.5 period of sending QSPI_CLK 0x2: Sample RX data latter the 1 period of sending QSPI_CLK 0x3: Sample RX data latter the 1.5 period of sending QSPI_CLK 0x4: Sample RX data latter the 2 periods of sending QSPI_CLK … 0xb: Sample RX data latter the 5.5 periods of sending QSPI_CLK Others are reserved. ● For DTR mode (QSPIx_CTL[23]) enable : 0x0: Sample RX data latter the 0 period of sending QSPI_CLK 0x1: Sample RX data latter the 0.25 period of sending QSPI_CLK 0x2: Sample RX data latter the 0.5 period of sending QSPI_CLK 0x3: Sample RX data latter the 0.75 period of sending QSPI_CLK 0x4: Sample RX data latter the 1 periods of sending QSPI_CLK … 0xb: Sample RX data latter the 2.75 periods of sending QSPI_CLK Others are reserved.
[11:0]	**Reserved**	Reserved.

6.29 I²S Controller (I²S)

6.29.1 Overview

The I²S controller consists of I²S protocol to interface with external audio CODEC. Two 16-level depth FIFO for reading path and writing path respectively are capable of handling 8/16/24/32 bits audio data sizes. A PDMA controller handles the data movement between FIFO and memory.

6.29.2 Features

- Supports Master mode and Slave mode
- Capable of handling 8, 16, 24 and 32 bits data sizes in each audio channel
- Supports monaural and stereo audio data
- Supports I²S protocols: Philips standard, MSB-justified, and LSB-justified data format
- Supports PCM protocols: PCM standard, MSB-justified, and LSB-justified data format
- PCM protocol supports TDM multi-channel transmission in one audio sample, and the number of data channel can be set as 2, 4, 6, or 8
- Provides two 16-level FIFO data buffers, one for transmitting and the other for receiving
- Generates interrupt requests when buffer levels cross a programmable boundary
- Supports two PDMA requests, one for transmitting and the other for receiving

6.29.3 Block Diagram

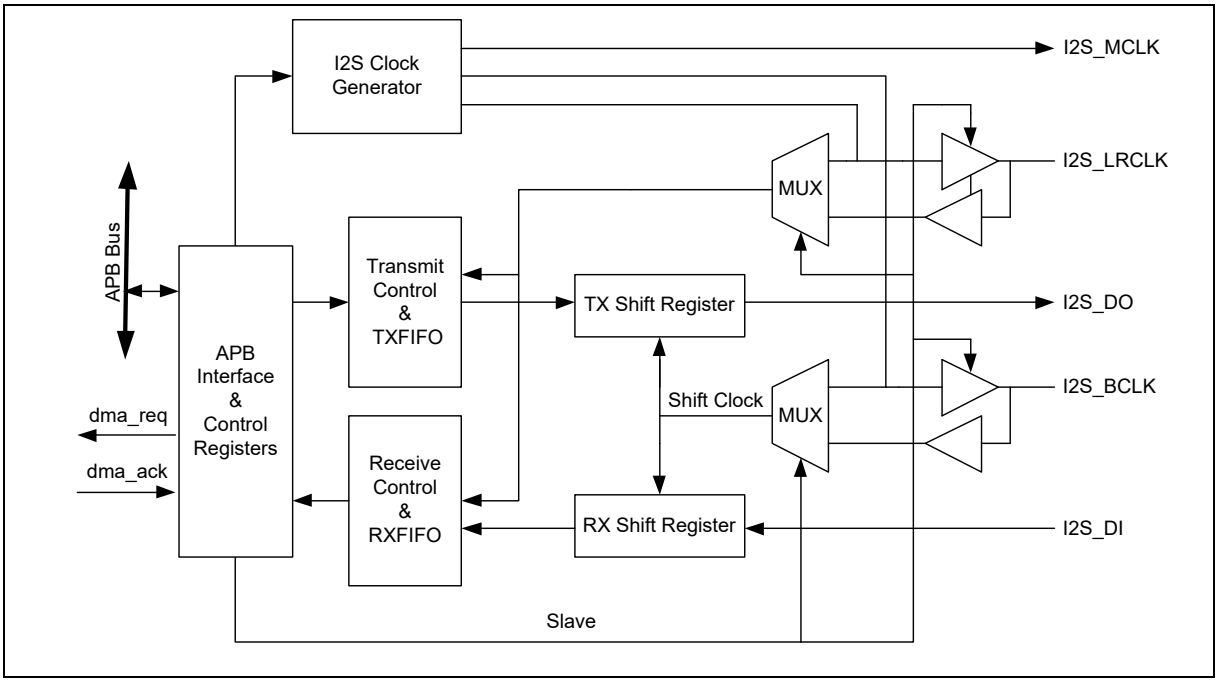


Figure 6.29-1 I²S Controller Block Diagram

6.29.4 Basic Configuration

6.29.4.1 I2S0 Basic Configuration

- Clock Source Configuration
  - Select the source of I2S0 peripheral clock on I2S0SEL (CLK_CLKSEL4[13:12]).

- Enable I2S0 peripheral clock in I2S0CKEN (CLK_APBCLK2[0]).

- Reset Configuration
  - Reset I2S0 controller in I2S0RST (SYS_IPRST1[29]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
I2S0	I2S0_BCLK	PG.13	MFP5
		PL.1	MFP7
		PJ.1	MFP7
		PK.13	MFP5
		PF.5	MFP4
		PF.11	MFP5
	I2S0_DI	PG.14	MFP5
		PL.2	MFP7
		PJ.2	MFP7
		PK.14	MFP5
		PF.6	MFP4
		PF.12	MFP5
	I2S0_DO	PG.15	MFP5
		PL.3	MFP7
		PJ.3	MFP7
		PK.15	MFP5
		PF.7	MFP4
		PF.13	MFP5
	I2S0_LRCK	PG.12	MFP5
		PL.0	MFP7
		PJ.0	MFP7
		PK.12	MFP5
		PF.4	MFP4
		PF.10	MFP5
	I2S0_MCLK	PG.11	MFP5
		PD.6	MFP7
		PL.4	MFP7
		PN.15	MFP5
		PF.14	MFP15
		PB.9	MFP6

6.29.4.2 I2S1 Basic Configuration

- Clock Source Configuration
  - Select the source of I2S1 peripheral clock on I2S0SEL (CLK_CLKSEL4[15:14]).
  - Enable I2S1 peripheral clock in I2S1CKEN (CLK_APBCLK2[1]).
- Reset Configuration
  - Reset I2S1 controller in I2S1RST (SYS_IPRST3[17]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
I2S1	I2S1_BCLK	PL.13	MFP6
		PD.13	MFP12
		PG.6	MFP9
		PI.5	MFP6
		PL.9	MFP7
		PD.9	MFP7
		PK.6	MFP7
		PB.13	MFP6
	I2S1_DI	PB.14	MFP6
		PL.14	MFP6
		PD.14	MFP12
		PG.5	MFP9
		PI.6	MFP6
		PL.10	MFP7
		PD.10	MFP7
		PK.5	MFP7
	I2S1_DO	PB.15	MFP6
		PL.15	MFP6
		PD.15	MFP12
		PG.4	MFP9
		PI.7	MFP6
		PL.11	MFP7
		PD.11	MFP7
	I2S1_LRCK	PL.12	MFP6
		PD.12	MFP12
		PG.7	MFP9
		PI.4	MFP6
		PL.8	MFP7

		PD.8	MFP7
		PK.7	MFP7
		PB.12	MFP6
	I2S1_MCLK	PG.0	MFP9
		PG.3	MFP12
		PM.12	MFP9
		PK.0	MFP6
		PD.7	MFP7
		PL.5	MFP7
		PB.9	MFP12
		PB.11	MFP6

6.29.5 Functional Description

6.29.5.1 I²S Clock

The I²S controller has four clock sources selected by I2S0SEL (CLK_CLKSEL4[15:14]) and I2S1SEL (CLK_CLKSEL4[13:12]). The I²S clock rate must be slower than or equal to system clock rate.

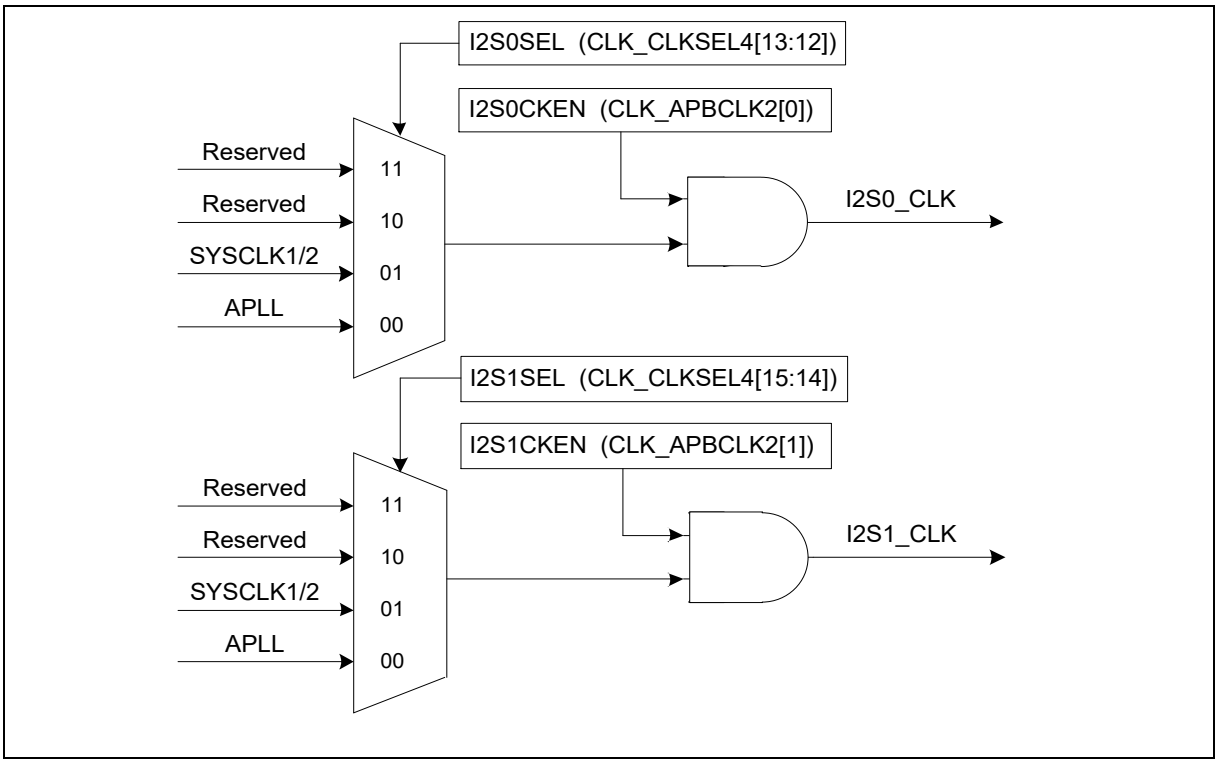


Figure 6.29-2 I²S Clock Control Diagram

6.29.5.2 Master/Slave Interface

The I²S function can operate as master or slave mode by setting SLAVE (I2S_CTL0[8]) to communicate with other I²S slave or master. The serial bus clock I2S_BCLK is permanently generated by the master

even though there is no transferring data bit at the moment. The word select signal I2S_LRCLK is also generated by the master and it indicates the beginning of a new data word and the targeted audio channel. Both the I2S_LRCLK and the transmitting data change synchronously to the falling edges of I2S_BCLK.

In some applications, especially for Audio-ADC or Audio-DAC, a master clock signal, I2S_MCLK, is required with a fixed phase relation to the I2S_BCLK. The I2S_MCLK is enabled by MCLKEN (I2S_CTL0[15]). In Master mode, the I2S_MCLK, I2S_BCLK, I2S_LRCLK is output to device slave. And if in slave mode, the I2S_MCLK is output to master, and I2S_BCLK or I2S_LRCLK is input from master.

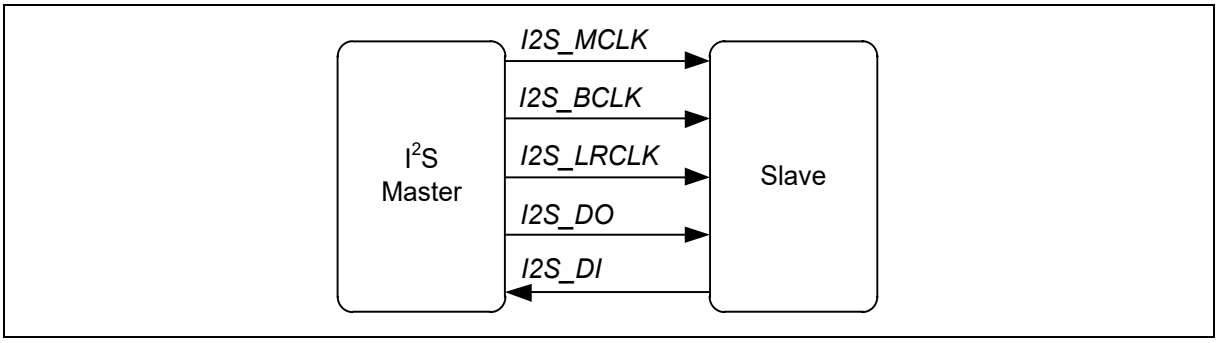


Figure 6.29-3 Master Mode Interface Block Diagram

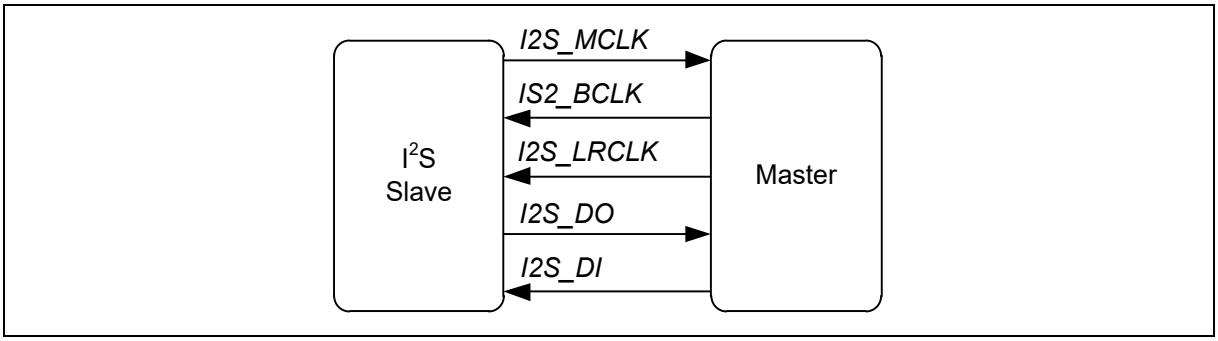


Figure 6.29-4 Slave Mode Interface Block Diagram

6.29.5.3 I²S Operation

The I²S controller supports MSB-justified, LSB-justified, and I²S Philips standard data format. The I2S_LRCLK signal indicates which audio channel is in transferring. The bit count of an audio channel is defined by CHWIDTH (I2S_CTL0[29:28]), and the bit-width of data word in an audio channel is determined by DATWIDTH (I2S_CTL0[5:4]). If CHWIDTH (I2S_CTL0[29:28]) is less than DATWIDTH (I2S_CTL0[5:4]), the hardware will set the channel bit-width to be same as data bit-width. However, there will be redundant zero bits in each audio channel if CHWIDTH (I2S_CTL0[29:28]) is greater than DATWIDTH (I2S_CTL0[5:4]).

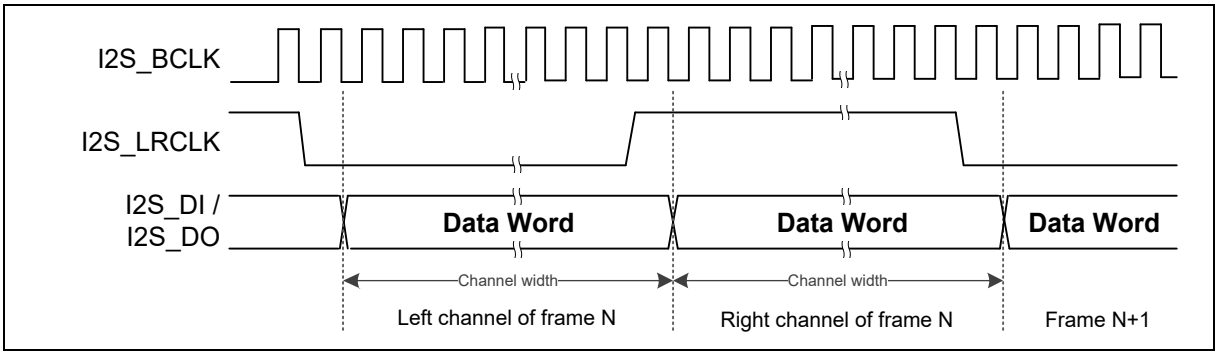


Figure 6.29-5 I²S Channel Width and Data Width (CHWIDTH≦DATWIDTH)

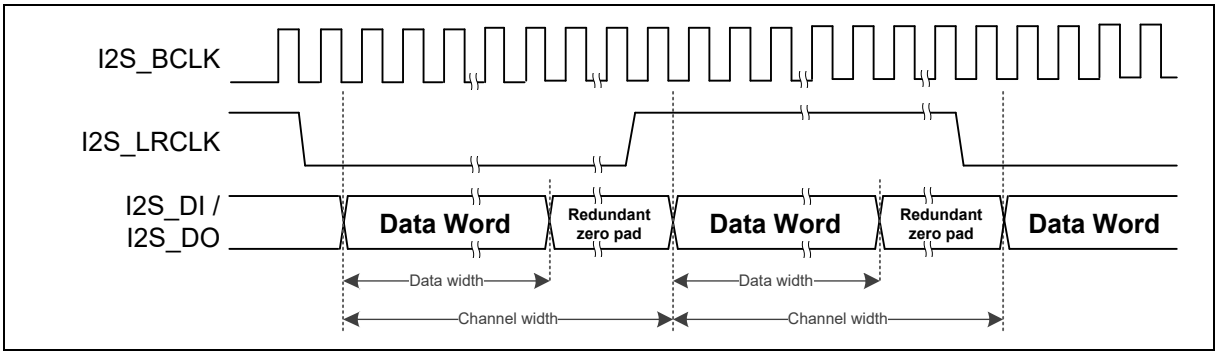


Figure 6.29-6 I²S Channel Width and Data Width (CHWIDTH > DATWIDTH)

The transferring data sequence is always started from the MSB (most significant bit) to the LSB (least significant bit). As shown in Figure 6.29-7, transmitting data are read at rising edge of I2S_BCLK and sent out at falling edge of I2S_BCLK in I²S protocol. In I²S data format, the MSB is sent and latched at the next falling edge of I2S_BCLK cycle after the transition of I2S_LRCLK. In MSB justified data format, the I2S_LRCLK changes the polarity at the transmitting of the first data bit (MSB) in each audio channel. In LSB justified data format, the LSB is sent and latched at the last I2S_BCLK cycle of an audio channel. The MSB justified and LSB justified data format of I²S protocol can be selected by FORMAT (I2S_CTL0[26:24]).

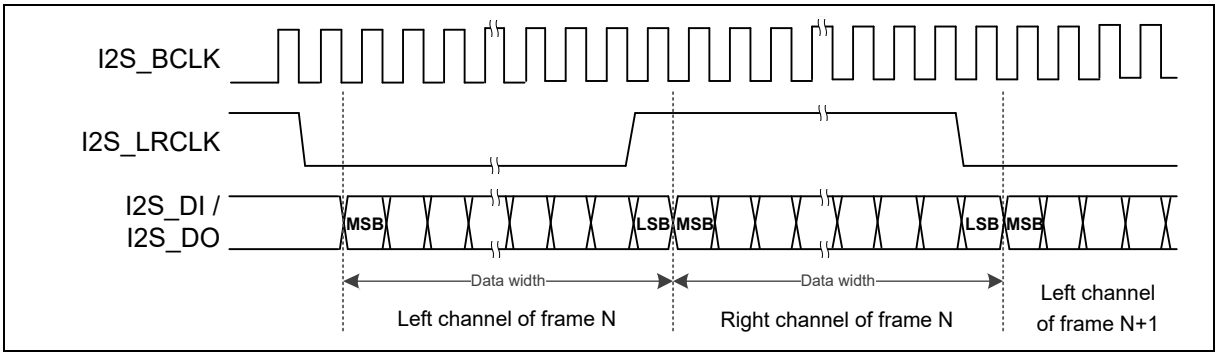


Figure 6.29-7 I²S Data Format Timing Diagram (FORMAT = 0x0; CHWIDTH≦DATWIDTH)

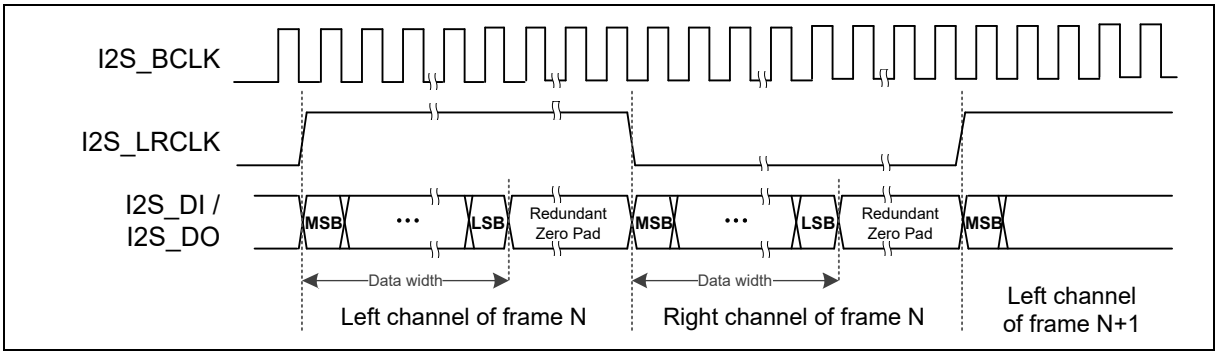


Figure 6.29-8 MSB Justified Data Format (FORMAT = 0x1; CHWIDTH > DATWIDTH)

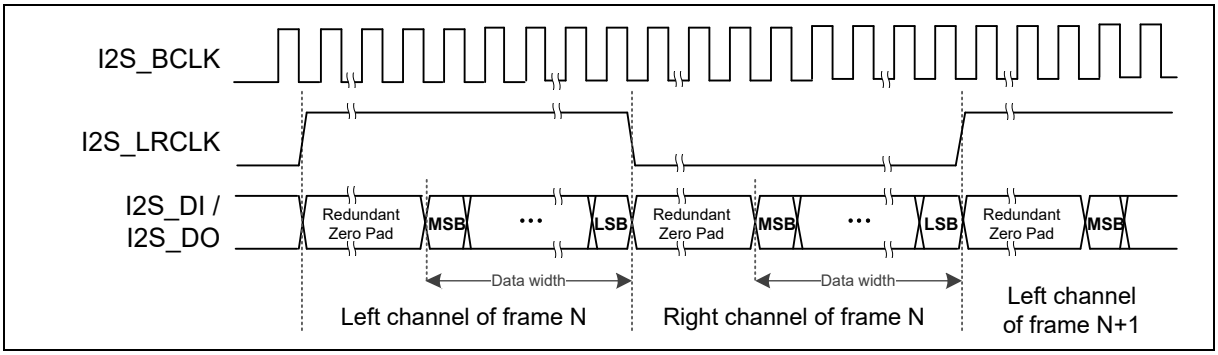


Figure 6.29-9 LSB Justified Data Format (FORMAT = 0x2; CHWIDTH > DATWIDTH)

The I²S controller also supports PCM audio transmission which can be selected by FORMAT (I2S_CTL0[26:24]). In PCM protocol, the function of I2S_LRCLK is simply to identify the beginning of an audio sample (or audio frame) and it is always indicated by the rising edge of the pulse. Therefore, the I2S_LRCLK in PCM protocol may be also called “frame start” or “frame sync” signal. In master device, there are two common representations for the width of the frame start pulse which can use PCMSYNC (I2S_CTL0[27]) to choose: One is equivalent to the period of a channel width and the other is equivalent to a single period of the I2S_BCLK.

Same as I²S protocol, the DATWIDTH (I2S_CTL0[5:4]) and CHWIDTH (I2S_CTL0[29:28]) can be used to configure the data bit-width and channel bit-width in PCM protocol. Besides, FORMAT (I2S_CTL0[26:24]) can also be used to select the different data formats of PCM standard mode, PCM with MSB justified, and PCM with LSB justified data format.

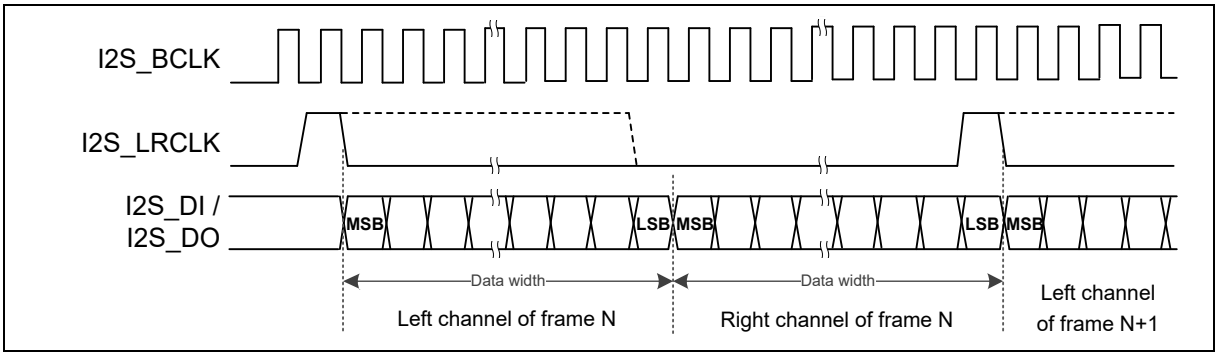


Figure 6.29-10 Standard PCM Audio Timing Diagram (FORMAT = 0x4; CHWIDTH≦DATWIDTH)

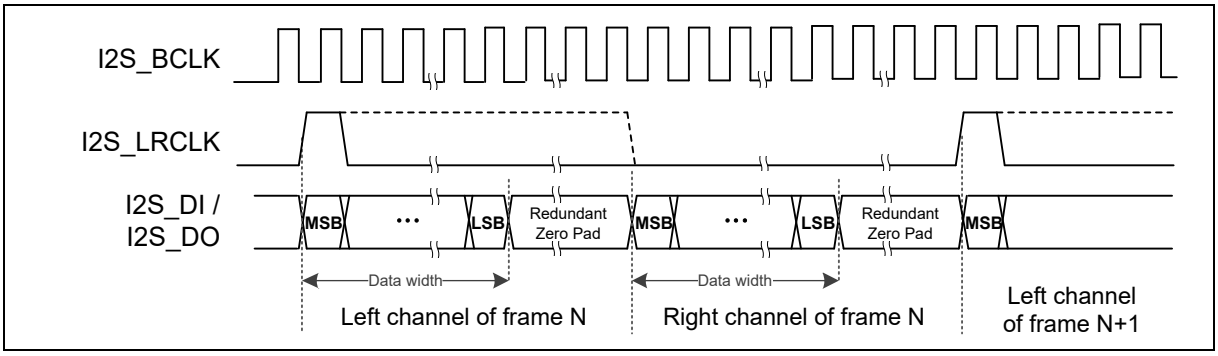


Figure 6.29-11 PCM with MSB Justified Data Format (FORMAT = 0x5; CHWIDTH > DATWIDTH)

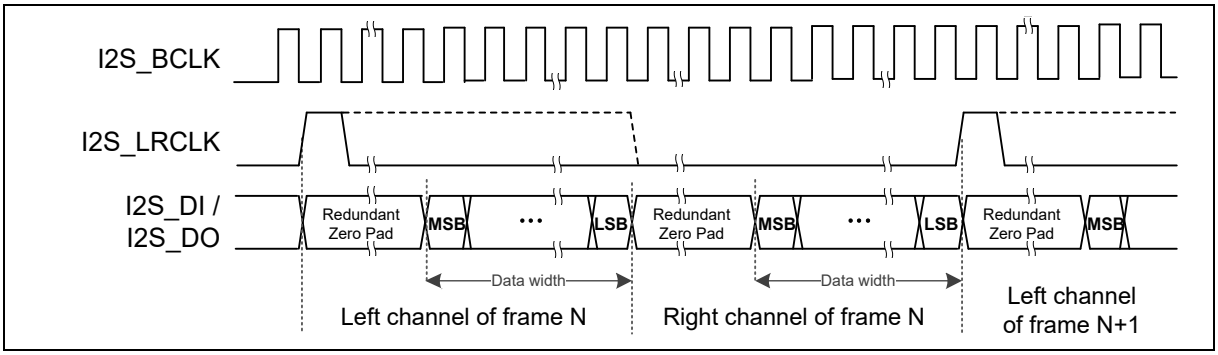


Figure 6.29-12 PCM with LSB Justified Data Format (FORMAT = 0x6; CHWIDTH > DATWIDTH)

6.29.5.4 TDM Multi-channel Transmission

The PCM mode in this I²S controller also supports TDM transmission. The Time Division Multiplexed (TDM) method allows multiple channels of audio data to be transmitted on a single data line. The TDM interface is similar to the 2-channel PCM audio interface with the exception that more audio channels are transmitted within a sample frame which is defined by a period of the I2S_LRCLK. The channel number of TDM interface is typically 4, 6, or 8 and it is selected by TDMCHNUM (I2S_CTL0[31:30]).

Same as previous I²S and PCM descriptions, each channel block is comprised of the audio data word followed by a sufficient number of zero data bits to complete one channel block. The bit-width of data word and channel block are defined by DATWIDTH (I2S_CTL0[5:4]) and CHWIDTH (I2S_CTL0[29:28]) respectively. Note that the TDM PCM mode supports 16-bit, 24-bit, 32-bit audio data word (excluding 8-bit data), and the hardware will set the bit-width of transmitting data as 16-bit if DATWIDTH (I2S_CTL0[5:4]) is 0x0. The pulse width of frame start signal is also selected by PCMSYNC (I2S_CTL0[27]).

The examples of 6-channel TDM transmission with 24-bit audio data in 32-bit channel block are shown in Figure 6.29-13. In 2-channel audio interface, the first and second audio channels are called as left-channel and right-channel (or channel0 and channel1). In TDM multi-channel application, the first and second audio channels are called as channel0 and channel1.

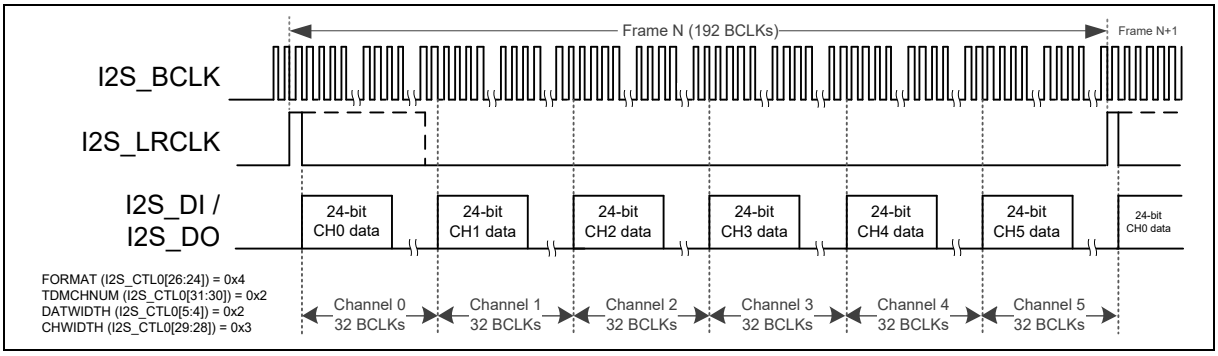


Figure 6.29-13 TDM 6-channel Audio Format with 24-bit Data in 32-bit Channel Block (PCM Standard Data Format; FORMAT=0x4)

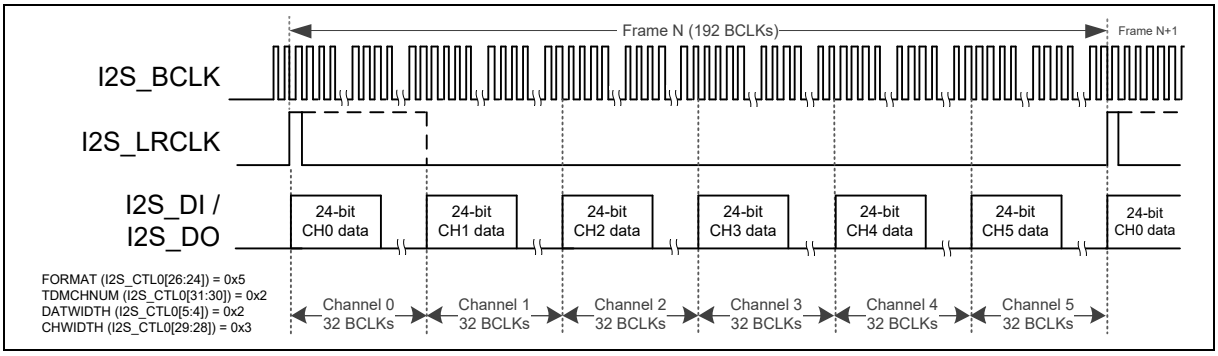


Figure 6.29-14 TDM 6-channel Audio Format with 24-bit Data in 32-bit Channel Block (PCM with MSB Justified; FORMAT=0x5)

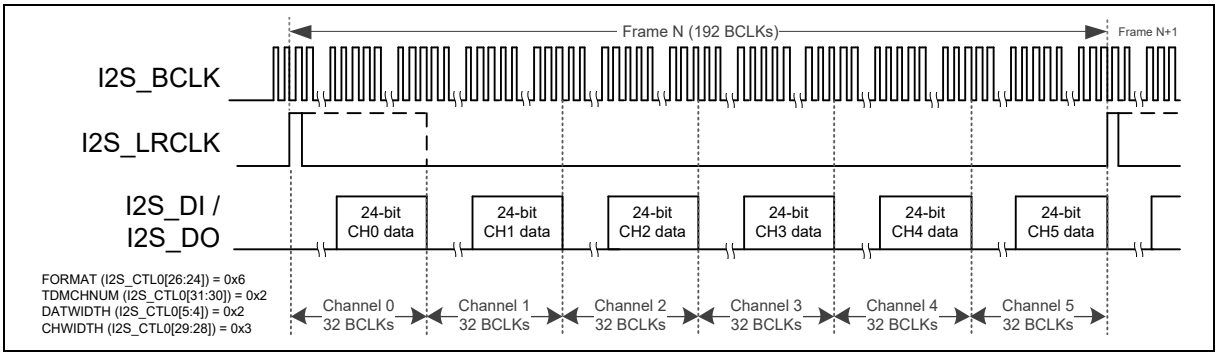


Figure 6.29-15 TDM 6-channel Audio Format with 24-bit Data in 32-bit Channel Block (PCM with LSB Justified; FORMAT=0x6)

6.29.5.5 Zero Crossing

When playing the audio by I^{2}S controller, the output transmitting data comes from the memory by PDMA or by CPU. However, there may be some pop noise which induces the uncomfortable hearing if the playing sound volume is changed greatly by user. The zero-crossing event of audio data means the playing sound is relatively silent at the moment. Therefore, the zero-cross interrupt can be used for the indication of gain level adjustment in order to prevent the huge variance of sound volume.

If zero-cross detection of individual audio channel is enabled by the corresponding control bit from CH0ZCEN to CH7ZCEN (I2S_CTL1[0] to I2S_CTL1[7]), the hardware will detect the next transferring data word of the corresponding audio channel whether it is 0 or its MSB has been changed. If zero value or MSB (sign bit) changing of the transmitting audio data has been detected while zero-cross detection is enabled, the hardware will set the corresponding status bit from CH0ZCIF to CH7ZCIF

(I2S_STATUS1[0] to I2S_STATUS1[7]) for the audio channel and then keep the output audio data silent (all data bit zero) automatically until the corresponding event status bit is cleared by software.

Therefore, if user wants to modify the audio playing gain, users can enable the zero crossing interrupt function, CH0ZCIEN to CH7ZCIEN (I2S_IEN[16:23]), to indicate the zero crossing time and to change the audio gain. This will reduce the pop noise.

6.29.5.6 PDMA Mode

The I²S function can use PDMA function for transmitting or receiving data access. If the PDMA function of transmitting data is enabled by TXPDMAEN (I2S_CTL0[20]), the I²S controller will generate the request signal and then get transmitting audio data from memory by PDMA IP automatically while TX FIFO is not full. If the PDMA function of receiving data is enabled by RXPDMAEN (I2S_CTL0[21]), the I²S controller will generate the request signal and then the receiving data will be moved into memory by PDMA hardware automatically while the RX FIFO is not empty. Therefore, using PDMA function will save the CPU loading to service other functions.

6.29.5.7 I²S Interrupt Sources

The I²S controller supports zero-cross interrupt of individual audio channel, transmit FIFO threshold level interrupt, transmit FIFO overflow interrupt and transmit FIFO underflow interrupt in transmit operation. In receive operation, it supports receive FIFO threshold level interrupt, receive FIFO overflow interrupt and receive FIFO underflow interrupt. When I²S interrupt occurs, user can check I2STXINT (I2S_STATUS0[2]) and I2SRXINT (I2S_STATUS0[1]) flags to recognize the interrupt sources.

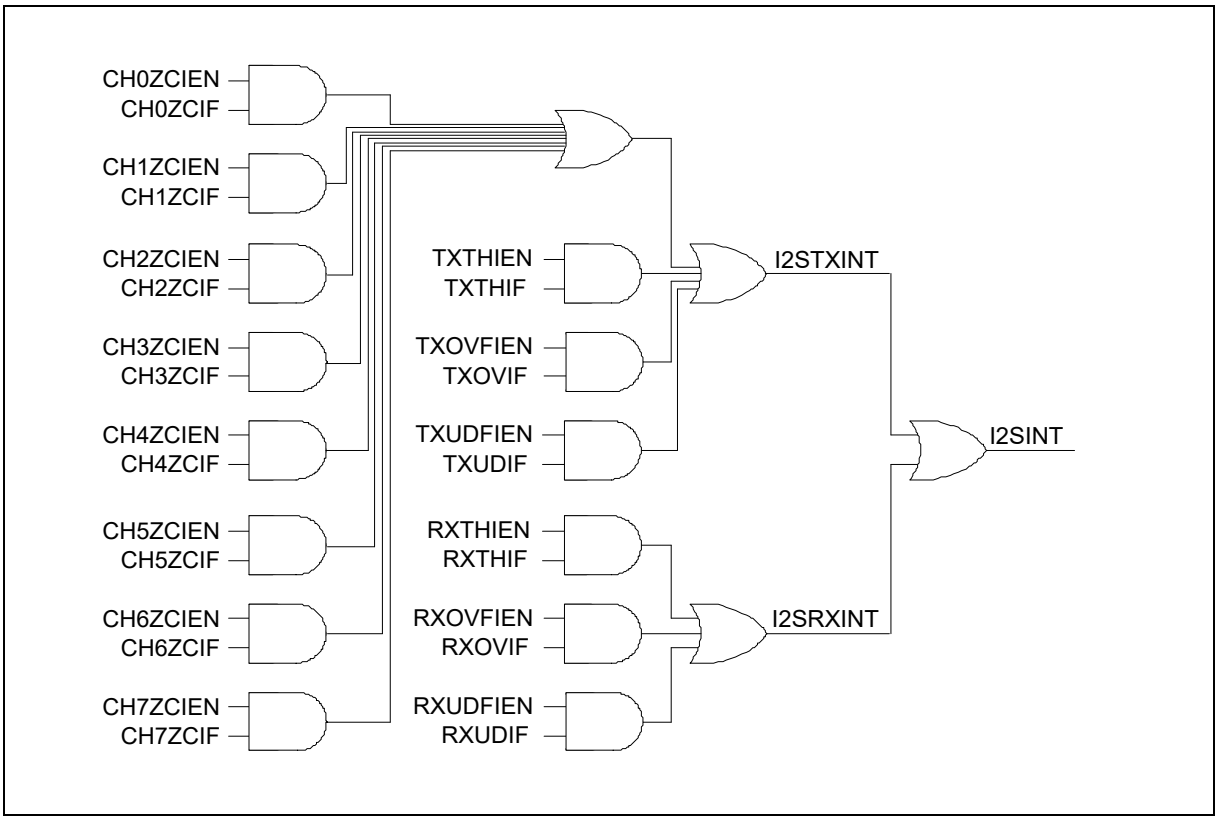


Figure 6.29-16 I²S Interrupts

6.29.5.8 FIFO Operation

In 2-channel I²S or PCM protocol, the bit-width of audio data in a channel block can be 8, 16, 24, or 32 bits. The memory arrangements of audio data for various settings are shown in Figure 6.29-17.

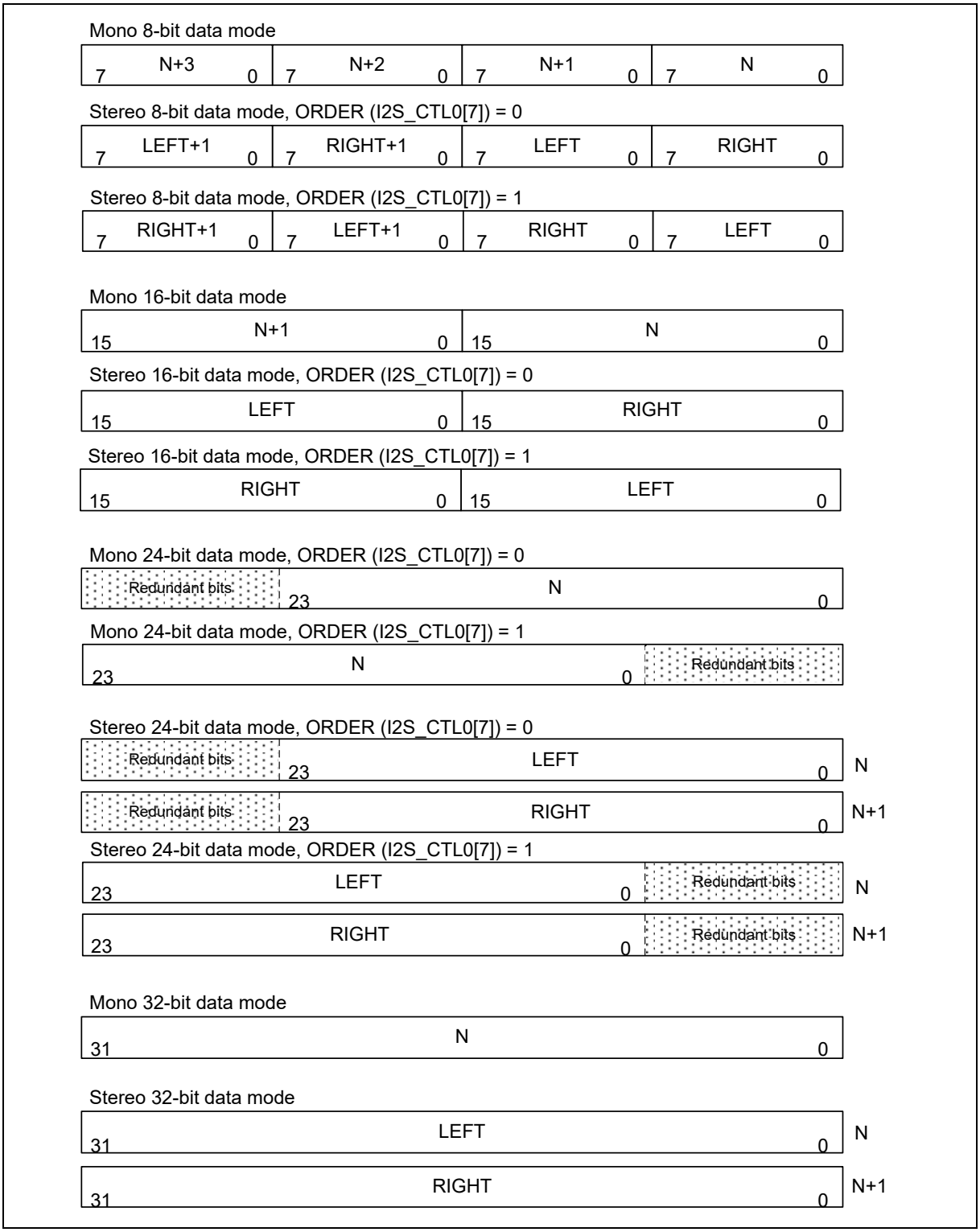


Figure 6.29-17 FIFO Contents for Various 2-channel Audio Modes

In 4-channel TDM PCM data format, the bit-width of audio data in a channel block can be 16, 24, or 32 bits. The memory arrangements of audio data for various settings are shown in Figure 6.29-18.

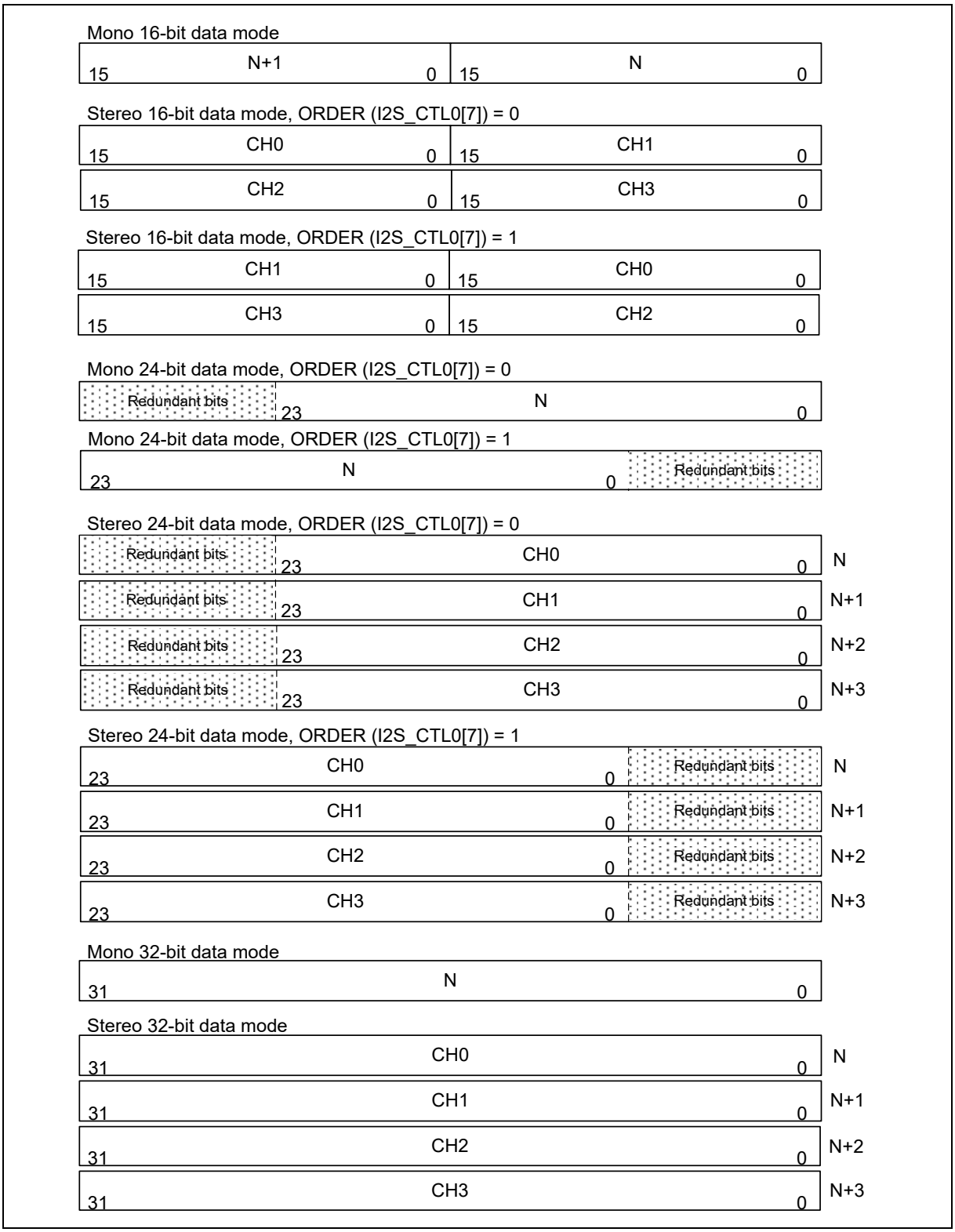


Figure 6.29-18 FIFO Contents for Various 4-channel Audio Modes

In 6-channel TDM PCM data format, the bit-width of audio data in a channel block can be 16, 24, or 32 bits. The memory arrangements of audio data for various settings are shown in Figure 6.29-19. In 16-

bit audio data transmission, ORDER (I2S_CTL0[7]) can be used to swap the audio data of even and odd channels which are stored in transmitting and receiving FIFO. In 24-bit audio data transmission, ORDER (I2S_CTL0[7]) can be also used to select the left-alignment or right-alignment formula of audio data which is stored in 32-bit FIFO entries.

The FIFO content of 8-channel TDM PCM data format is similar to 6-channel and it can be analogized easily.

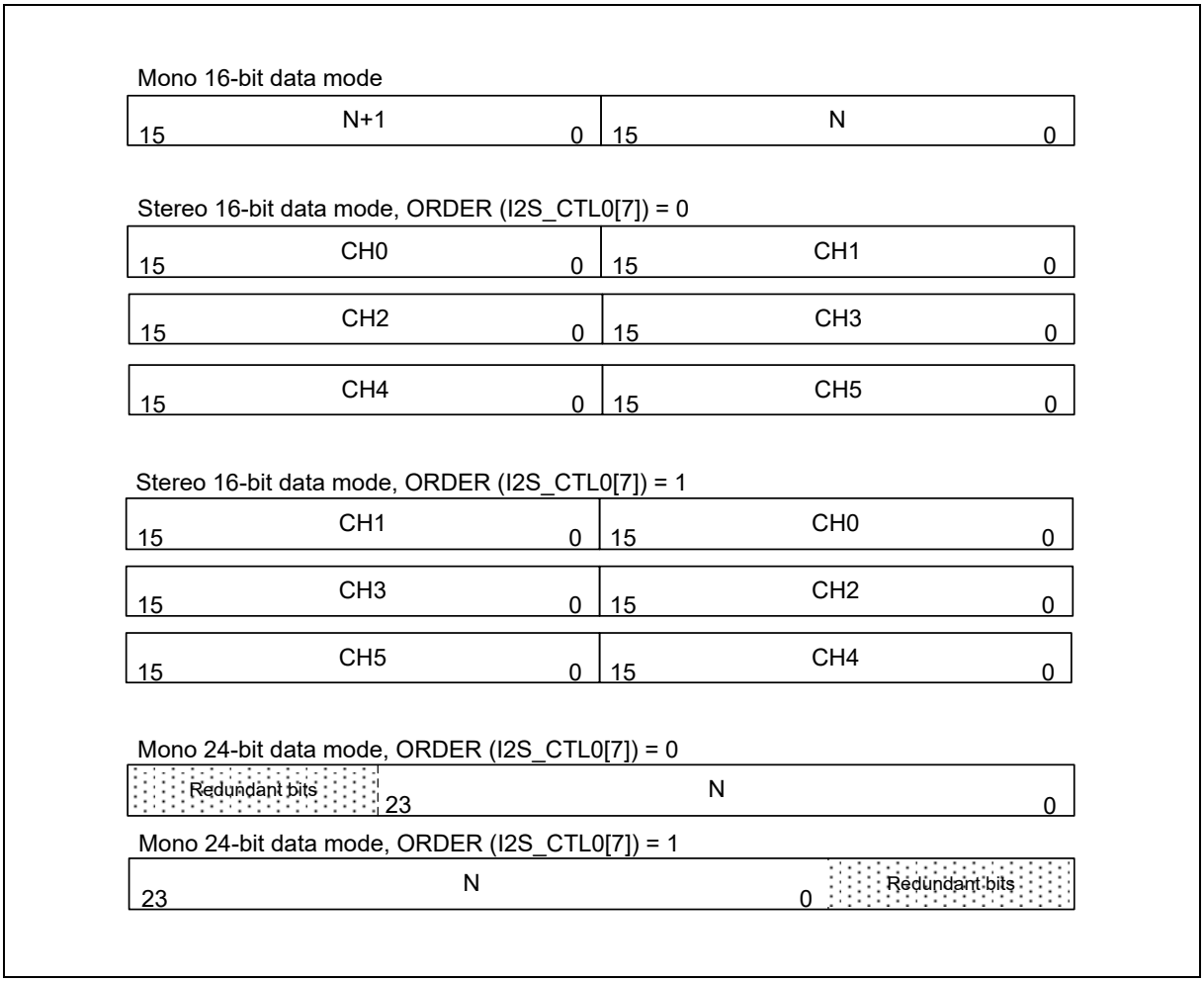


Figure 6.29-19 FIFO Contents for Various 6-channel Audio Modes (Part-1)

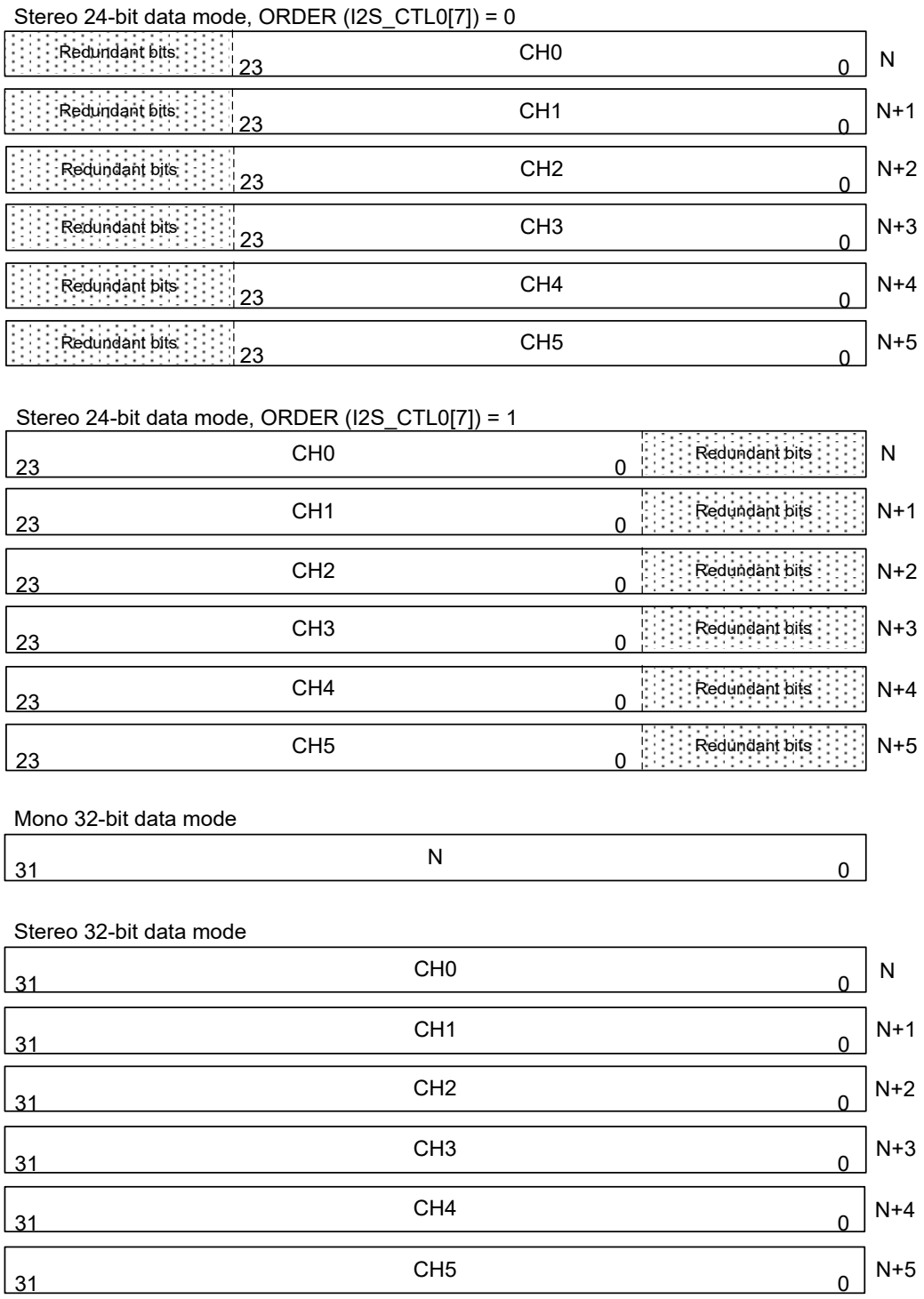


Figure 6.29-20 FIFO Contents for Various 6-channel Audio Modes (Part-2)

6.29.6 Register Map

R: Read only, W: Write only, R/W: Both read and write

Register	Offset	R/W	Description	Reset Value
I²S Base Address **I2Sx_BA = 0x4048_0000 + (0x10000 * x)** **x=0,1**				
I2S_CTL0	I2Sx_BA+0x00	R/W	I²S Control Register 0	0x0000_0000
I2S_CTL1	I2Sx_BA+0x20	R/W	I²S Control Register 1	0x0000_0000
I2S_CLKDIV	I2Sx_BA+0x04	R/W	I²S Clock Divider Register	0x0000_0000
I2S_IEN	I2Sx_BA+0x08	R/W	I²S Interrupt Enable Register	0x0000_0000
I2S_STATUS0	I2Sx_BA+0x0C	R/W	I²S Status Register 0	0x0014_1038
I2S_STATUS1	I2Sx_BA+0x24	R/W	I²S Status Register 1	0x0000_0000
I2S_TXFIFO	I2Sx_BA+0x10	W	I²S Transmit FIFO Register	0x0000_0000
I2S_RXFIFO	I2Sx_BA+0x14	R	I²S Receive FIFO Register	0x0000_0000

6.29.7 Register Description

I²S Control Register 0 (I2S_CTL0)

Register	Offset	R/W	Description	Reset Value
I2S_CTL0	I2Sx_BA+0x00	R/W	I²S Control Register 0	0x0000_0000

31	30	29	28	27	26	25	24
TDMCHNUM		CHWIDTH		PCMSYNC	FORMAT		
23	22	21	20	19	18	17	16
RXLCH	Reserved	RXPDMAEN	TXPDMAEN	RXFBCLR	TXFBCLR	Reserved	
15	14	13	12	11	10	9	8
MCLKEN	Reserved						SLAVE
7	6	5	4	3	2	1	0
ORDER	MONO	DATWIDTH		MUTE	RXEN	TXEN	I2SEN

Bits	Description	
[31:30]	**TDMCHNUM**	**TDM Channel Number** This bit fields are used to define the TDM channel number in one audio frame while PCM mode (FORMAT[2] = 1). 00 = 2 channels in audio frame. 01 = 4 channels in audio frame. 10 = 6 channels in audio frame. 11 = 8 channels in audio frame.
[29:28]	**CHWIDTH**	**Channel Width** This bit fields are used to define the length of audio channel. If CHWIDTH < DATWIDTH, the hardware will set the real channel length as the bit-width of audio data which is defined by DATWIDTH. 00 = The bit-width of each audio channel is 8-bit. 01 = The bit-width of each audio channel is 16-bit. 10 = The bit-width of each audio channel is 24-bit. 11 = The bit-width of each audio channel is 32-bit.
[27]	**PCMSYNC**	**PCM Synchronization Pulse Length Selection** This bit field is used to select the high pulse length of frame synchronization signal in PCM protocol 0 = One BCLK period. 1 = One channel period. **Note:** This bit is only available in master mode

[26:24]	**FORMAT**	**Data Format Selection** 000 = I²S standard data format. 001 = I²S with MSB justified. 010 = I²S with LSB justified. 011 = Reserved. 100 = PCM standard data format. 101 = PCM with MSB justified. 110 = PCM with LSB justified. 111 = Reserved.
[23]	**RXLCH**	**Receive Left Channel Enable Bit** When monaural format is selected (MONO = 1), I²S will receive channel1 data if RXLCH is set to 0, and receive channel0 data if RXLCH is set to 1. 0 = Receive channel1 data in MONO mode. 1 = Receive channel0 data in MONO mode.
[22]	**Reserved**	Reserved.
[21]	**RXPDMAEN**	**Receive PDMA Enable Bit** 0 = Receive PDMA function Disabled. 1 = Receive PDMA function Enabled.
[20]	**TXPDMAEN**	**Transmit PDMA Enable Bit** 0 = Transmit PDMA function Disabled. 1 = Transmit PDMA function Enabled.
[19]	**RXFBCLR**	**Receive FIFO Buffer Clear** 0 = No Effect. 1 = Clear RX FIFO. **Note 1:** Write 1 to clear receive FIFO, internal pointer is reset to FIFO start point, and RXCNT (I2S_STATUS1[20:16]) returns 0 and receive FIFO becomes empty. **Note 2:** This bit is cleared by hardware automatically, read it return 0.
[18]	**TXFBCLR**	**Transmit FIFO Buffer Clear** 0 = No Effect. 1 = Clear TX FIFO. **Note 1:** Write 1 to clear transmit FIFO, internal pointer is reset to FIFO start point, and TXCNT (I2S_STATUS1[12:8]) returns 0 and transmit FIFO becomes empty but data in transmit FIFO is not changed. **Note 2:** This bit is clear by hardware automatically, read it return 0.
[17:16]	**Reserved**	Reserved.
[15]	**MCLKEN**	**Master Clock Enable Bit** If MCLKEN is set to 1, I²S controller will generate master clock on I2S_MCLK pin for external audio devices. 0 = Master clock Disabled. 1 = Master clock Enabled.
[14:9]	**Reserved**	Reserved.
[8]	**SLAVE**	**Slave Mode Enable Bit** 0 = Master mode. 1 = Slave mode. **Note:** I²S can operate as master or slave. For Master mode, I2S_BCLK and I2S_LRCLK pins are output mode and send out bit clock to Audio CODEC chip. In Slave mode, I2S_BCLK and I2S_LRCLK pins are input mode and I2S_BCLK and I2S_LRCLK signals are received from outer Audio CODEC chip.

[7]	**ORDER**	**Stereo Data Order in FIFO** In 8-bit/16-bit data width, this bit is used to select whether the even or odd channel data is stored in higher byte. In 24-bit data width, this is used to select the left/right alignment method of audio data which is stored in data memory consisted of 32-bit FIFO entries. 0 = Even channel data at high byte in 8-bit/16-bit data width. LSB of 24-bit audio data in each channel is aligned to right side in 32-bit FIFO entries. 1 = Even channel data at low byte in 8-bit/16-bit data width. MSB of 24-bit audio data in each channel is aligned to left side in 32-bit FIFO entries.
[6]	**MONO**	**Monaural Data Control** 0 = Data is stereo format. 1 = Data is monaural format. **Note:** When chip records data, RXLCH (I2S_CTL0[23]) indicates which channel data will be saved if monaural format is selected.
[5:4]	**DATWIDTH**	**Data Width** This bit field is used to define the bit-width of data word in each audio channel 00 = The bit-width of data word is 8-bit. 01 = The bit-width of data word is 16-bit. 10 = The bit-width of data word is 24-bit. 11 = The bit-width of data word is 32-bit.
[3]	**MUTE**	**Transmit Mute Enable Bit** 0 = Transmit data is shifted from buffer. 1 = Send zero on transmit channel.
[2]	**RXEN**	**Receive Enable Bit** 0 = Data receiving Disabled. 1 = Data receiving Enabled.
[1]	**TXEN**	**Transmit Enable Bit** 0 = Data transmission Disabled. 1 = Data transmission Enabled.
[0]	**I2SEN**	**I^2S Controller Enable Bit** 0 = I^2S controller Disabled. 1 = I^2S controller Enabled.

I²S Control Register 1 (I2S_CTL1)

Register	Offset	R/W	Description	Reset Value
I2S_CTL1	I2Sx_BA+0x20	R/W	I²S Control Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						PB16ORD	PBWIDTH
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				RXTH			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				TXTH			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CH7ZCEN	CH6ZCEN	CH5ZCEN	CH4ZCEN	CH3ZCEN	CH2ZCEN	CH1ZCEN	CH0ZCEN

Bits	Description	
[31:26]	**Reserved**	Reserved.
[25]	**PB16ORD**	**FIFO Read/Write Order in 16-bit Width of Peripheral Bus** When PBWIDTH = 1, the data FIFO will be increased or decreased by two peripheral bus access. This bit is used to select the order of FIFO access operations to meet the 32-bit transmitting/receiving FIFO entries. 0 = Low 16-bit read/write access first. 1 = High 16-bit read/write access first. **Note:** This bit is available while PBWIDTH = 1.
[24]	**PBWIDTH**	**Peripheral Bus Data Width Selection** This bit is used to choice the available data width of APB bus. It must be set to 1 while PDMA function is enable and it is set to 16-bit transmission mode 0 = 32 bits data width. 1 = 16 bits data width. **Note 1:** If PBWIDTH=1, the low 16 bits of 32-bit data bus are available. **Note 2:** If PBWIDTH=1, the transmitting FIFO level will be increased after two FIFO write operations. **Note 3:** If PBWIDTH=1, the receiving FIFO level will be decreased after two FIFO read operations.
[23:20]	**Reserved**	Reserved.
[19:16]	**RXTH**	**Receive FIFO Threshold Level** 0000 = 1 data word in receive FIFO. 0001 = 2 data words in receive FIFO. 0010 = 3 data words in receive FIFO. …. 1110 = 15 data words in receive FIFO. 1111 = 16 data words in receive FIFO. **Note:** When received data word number in receive buffer is larger than threshold level then RXTHIF (I2S_STATUS0[10]) flag is set.
[15:12]	**Reserved**	Reserved.

[11:8]	**TXTH**	**Transmit FIFO Threshold Level** 0000 = 0 data word in transmit FIFO. 0001 = 1 data word in transmit FIFO. 0010 = 2 data words in transmit FIFO. .... 1110 = 14 data words in transmit FIFO. 1111 = 15 data words in transmit FIFO. **Note:** If remain data word number in transmit FIFO is less than or equal to threshold level then TXTHIF (I2S_STATUS0[18]) flag is set.
[7]	**CH7ZCEN**	**Channel7 Zero-cross Detect Enable Bit** 0 = channel7 zero-cross detect Disabled. 1 = channel7 zero-cross detect Enabled. **Note 1:** This bit is available while multi-channel PCM mode and TDMCHNUM (I2S_CTL0[31:30]) = 0x3. **Note 2:** If this bit is set to 1, when channel7 data sign bit change or next shift data bits are all 0 then CH7ZCIF (I2S_STATUS1[7]) flag is set to 1. **Note 3:** If CH7ZCIF flag is set to 1, the channel7 will be mute.
[6]	**CH6ZCEN**	**Channel6 Zero-cross Detect Enable Bit** 0 = channel6 zero-cross detect Disabled. 1 = channel6 zero-cross detect Enabled. **Note 1:** This bit is available while multi-channel PCM mode and TDMCHNUM (I2S_CTL0[31:30]) = 0x3. **Note 2:** If this bit is set to 1, when channel6 data sign bit change or next shift data bits are all 0 then CH6ZCIF(I2S_STATUS1[6]) flag is set to 1. **Note 3:** If CH6ZCIF flag is set to 1, the channel6 will be mute.
[5]	**CH5ZCEN**	**Channel5 Zero-cross Detect Enable Bit** 0 = channel5 zero-cross detect Disabled. 1 = channel5 zero-cross detect Enabled. **Note 1:** This bit is available while multi-channel PCM mode and TDMCHNUM (I2S_CTL0[31:30]) = 0x2, 0x3. **Note 2:** If this bit is set to 1, when channel5 data sign bit change or next shift data bits are all 0 then CH5ZCIF(I2S_STATUS1[5]) flag is set to 1. **Note 3:** If CH5ZCIF flag is set to 1, the channel5 will be mute.
[4]	**CH4ZCEN**	**Channel4 Zero-cross Detect Enable Bit** 0 = channel4 zero-cross detect Disabled. 1 = channel4 zero-cross detect Enabled. **Note 1:** This bit is available while multi-channel PCM mode and TDMCHNUM (I2S_CTL0[31:30]) = 0x2, 0x3. **Note 2:** If this bit is set to 1, when channel4 data sign bit change or next shift data bits are all 0 then CH4ZCIF(I2S_STATUS1[4]) flag is set to 1. **Note 3:** If CH4ZCIF flag is set to 1, the channel4 will be mute.
[3]	**CH3ZCEN**	**Channel3 Zero-cross Detect Enable Bit** 0 = channel3 zero-cross detect Disabled. 1 = channel3 zero-cross detect Enabled. **Note 1:** This bit is available while multi-channel PCM mode and TDMCHNUM (I2S_CTL0[31:30]) = 0x1, 0x2, 0x3. **Note 2:** If this bit is set to 1, when channel3 data sign bit change or next shift data bits are all 0 then CH3ZCIF(I2S_STATUS1[3]) flag is set to 1. **Note 3:** If CH3ZCIF flag is set to 1, the channel3 will be mute.

[2]	**CH2ZCEN**	**Channel2 Zero-cross Detect Enable Bit** 0 = channel2 zero-cross detect Disabled. 1 = channel2 zero-cross detect Enabled. **Note 1:** This bit is available while multi-channel PCM mode and TDMCHNUM (I2S_CTL0[31:30]) = 0x1, 0x2, 0x3. **Note 2:** If this bit is set to 1, when channel2 data sign bit change or next shift data bits are all 0 then CH2ZCIF(I2S_STATUS1[2]) flag is set to 1. **Note 3:** If CH2ZCIF flag is set to 1, the channel2 will be mute.
[1]	**CH1ZCEN**	**Channel1 Zero-cross Detect Enable Bit** 0 = channel1 zero-cross detect Disabled. 1 = channel1 zero-cross detect Enabled. **Note 1:** Channel1 also means right audio channel while I^2S (FORMAT[2]=0) or 2-channel PCM mode. **Note 2:** If this bit is set to 1, when channel1 data sign bit change or next shift data bits are all 0 then CH1ZCIF(I2S_STATUS1[1]) flag is set to 1. **Note 3:** If CH1ZCIF flag is set to 1, the channel1 will be mute.
[0]	**CH0ZCEN**	**Channel0 Zero-cross Detection Enable Bit** 0 = channel0 zero-cross detect Disabled. 1 = channel0 zero-cross detect Enabled. **Note 1:** Channel0 also means left audio channel while I^2S (FORMAT[2]=0) or 2-channel PCM mode. **Note 2:** If this bit is set to 1, when channel0 data sign bit change or next shift data bits are all 0 then CH0ZCIF(I2S_STATUS1[0]) flag is set to 1. **Note 3:** If CH0ZCIF flag is set to 1, the channel0 will be mute.

I²S Clock Divider (I2S_CLKDIV)

Register	Offset	R/W	Description	Reset Value
I2S_CLKDIV	I2Sx_BA+0x04	R/W	I²S Clock Divider Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						BCLKDIV	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
BCLKDIV							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	MCLKDIV						

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17:8]	**BCLKDIV**	**Bit Clock Divider** The I²S controller will generate bit clock in Master mode. Software can program these bit fields to generate sampling rate clock frequency. F_BCLK= F_I2SCLK / (2*(BCLKDIV + 1)). **Note:** F_BCLK is the frequency of BCLK and F_I2SCLK is the frequency of I2S_CLK
[7]	**Reserved**	Reserved.
[6:0]	**MCLKDIV**	**Master Clock Divider** If chip external crystal frequency is (2xMCLKDIV)*256fs then software can program these bits to generate 256fs clock frequency to audio codec chip. If MCLKDIV is set to 0, MCLK is the same as external clock input. For example, sampling rate is 24 kHz and chip external crystal clock is 12.288 MHz, set MCLKDIV = 1. F_MCLK = F_I2SCLK/(2x(MCLKDIV)) (When MCLKDIV is >= 1). F_MCLK = F_I2SCLK (When MCLKDIV is set to 0). **Note:** F_MCLK is the frequency of MCLK, and F_I2SCLK is the frequency of the I2S_CLK

I²S Interrupt Enable Register (I2S_IEN)

Register	Offset	R/W	Description	Reset Value
I2S_IEN	I2Sx_BA+0x08	R/W	I²S Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CH7ZCIEN	CH6ZCIEN	CH5ZCIEN	CH4ZCIEN	CH3ZCIEN	CH2ZCIEN	CH1ZCIEN	CH0ZCIEN
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					TXTHIEN	TXOVFIEN	TXUDFIEN
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					RXTHIEN	RXOVFIEN	RXUDFIEN

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23]	**CH7ZCIEN**	**Channel7 Zero-cross Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled. **Note 1:** Interrupt occurs if this bit is set to 1 and channel7 zero-cross **Note 2:** This bit is available while multi-channel PCM mode and TDMCHNUM (I2S_CTL0[31:30]) = 0x3.
[22]	**CH6ZCIEN**	**Channel6 Zero-cross Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled. **Note 1:** Interrupt occurs if this bit is set to 1 and channel6 zero-cross **Note 2:** This bit is available while multi-channel PCM mode and TDMCHNUM (I2S_CTL0[31:30]) = 0x3.
[21]	**CH5ZCIEN**	**Channel5 Zero-cross Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled. **Note 1:** Interrupt occurs if this bit is set to 1 and channel5 zero-cross **Note 2:** This bit is available while multi-channel PCM mode and TDMCHNUM (I2S_CTL0[31:30]) = 0x2, 0x3.
[20]	**CH4ZCIEN**	**Channel4 Zero-cross Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled. **Note 1:** Interrupt occurs if this bit is set to 1 and channel4 zero-cross **Note 2:** This bit is available while multi-channel PCM mode and TDMCHNUM (I2S_CTL0[31:30]) = 0x2, 0x3.

[19]	**CH3ZCIEN**	**Channel3 Zero-cross Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled. **Note 1:** Interrupt occurs if this bit is set to 1 and channel3 zero-cross **Note 2:** This bit is available while multi-channel PCM mode and TDMCHNUM (I2S_CTL0[31:30]) = 0x1, 0x2, 0x3.
[18]	**CH2ZCIEN**	**Channel2 Zero-cross Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled. **Note 1:** Interrupt occurs if this bit is set to 1 and channel2 zero-cross **Note 2:** This bit is available while multi-channel PCM mode and TDMCHNUM (I2S_CTL0[31:30]) = 0x1, 0x2, 0x3.
[17]	**CH1ZCIEN**	**Channel1 Zero-cross Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled. **Note 1:** Interrupt occurs if this bit is set to 1 and channel1 zero-cross **Note 2:** Channel1 also means right audio channel while I^2S (FORMAT[2]=0) or 2-channel PCM mode.
[16]	**CH0ZCIEN**	**Channel0 Zero-cross Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled. **Note 1:** Interrupt occurs if this bit is set to 1 and channel0 zero-cross **Note 2:** Channel0 also means left audio channel while I^2S (FORMAT[2]=0) or 2-channel PCM mode.
[15:11]	**Reserved**	Reserved.
[10]	**TXTHIEN**	**Transmit FIFO Threshold Level Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled. **Note:** Interrupt occurs if this bit is set to 1 and data words in transmit FIFO is less than or equal to TXTH (I2S_CTL1[11:8]).
[9]	**TXOVFIEN**	**Transmit FIFO Overflow Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled. **Note:** Interrupt occurs if this bit is set to 1 and TXOVIF (I2S_STATUS0[17]) flag is set to 1
[8]	**TXUDFIEN**	**Transmit FIFO Underflow Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled. **Note:** Interrupt occur if this bit is set to 1 and TXUDIF (I2S_STATUS0[16]) flag is set to 1.
[7:3]	**Reserved**	Reserved.
[2]	**RXTHIEN**	**Receive FIFO Threshold Level Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled. **Note:** Interrupt occurs if this bit is set to 1 and data words in receive FIFO is larger than RXTH (I2S_CTL1[19:16]).
[1]	**RXOVFIEN**	**Receive FIFO Overflow Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled. **Note:** Interrupt occurs if this bit is set to 1 and RXOVIF (I2S_STATUS0[9]) flag is set to 1

[0]	**RXUDFIEN**	**Receive FIFO Underflow Interrupt Enable Bit** 0 = Interrupt Disabled. 1 = Interrupt Enabled. **Note:** If software reads receive FIFO when it is empty then RXUDIF (I2S_STATUS0[8]) flag is set to 1. If RXUDFIEN bit is enabled, interrupt occurs.

I²S Status Register 0 (I2S_STATUS0)

Register	Offset	R/W	Description	Reset Value
I2S_STATUS0	I2Sx_BA+0x0C	R/W	I²S Status Register 0	0x0014_1038

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved		TXBUSY	TXEMPTY	TXFULL	TXTHIF	TXOVIF	TXUDIF
15	14	13	12	11	10	9	8
Reserved			RXEMPTY	RXFULL	RXTHIF	RXOVIF	RXUDIF
7	6	5	4	3	2	1	0
Reserved		DATACH			I2STXINT	I2SRXINT	I2SINT

Bits	Description	
[31:22]	**Reserved**	Reserved.
[21]	**TXBUSY**	**Transmit Busy (Read Only)** 0 = Transmit shift buffer is empty. 1 = Transmit shift buffer is busy. **Note:** This bit is cleared to 0 when all data in transmit FIFO and shift buffer is shifted out. And set to 1 when 1st data is load to shift buffer.
[20]	**TXEMPTY**	**Transmit FIFO Empty (Read Only)** This bit reflect data word number in transmit FIFO is 0 0 = Not empty. 1 = Empty.
[19]	**TXFULL**	**Transmit FIFO Full (Read Only)** This bit reflect data word number in transmit FIFO is 16 0 = Not full. 1 = Full.
[18]	**TXTHIF**	**Transmit FIFO Threshold Interrupt Flag (Read Only)** 0 = Data word(s) in FIFO is larger than threshold level. 1 = Data word(s) in FIFO is less than or equal to threshold level. **Note:** When data word(s) in transmit FIFO is less than or equal to threshold value set in TXTH (I2S_CTL1[11:8]) the TXTHIF bit becomes to 1. It keeps at 1 till TXCNT (I2S_STATUS1[12:8]) is larger than TXTH (I2S_CTL1[11:8]) after software write TXFIFO register.
[17]	**TXOVIF**	**Transmit FIFO Overflow Interrupt Flag** 0 = No overflow. 1 = Overflow. **Note 1:** Write data to transmit FIFO when it is full and this bit set to 1 **Note 2:** Write 1 to clear this bit to 0.

[16]	**TXUDIF**	**Transmit FIFO Underflow Interrupt Flag** 0 = No underflow. 1 = Underflow. **Note 1:** This bit will be set to 1 when shift logic hardware read data from transmitting FIFO and the filling data level in transmitting FIFO is not enough for one audio frame. **Note 2:** Write 1 to clear this bit to 0.
[15:13]	**Reserved**	Reserved.
[12]	**RXEMPTY**	**Receive FIFO Empty (Read Only)** 0 = Not empty. 1 = Empty. **Note:** This bit reflects data words number in receive FIFO is 0.
[11]	**RXFULL**	**Receive FIFO Full (Read Only)** 0 = Not full. 1 = Full. **Note:** This bit reflects data words number in receive FIFO is 16.
[10]	**RXTHIF**	**Receive FIFO Threshold Interrupt Flag (Read Only)** 0 = Data word(s) in FIFO is less than or equal to threshold level. 1 = Data word(s) in FIFO is larger than threshold level. **Note:** When data word(s) in receive FIFO is larger than threshold value set in RXTH (I2S_CTL1[19:16]) the RXTHIF bit becomes to 1. It keeps at 1 till RXCNT (I2S_STATUS1[20:16]) is less than or equal to RXTH (I2S_CTL1[19:16]) after software read RXFIFO register.
[9]	**RXOVIF**	**Receive FIFO Overflow Interrupt Flag** 0 = No overflow occur. 1 = Overflow occur. **Note 1:** When receive FIFO is full and receive hardware attempt to write data into receive FIFO then this bit is set to 1, data in 1st buffer is overwritten. **Note 2:** Write 1 to clear this bit to 0.
[8]	**RXUDIF**	**Receive FIFO Underflow Interrupt Flag** 0 = No underflow occur. 1 = Underflow occur. **Note 1:** When receive FIFO is empty, and software reads the receive FIFO again. This bit will be set to 1, and it indicates underflow situation occurs. **Note 2:** Write 1 to clear this bit to 0
[7:6]	**Reserved**	Reserved.
[5:3]	**DATACH**	**Transmission Data Channel (Read Only)** This bit fields are used to indicate which audio channel is current transmit data belong. 000 = channel0 (means left channel while 2-channel I2S/PCM mode). 001 = channel1 (means right channel while 2-channel I2S/PCM mode). 010 = channel2 (available while 4-channel TDM PCM mode). 011 = channel3 (available while 4-channel TDM PCM mode). 100 = channel4 (available while 6-channel TDM PCM mode). 101 = channel5 (available while 6-channel TDM PCM mode). 110 = channel6 (available while 8-channel TDM PCM mode). 111 = channel7 (available while 8-channel TDM PCM mode).
[2]	**I2STXINT**	**I^2S Transmit Interrupt (Read Only)** 0 = No transmit interrupt. 1 = Transmit interrupt.

[1]	**I2SRXINT**	**I^2S Receive Interrupt (Read Only)** 0 = No receive interrupt. 1 = Receive interrupt.
[0]	**I2SINT**	**I^2S Interrupt Flag (Read Only)** 0 = No I^2S interrupt. 1 = I^2S interrupt. **Note:** It is wire-OR of I2STXINT and I2SRXINT bits.

I²S Status Register 1 (I2S_STATUS1)

Register	Offset	R/W	Description	Reset Value
I2S_STATUS1	I2Sx_BA+0x24	R/W	I²S Status Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			RXCNT				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			TXCNT				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CH7ZCIF	CH6ZCIF	CH5ZCIF	CH4ZCIF	CH3ZCIF	CH2ZCIF	CH1ZCIF	CH0ZCIF

Bits	Description	
[31:21]	**Reserved**	Reserved.
[20:16]	**RXCNT**	**Receive FIFO Level (Read Only)** These bits indicate the number of available entries in receive FIFO 00000 = No data. 00001 = 1 word in receive FIFO. 00010 = 2 words in receive FIFO. …. 01110 = 14 words in receive FIFO. 01111 = 15 words in receive FIFO. 10000 = 16 words in receive FIFO. Others are reserved.
[15:13]	**Reserved**	Reserved.
[12:8]	**TXCNT**	**Transmit FIFO Level (Read Only)** These bits indicate the number of available entries in transmit FIFO 00000 = No data. 00001 = 1 word in transmit FIFO. 00010 = 2 words in transmit FIFO. …. 01110 = 14 words in transmit FIFO. 01111 = 15 words in transmit FIFO. 10000 = 16 words in transmit FIFO. Others are reserved.

[7]	**CH7ZCIF**	**Channel7 Zero-cross Interrupt Flag** It indicates channel7 next sample data sign bit is changed or all data bits are 0. 0 = No zero-cross in channel7. 1 = Channel7 zero-cross is detected. **Note 1:** Write 1 to clear this bit to 0. **Note 2:** This bit is available while multi-channel PCM mode and TDMCHNUM (I2S_CTL0[31:30]) = 0x3.
[6]	**CH6ZCIF**	**Channel6 Zero-cross Interrupt Flag** It indicates channel6 next sample data sign bit is changed or all data bits are 0. 0 = No zero-cross in channel6. 1 = Channel6 zero-cross is detected. **Note 1:** Write 1 to clear this bit to 0. **Note 2:** This bit is available while multi-channel PCM mode and TDMCHNUM (I2S_CTL0[31:30]) = 0x3.
[5]	**CH5ZCIF**	**Channel5 Zero-cross Interrupt Flag** It indicates channel5 next sample data sign bit is changed or all data bits are 0. 0 = No zero-cross in channel5. 1 = Channel5 zero-cross is detected. **Note 1:** Write 1 to clear this bit to 0. **Note 2:** This bit is available while multi-channel PCM mode and TDMCHNUM (I2S_CTL0[31:30]) = 0x2, 0x3.
[4]	**CH4ZCIF**	**Channel4 Zero-cross Interrupt Flag** It indicates channel4 next sample data sign bit is changed or all data bits are 0. 0 = No zero-cross in channel4. 1 = Channel4 zero-cross is detected. **Note 1:** Write 1 to clear this bit to 0. **Note 2:** This bit is available while multi-channel PCM mode and TDMCHNUM (I2S_CTL0[31:30]) = 0x2, 0x3.
[3]	**CH3ZCIF**	**Channel3 Zero-cross Interrupt Flag** It indicates channel3 next sample data sign bit is changed or all data bits are 0. 0 = No zero-cross in channel3. 1 = Channel3 zero-cross is detected. **Note 1:** Write 1 to clear this bit to 0. **Note 2:** This bit is available while multi-channel PCM mode and TDMCHNUM (I2S_CTL0[31:30]) = 0x1, 0x2, 0x3.
[2]	**CH2ZCIF**	**Channel2 Zero-cross Interrupt Flag** It indicates channel2 next sample data sign bit is changed or all data bits are 0. 0 = No zero-cross in channel2. 1 = Channel2 zero-cross is detected. **Note 1:** Write 1 to clear this bit to 0. **Note 2:** This bit is available while multi-channel PCM mode and TDMCHNUM (I2S_CTL0[31:30]) = 0x1, 0x2, 0x3.
[1]	**CH1ZCIF**	**Channel1 Zero-cross Interrupt Flag** It indicates channel1 next sample data sign bit is changed or all data bits are 0. 0 = No zero-cross in channel1. 1 = Channel1 zero-cross is detected. **Note 1:** Write 1 to clear this bit to 0. **Note 2:** Channel1 also means right audio channel while I^2S (FORMAT[2]=0) or 2-channel PCM mode.

[0]	**CH0ZCIF**	**Channel0 Zero-cross Interrupt Flag** It indicates channel0 next sample data sign bit is changed or all data bits are 0. 0 = No zero-cross in channel0. 1 = Channel0 zero-cross is detected. **Note 1:** Write 1 to clear this bit to 0. **Note 2:** Channel0 also means left audio channel while I^2S (FORMAT[2]=0) or 2-channel PCM mode.

I²S Transmit FIFO (I2S_TXFIFO)

Register	Offset	R/W	Description	Reset Value
I2S_TXFIFO	I2Sx_BA+0x10	W	I²S Transmit FIFO Register	0x0000_0000

31	30	29	28	27	26	25	24
TXFIFO							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TXFIFO							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TXFIFO							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TXFIFO							

Bits	Description	
[31:0]	**TXFIFO**	**Transmit FIFO Bits** The I²S contains 16 words (16x32 bits) data buffer for data transmit. Write data to this register to prepare data for transmit. The remaining word number is indicated by TXCNT (I2S_STATUS1[12:8]).

I²S Receive FIFO (I2S_RXFIFO)

Register	Offset	R/W	Description	Reset Value
I2S_RXFIFO	I2Sx_BA+0x14	R	I²S Receive FIFO Register	0x0000_0000

31	30	29	28	27	26	25	24
RXFIFO							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RXFIFO							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RXFIFO							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RXFIFO							

Bits	Description	
[31:0]	**RXFIFO**	**Receive FIFO Bits** I²S contains 16 words (16x32 bits) data buffer for data receive. Read this register to get data in FIFO. The remaining data word number is indicated by RXCNT (I2S_STATUS1[20:16]).

6.30 High Speed USB 2.0 Device Controller (HSUSBD)

6.30.1 Overview

This chip is equipped with a USB 2.0 HS/FS Device Controller. The USB Device Controller interfaces the AHB bus and the UTMI bus. The USB Device Controller contains both the AHB master interface and AHB slave interface. CPU programs the USB Device Controller control and status registers through the AHB slave interface. The USB Device Controller is complaint with USB 2.0 specification and it contains 8 bidirectional endpoints in addition to control endpoint. These endpoints could be configured to BULK, INTERRUPT or ISOCHRONOUS. The USB Device Controller has a built-in DMA to relieve the load of CPU.

6.30.2 Features

- Compliant with Universal Serial Bus (USB) Specification Revision 2.0.
- Supports up to eight bidirectional endpoints, in addition to control endpoint 0.
- Supports Control, Bulk, Interrupt, Isochronous transfers.
- Supports Descriptor (Scatter gather) DMA operation.
- Supports LPM feature.
- Supports V_{BUS}/Resume wake-up from system power-down mode.

6.30.3 Block Diagram

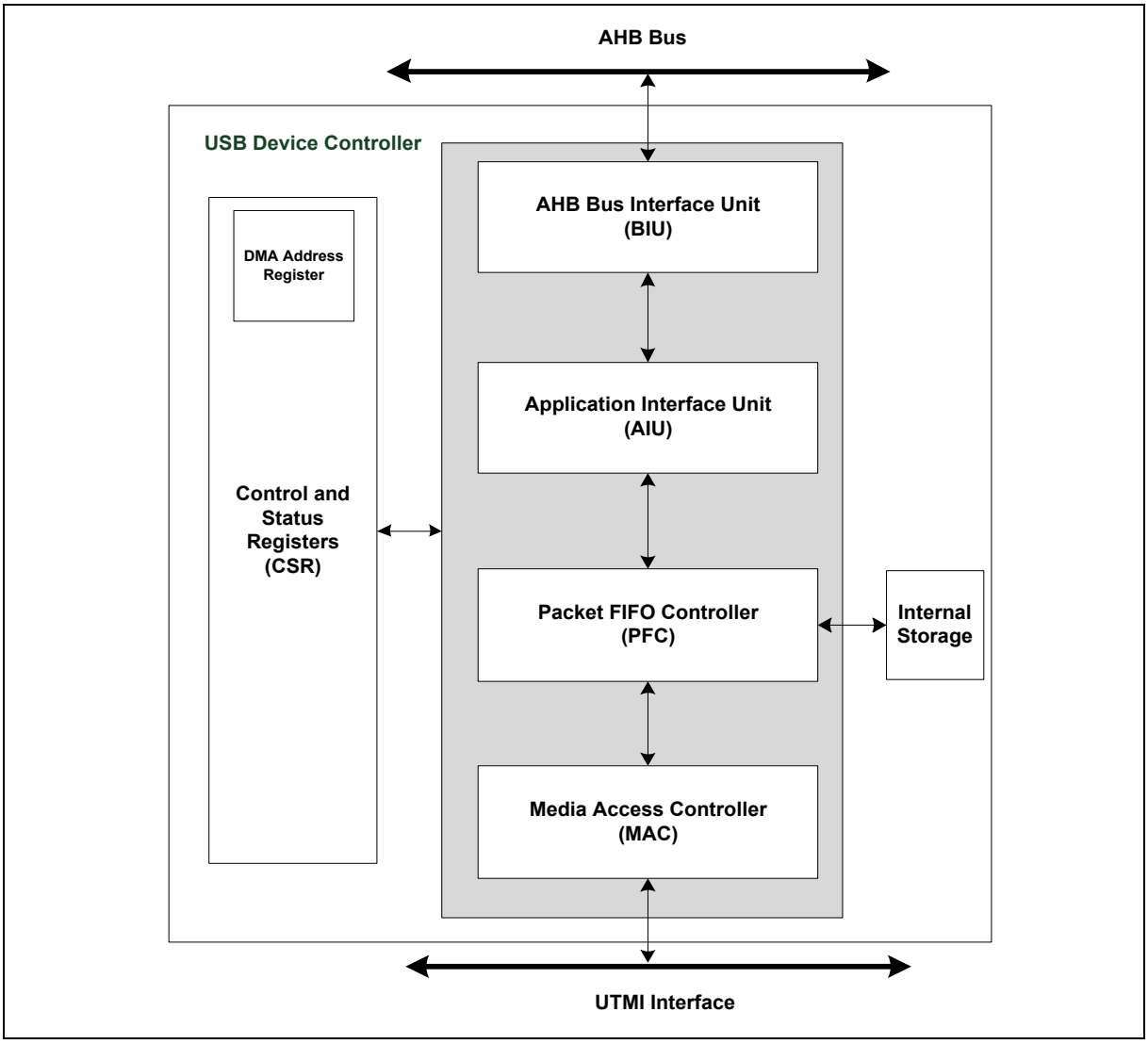


Figure 6.30-1 USB 2.0 HS Device Controller Block Diagram

6.30.4 Basic Configuration

6.30.4.1 HSUSBD Controller Basic Configuration

- Clock Source Configuration
  - Enable HSUSBD Bus clock in USBDEN (CLK_SYSCLK0[19]).
  - Enable HSUSBD engine clock in PHY0SUSPEND (SYS_USBPMISCR[1])
- Reset Configuration
  - Reset HSUSBD controller in HSUSBDRST (SYS_IPRST0[22]).

6.30.5 Functional Description

Device Controller

The Device controller includes 5 functional blocks, including AHB Bus Interface Unit (BIU), Control and Status Registers (CSR), Application Interface Unit (AIU), Packet FIFO Controller (PFC), Media Access Controller (MAC).

6.30.5.2 AHB Bus Interface Unit (BIU)

The USB Device Controller is connected to the system by the AHB bus. The design requires both Master and Slave bus operations. AHB Master interface for Descriptor DMA read/write access. AHB Slave interface for accessing Control and Status Registers (CSR).

6.30.5.3 Control and Status Registers (CSR)

The CSR block resides in the AHB clock domain. The AHB Slave Interface unit can write and read registers, and the controller can set or reset the interrupt and status bit. The interrupts have priority over AHB Slave Interface update.

6.30.5.4 Application Interface Unit (AIU)

The AIU is responsible for the following functions

1. Generating and writing the delimiter into the transmit FIFOs for transmit packets in Device (IN)
2. Writing a token into the Request Queue for transmit transactions in Device
3. Selecting the Periodic or Non-periodic Request Queue based on the type of transaction
4. Generating interrupts for both transmit and receiving operations in Device
5. Address decoding for selecting the appropriate transmit FIFOs and queues
6. Generating DMA interface signals to access an internal DMA controller for data transfer

6.30.5.5 Packet FIFO Controller (PFC)

The PFC handles the periodic/non-periodic packet transaction with both receive and transmit, and the data transmit interface with the internal storage.

6.30.5.6 Media Access Controller (MAC)

The MAC module handles USB transactions, and device protocols.

USB Transaction Handling

The MAC decodes and checks the integrity of token packets as it receives them from the host. If the received token is a valid OUT or SETUP token, the MAC waits and checks the PID of the following data packets, then writes the data to the RxFIFO when it is available. After the packets are completed, the MAC checks the data integrity, sends the appropriate handshake when required to the host, and writes the transaction status to the receiving status queue.

If an IN token is received and the data is available in the FIFO, the MAC reads the data, builds the data packet and sends it, waits for a handshake packet, if any, from the host, then updates the transactions status to the PFC.

Device Protocol Handling

The MAC handles the USB reset sequence and speed enumeration process to determine the USB operating speed. The MAC detects USB suspend and resume signaling form the host, initiates remote wakeup, handles soft connect and disconnect, decodes and tracks SOF packets.

6.30.5.7 V_{BUS}/Resume Wakeup Function

The HSUSBD supports V_{BUS} and resume wakeup if system enter power-down mode, with two separate enable bits VBUSWKEN (SYS_MISCFCR0[18]) and LNSTWKEN (SYS_MISCFCR0[19]).

6.30.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
HSUSBD Base Address: **HSUSBD_BA = 0x4020_0000**				
GOTGCTL	HSUSBD_BA+0x000	R/W	Control and Status Register	0x000D_0000
GAHBCFG	HSUSBD_BA+0x008	R/W	AHB Configuration Register	0x0000_0000
GUSBCFG	HSUSBD_BA+0x00c	R/W	USB Configuration Register	0x0000_1400
GRSTCTL	HSUSBD_BA+0x010	R/W	Reset Register	0x0000_0000
GINTSTS	HSUSBD_BA+0x014	R/W	Interrupt Register	0x5400_0020
GINTMSK	HSUSBD_BA+0x018	R/W	Interrupt Mask Register	0x0000_0000
GRXSTSR	HSUSBD_BA+0x01c	R	Receive Status Debug Read Register	0x0000_0000
GRXSTSP	HSUSBD_BA+0x020	R	Receive Status Read/Pop Register	0x0000_0000
GRXFSIZ	HSUSBD_BA+0x024	R/W	Receive FIFO Size Register	0x0000_0100
GNPTXFSIZ	HSUSBD_BA+0x028	R/W	Non-periodic Transmit FIFO Size Register	0x0020_0100
GLPMCFG	HSUSBD_BA+0x054	R/W	LPM Config Register	0x0000_0000
GDFIFOCFG	HSUSBD_BA+0x05c	R/W	Global DFIFO Configuration Register	0x0BB8_0C00
GREFCLK	HSUSBD_BA+0x064	R/W	ref_clk Control Register	0x0000_0000
GINTMSK2	HSUSBD_BA+0x068	R/W	Interrupt Mask Register 2	0x0000_0000
GINTSTS2	HSUSBD_BA+0x06c	R/W	Interrupt Register 2	0x0000_0000
DIEPTXF1	HSUSBD_BA+0x104	R/W	Device IN Endpoint Transmit FIFO Size Register 1	0x0100_0120
DIEPTXF2	HSUSBD_BA+0x108	R/W	Device IN Endpoint Transmit FIFO Size Register 2	0x0100_0220
DIEPTXF3	HSUSBD_BA+0x10c	R/W	Device IN Endpoint Transmit FIFO Size Register 3	0x0100_0320
DIEPTXF4	HSUSBD_BA+0x110	R/W	Device IN Endpoint Transmit FIFO Size Register 4	0x0100_0420
DIEPTXF5	HSUSBD_BA+0x114	R/W	Device IN Endpoint Transmit FIFO Size Register 5	0x0100_0520
DIEPTXF6	HSUSBD_BA+0x118	R/W	Device IN Endpoint Transmit FIFO Size Register 6	0x0100_0620
DIEPTXF7	HSUSBD_BA+0x11c	R/W	Device IN Endpoint Transmit FIFO Size Register 7	0x0100_0720
DIEPTXF8	HSUSBD_BA+0x120	R/W	Device IN Endpoint Transmit FIFO Size Register 8	0x0100_0820
DCFG	HSUSBD_BA+0x800	R/W	Device Configuration Register	0x0812_0000
DCTL	HSUSBD_BA+0x804	R/W	Device Control Register	0x0000_0002
DSTS	HSUSBD_BA+0x808	R	Device Status Register	0x0000_0002
DIEPMSK	HSUSBD_BA+0x810	R/W	Device IN Endpoint Common Interrupt Mask Register	0x0000_0000

DOEPMSK	HSUSBD_BA+0x814	R/W	Device OUT Endpoint Common Interrupt Mask Register	0x0000_0000
DAINT	HSUSBD_BA+0x818	R	Device All Endpoints Interrupt Register	0x0000_0000
DAINTMSK	HSUSBD_BA+0x81c	R/W	Device All Endpoints Interrupt Mask Register	0x0000_0000
DTHRCTL	HSUSBD_BA+0x830	R/W	Device Threshold Control Register	0x0C10_0020
DIEPEMPMSK	HSUSBD_BA+0x834	R/W	Device IN Endpoint FIFO Empty Interrupt Mask Register	0x0000_0000
DIEPCTL0	HSUSBD_BA+0x900	R/W	Device Control IN Endpoint 0 Control Register	0x0000_8000
DIEPINT0	HSUSBD_BA+0x908	R/W	Device IN Endpoint 0 Interrupt Register	0x0000_0080
DIEPDMA0	HSUSBD_BA+0x914	R/W	Device IN Endpoint 0 DMA Address Register	0x0000_0000
DTXFSTS0	HSUSBD_BA+0x918	R	Device IN Endpoint Transmit FIFO Status Register 0	0x0000_0020
DIEPDMAB0	HSUSBD_BA+0x91c	R	Device IN Endpoint 16 Buffer Address Register	0x0000_0000
DIEPCTL1	HSUSBD_BA+0x920	R/W	Device Control IN Endpoint 1 Control Register	0x0000_0000
DIEPINT1	HSUSBD_BA+0x928	R/W	Device IN Endpoint 1 Interrupt Register	0x0000_0080
DIEPDMA1	HSUSBD_BA+0x934	R/W	Device IN Endpoint 1 DMA Address Register	0x0000_0000
DTXFSTS1	HSUSBD_BA+0x938	R	Device IN Endpoint Transmit FIFO Status Register 1	0x0000_0020
DIEPDMAB1	HSUSBD_BA+0x93c	R	Device IN Endpoint 1 Buffer Address Register	0x0000_0000
DIEPCTL2	HSUSBD_BA+0x940	R/W	Device Control IN Endpoint 2 Control Register	0x0000_0000
DIEPINT2	HSUSBD_BA+0x948	R/W	Device IN Endpoint 2 Interrupt Register	0x0000_0080
DIEPDMA2	HSUSBD_BA+0x954	R/W	Device IN Endpoint 2 DMA Address Register	0x0000_0000
DTXFSTS2	HSUSBD_BA+0x958	R	Device IN Endpoint Transmit FIFO Status Register 2	0x0000_0020
DIEPDMAB2	HSUSBD_BA+0x95c	R	Device IN Endpoint 2 Buffer Address Register	0x0000_0000
DIEPCTL3	HSUSBD_BA+0x960	R/W	Device Control IN Endpoint 3 Control Register	0x0000_0000
DIEPINT3	HSUSBD_BA+0x968	R/W	Device IN Endpoint 3 Interrupt Register	0x0000_0080
DIEPDMA3	HSUSBD_BA+0x974	R/W	Device IN Endpoint 3 DMA Address Register	0x0000_0000
DTXFSTS3	HSUSBD_BA+0x978	R	Device IN Endpoint Transmit FIFO Status Register 3	0x0000_0020
DIEPDMAB3	HSUSBD_BA+0x97c	R	Device IN Endpoint 3 Buffer Address Register	0x0000_0000
DIEPCTL4	HSUSBD_BA+0x980	R/W	Device Control IN Endpoint 4 Control Register	0x0000_0000
DIEPINT4	HSUSBD_BA+0x988	R/W	Device IN Endpoint 4 Interrupt Register	0x0000_0080
DIEPDMA4	HSUSBD_BA+0x994	R/W	Device IN Endpoint 4 DMA Address Register	0x0000_0000
DTXFSTS4	HSUSBD_BA+0x998	R	Device IN Endpoint Transmit FIFO Status Register 4	0x0000_0020
DIEPDMAB4	HSUSBD_BA+0x99c	R	Device IN Endpoint 4 Buffer Address Register	0x0000_0000
DIEPCTL5	HSUSBD_BA+0x9a0	R/W	Device Control IN Endpoint 5 Control Register	0x0000_0000

DIEPINT5	HSUSBD_BA+0x9a8	R/W	Device IN Endpoint 5 Interrupt Register	0x0000_0080
DIEPDMA5	HSUSBD_BA+0x9b4	R/W	Device IN Endpoint 5 DMA Address Register	0x0000_0000
DTXFSTS5	HSUSBD_BA+0x9b8	R	Device IN Endpoint Transmit FIFO Status Register 5	0x0000_0020
DIEPDMAB5	HSUSBD_BA+0x9bc	R	Device IN Endpoint 5 Buffer Address Register	0x0000_0000
DIEPCTL6	HSUSBD_BA+0x9c0	R/W	Device Control IN Endpoint 6 Control Register	0x0000_0000
DIEPINT6	HSUSBD_BA+0x9c8	R/W	Device IN Endpoint 6 Interrupt Register	0x0000_0080
DIEPDMA6	HSUSBD_BA+0x9d4	R/W	Device IN Endpoint 6 DMA Address Register	0x0000_0000
DTXFSTS6	HSUSBD_BA+0x9d8	R	Device IN Endpoint Transmit FIFO Status Register 6	0x0000_0020
DIEPDMAB6	HSUSBD_BA+0x9dc	R	Device IN Endpoint 6 Buffer Address Register	0x0000_0000
DIEPCTL7	HSUSBD_BA+0x9e0	R/W	Device Control IN Endpoint 7 Control Register	0x0000_0000
DIEPINT7	HSUSBD_BA+0x9e8	R/W	Device IN Endpoint 7 Interrupt Register	0x0000_0080
DIEPDMA7	HSUSBD_BA+0x9f4	R/W	Device IN Endpoint 7 DMA Address Register	0x0000_0000
DTXFSTS7	HSUSBD_BA+0x9f8	R	Device IN Endpoint Transmit FIFO Status Register 7	0x0000_0020
DIEPDMAB7	HSUSBD_BA+0x9fc	R	Device IN Endpoint 7 Buffer Address Register	0x0000_0000
DIEPCTL8	HSUSBD_BA+0xa00	R/W	Device Control IN Endpoint 8 Control Register	0x0000_0000
DIEPINT8	HSUSBD_BA+0xa08	R/W	Device IN Endpoint 8 Interrupt Register	0x0000_0080
DIEPDMA8	HSUSBD_BA+0xa14	R/W	Device IN Endpoint 8 DMA Address Register	0x0000_0000
DTXFSTS8	HSUSBD_BA+0xa18	R	Device IN Endpoint Transmit FIFO Status Register 8	0x0000_0020
DIEPDMAB8	HSUSBD_BA+0xa1c	R	Device IN Endpoint 8 Buffer Address Register	0x0000_0000
DOEPCTL0	HSUSBD_BA+0xb00	R/W	Device Control OUT Endpoint 0 Control Register	0x0000_0000
DOEPINT0	HSUSBD_BA+0xb08	R/W	Device OUT Endpoint 0 Interrupt Register	0x0000_0000
DOEPDMA0	HSUSBD_BA+0xb14	R/W	Device OUT Endpoint 0 DMA Address Register	0x0000_0000
DOEPDMAB0	HSUSBD_BA+0xb1c	R	Device OUT Endpoint 16 Buffer Address Register	0x0000_0000
DOEPCTL1	HSUSBD_BA+0xb20	R/W	Device Control OUT Endpoint 1 Control Register	0x0000_0000
DOEPINT1	HSUSBD_BA+0xb28	R/W	Device OUT Endpoint 1 Interrupt Register	0x0000_0000
DOEPDMA1	HSUSBD_BA+0xb34	R/W	Device OUT Endpoint 1 DMA Address Register	0x0000_0000
DOEPDMAB1	HSUSBD_BA+0xb3c	R	Device OUT Endpoint 1 Buffer Address Register	0x0000_0000
DOEPCTL2	HSUSBD_BA+0xb40	R/W	Device Control OUT Endpoint 2 Control Register	0x0000_0000
DOEPINT2	HSUSBD_BA+0xb48	R/W	Device OUT Endpoint 2 Interrupt Register	0x0000_0000
DOEPDMA2	HSUSBD_BA+0xb54	R/W	Device OUT Endpoint 2 DMA Address Register	0x0000_0000
DOEPDMAB2	HSUSBD_BA+0xb5c	R	Device OUT Endpoint 2 Buffer Address Register	0x0000_0000

DOEPCTL3	HSUSBD_BA+0xb60	R/W	Device Control OUT Endpoint 3 Control Register	0x0000_0000
DOEPINT3	HSUSBD_BA+0xb68	R/W	Device OUT Endpoint 3 Interrupt Register	0x0000_0000
DOEPDMA3	HSUSBD_BA+0xb74	R/W	Device OUT Endpoint 3 DMA Address Register	0x0000_0000
DOEPDMAB3	HSUSBD_BA+0xb7c	R	Device OUT Endpoint 3 Buffer Address Register	0x0000_0000
DOEPCTL4	HSUSBD_BA+0xb80	R/W	Device Control OUT Endpoint 4 Control Register	0x0000_0000
DOEPINT4	HSUSBD_BA+0xb88	R/W	Device OUT Endpoint 4 Interrupt Register	0x0000_0000
DOEPDMA4	HSUSBD_BA+0xb94	R/W	Device OUT Endpoint 4 DMA Address Register	0x0000_0000
DOEPDMAB4	HSUSBD_BA+0xb9c	R	Device OUT Endpoint 4 Buffer Address Register	0x0000_0000
DOEPCTL5	HSUSBD_BA+0xba0	R/W	Device Control OUT Endpoint 5 Control Register	0x0000_0000
DOEPINT5	HSUSBD_BA+0xba8	R/W	Device OUT Endpoint 5 Interrupt Register	0x0000_0000
DOEPDMA5	HSUSBD_BA+0xbb4	R/W	Device OUT Endpoint 5 DMA Address Register	0x0000_0000
DOEPDMAB5	HSUSBD_BA+0xbbc	R	Device OUT Endpoint 5 Buffer Address Register	0x0000_0000
DOEPCTL6	HSUSBD_BA+0xbc0	R/W	Device Control OUT Endpoint 6 Control Register	0x0000_0000
DOEPINT6	HSUSBD_BA+0xbc8	R/W	Device OUT Endpoint 6 Interrupt Register	0x0000_0000
DOEPDMA6	HSUSBD_BA+0xbd4	R/W	Device OUT Endpoint 6 DMA Address Register	0x0000_0000
DOEPDMAB6	HSUSBD_BA+0xbdc	R	Device OUT Endpoint 6 Buffer Address Register	0x0000_0000
DOEPCTL7	HSUSBD_BA+0xbe0	R/W	Device Control OUT Endpoint 7 Control Register	0x0000_0000
DOEPINT7	HSUSBD_BA+0xbe8	R/W	Device OUT Endpoint 7 Interrupt Register	0x0000_0000
DOEPDMA7	HSUSBD_BA+0xbf4	R/W	Device OUT Endpoint 7 DMA Address Register	0x0000_0000
DOEPDMAB7	HSUSBD_BA+0xbfc	R	Device OUT Endpoint 7 Buffer Address Register	0x0000_0000
DOEPCTL8	HSUSBD_BA+0xc00	R/W	Device Control OUT Endpoint 8 Control Register	0x0000_0000
DOEPINT8	HSUSBD_BA+0xc08	R/W	Device OUT Endpoint 8 Interrupt Register	0x0000_0000
DOEPDMA8	HSUSBD_BA+0xc14	R/W	Device OUT Endpoint 8 DMA Address Register	0x0000_0000
DOEPDMAB8	HSUSBD_BA+0xc1c	R	Device OUT Endpoint 8 Buffer Address Register	0x0000_0000
PCGCCTL	HSUSBD_BA+0xe00	R/W	Power and Clock Gating Control Register	0x0000_0000
PCGCCTL1	HSUSBD_BA+0xe04	R/W	Power and Clock Gating Control Register1	0x0000_0000

6.30.7 Register Description

GOTGCTL Register (GOTGCTL)

Register	Offset	R/W	Description	Reset Value
GOTGCTL	HSUSBD_BA+0x000	R/W	Control and Status Register	0x000D_0000

31	30	29	28	27	26	25	24
Testmode_corr_eUSB2	Reserved			ChirpEn	MultValidBC		
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							

Bits	Description	
[31]	**Testmode_corr_eUSB2**	**UTMI IF correction for eUSB2 PHY during Test mode** This bit is used to modify the behavior of UTMI 8-bit interface signals during test J and test K sequences when eUSB2 PHY is used. When this bit is set to 1'b1, the controller asserts utmi_txvalid and utmi_opmode in the same cycle during test J or test K sequence execution. **Note:** This bit is applicable only if eUSB2 PHY is used with 8-bit UTMI interface. 0x0 (eUSB2_corr_disable): The controller asserts utmi_txvalid one cycle later than utmi_opmode. 0x1 (eUSB2_corr_enable): The controller asserts utmi_txvalid and utmi_opmode in the same cycle.
[30:28]	**Reserved**	**Reserved.**
[27]	**ChirpEn**	**Mode: Device Only** This bit when programmed to 1'b1 results in the core asserting chirp_on before sending an actual Chirp "K" signal on USB. 0x0 (CHIRP_DISABLE): The controller does not assert chirp_on before sending an actual Chirp "K" signal on USB. 0x1 (CHIRP_ENABLE): The controller asserts chirp_on before sending an actual Chirp "K" signal on USB.
[26:0]	**Reserved**	**Reserved.**

GAHBCFG Register (GAHBCFG)

Register	Offset	R/W	Description	Reset Value
GAHBCFG	HSUSBD_BA+0x008	R/W	AHB Configuration Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							InvDescEndianess
23	22	21	20	19	18	17	16
AHBSingle	Reserved						
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		DMAEn	HBstLen				GlblIntrMsk

Bits	Description	
[31:25]	**Reserved**	**Reserved.**
[24]	**InvDescEndianess**	**Invert Descriptor Endianess** 1'b0: Descriptor Endianness is same as AHB Master Endianness. 1'b1: If the AHB Master endianness is Big Endian, the Descriptor Endianness is Little Endian. If the AHB Master endianness is Little Endian, the Descriptor Endianness is Big Endian. **Values:** 0x0 (DISABLE): Descriptor Endianness is same as AHB Master Endianness 0x1 (ENABLE): Descriptor Endianness is opposite to AHB Master Endianness
[23]	**AHBSingle**	**AHB Single Support** This bit when programmed supports Single transfers for the remaining data in a transfer when the core is operating in DMA mode. 1'b0: The remaining data in the transfer is sent using INCR burst size. 1'b1: The remaining data in the transfer is sent using Single burst size. **Note:** If this feature is enabled, the AHB RETRY and SPLIT transfers still have INCR burst type. Enable this feature when the AHB Slave connected to the core does not support INCR burst (and when Split, and Retry transactions are not being used in the bus). **Values:** 0x0 (INCRBURST): The remaining data in the transfer is sent using INCR burst size 0x1 (SINGLEBURST): The remaining data in the transfer is sent using Single burst size
[22:21]	**Reserved**	Reserved.
[20:6]		**Reserved.**
[5]	**DMAEn**	**DMA Enable** **Reset**: 1'b0 **Values:** 0x1 (DMAMODE): Core operates in a DMA mode
[4:1]	**HBstLen**	**Burst Length/Type** Internal DMA Mode AHB Master burst type:

		4'b0000 Single 4'b0001 INCR 4'b0011 INCR4 4'b0101 INCR8 4'b0111 INCR16 Others: Reserved **Values:** 0x0 (WORD1ORSINGLE): 1 word or single 0x1 (WORD4ORINCR): 4 words or INCR 0x3 (WORD16ORINCR4): 16 words or INCR4 0x5 (WORD64ORINCR8): 64 words or INCR8 0x7 (WORD256ORINCR16): 256 words or INCR16
[0]	**GlblIntrMsk**	**Global Interrupt Mask** The application uses this bit to mask or unmask the interrupt line assertion to itself. Irrespective of this bit's setting, the interrupt status registers are updated by the controller. 1'b0: Mask the interrupt assertion to the application. 1'b1: Unmask the interrupt assertion to the application. **Values:** 0x0 (MASK): Mask the interrupt assertion to the application 0x1 (NOMASK): Unmask the interrupt assertion to the application.

GUSBCFG Register (GUSBCFG)

Register	Offset	R/W	Description	Reset Value
GUSBCFG	HSUSBD_BA+0x00c	R/W	USB Configuration Register	0x0000_1400

31	30	29	28	27	26	25	24
Reserved			TxEndDelay	Reserved			
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		USBTrdTim				Reserved	
7	6	5	4	3	2	1	0
Reserved					TOutCal		

Bits	Description	
[31:29]	**Reserved**	**Reserved.**
[28]	**TxEndDelay**	**Tx End Delay** Writing 1'b1 to this bit enables the controller to follow the TxEndDelay timings as per UTMI+ specification 1.05 section 4.1.5 for opmode signal during remote wakeup. 1'b0 : Normal Mode. 1'b1 : Tx End delay. **Values:** 0x0 (DISABLED): Normal Mode 0x1 (ENABLED): Tx End delay
[27:14]	**Reserved**	Reserved.
[13:10]	**USBTrdTim**	**USB Turnaround Time** Sets the turnaround time in PHY clocks. Specifies the response time for a MAC request to the Packet FIFO Controller (PFC) to fetch data from the DFIFO (Interanl Storage). This must be programmed to 4'h9: When the MAC interface is 8-bit UTMI+ . **Values:** 0x9 (TURNTIME8BIT): MAC interface is 8-bit UTMI+.
[9:3]	**Reserved**	**Reserved.**
[2:0]	**TOutCal**	**HS/FS Timeout Calibration** The number of PHY clocks that the application programs in this field is added to the high-speed/full-speed interpacket timeout duration in the core to account for any additional delays introduced by the PHY. This can be required, because the delay introduced by the PHY in generating the linestate condition can vary from one PHY to another. The USB standard timeout value for high-speed operation is 736 to 816 (inclusive) bit times. The USB standard timeout value for full-speed operation is 16 to 18 (inclusive) bit times. The application must program this field based on the speed of enumeration. The number of bit times added per PHY clock are as follows: **High-speed operation:** One 60- MHz PHY clock = 8 bit times. **Full-speed operation:** One 60- MHz PHY clock = 0.2 bit times.

GRSTCTL Register (GRSTCTL)

Register	Offset	R/W	Description	Reset Value
GRSTCTL	HSUSBD_BA+0x010	R/W	Reset Register	0x0000_0000

31	30	29	28	27	26	25	24
AHBIdle	Reserved						
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved					TxFNum		
7	6	5	4	3	2	1	0
TxFNum		TxFFIsh	RxFFIsh	Reserved		PIUFSSftRst	CSftRst

Bits	Description	
[31]	**AHBIdle**	**AHB Master Idle (Read Only)** Indicates that the AHB Master State Machine is in the IDLE condition. **Values:** 0x0 (INACTIVE): Not Idle 0x1 (ACTIVE): AHB Master Idle
[30:11]	**Reserved**	**Reserved.**
[10:6]	**TxFNum**	**TxFIFO Number** This is the FIFO number that must be flushed using the TxFIFO Flush bit. This field must not be changed until the core clears the TxFIFO Flush bit. 5'h0: Tx FIFO 0 flush in device mode when in dedicated FIFO mode 5'h1: TXFIFO 1 flush in device mode when in dedicated FIFO mode 5'h2: TXFIFO 2 flush in device mode when in dedicated FIFO mode ... 5'h8 TXFIFO 8 flush in device mode when in dedicated FIFO mode 5'h10: Flush all the transmit FIFOs **Values:** 0x0 (TXF0): TXFIFO 0 flush in device mode when in dedicated FIFO mode 0x1 (TXF1): TXFIFO 1 flush in device mode when in dedicated FIFO mode 0x2 (TXF2): TXFIFO 2 flush in device mode when in dedicated FIFO mode 0x3 (TXF3): TXFIFO 3 flush in device mode when in dedicated FIFO mode 0x4 (TXF4): TXFIFO 4 flush in device mode when in dedicated FIFO mode 0x5 (TXF5): TXFIFO 5 flush in device mode when in dedicated FIFO mode 0x6 (TXF6): TXFIFO 6 flush in device mode when in dedicated FIFO mode 0x7 (TXF7): TXFIFO 7 flush in device mode when in dedicated FIFO mode 0x8 (TXF8): TXFIFO 8 flush in device mode when in dedicated FIFO mode

		0x10 (TXF16): Flush all the transmit FIFOs
[5]	**TxFFIsh**	**TxFIFO Flush** This bit selectively flushes a single or all transmit FIFOs, but cannot do so If the core is in the midst of a transaction. The application must write this bit only after checking that the core is neither writing to the TxFIFO nor reading from the TxFIFO. Verify using these registers: ReadNAK Effective Interrupt ensures the core is not reading from the FIFO WriteGRSTCTL.AHBIdle ensures the core is not writing anything to the FIFO. Flushing is normally recommended when FIFOs are reconfigured. FIFO flushing is also recommended during device endpoint disable. The application must wait until the core clears this bit before performing any operations. This bit takes eight clocks to clear, using the slower clock of phy_clk or hclk. **Values:** 0x0 (INACTIVE): No Flush 0x1 (ACTIVE): Selectively flushes a single or all transmit FIFOs
[4]	**RxFFIsh**	**RxFIFO Flush** The application can flush the entire RxFIFO using this bit, but must first ensure that the core is not in the middle of a transaction. The application must only write to this bit after checking that the controller is neither reading from the RxFIFO nor writing to the RxFIFO. The application must wait until the bit is cleared before performing any other operations. This bit requires eight clocks (slowest of PHY or AHB clock) to clear. **Values:** 0x0 (INACTIVE): Does not flush the entire RxFIFO 0x1 (ACTIVE): Flushes the entire RxFIFO
[3:2]	**Reserved**	**Reserved.**
[1]	**PIUFSSftRst**	**PIU FS Dedicated Controller Soft Reset** Resets the PIU FS Dedicated Controller All module state machines in FS Dedicated Controller of PIU are reset to the IDLE state. Used to reset the FS Dedicated controller in PIU in case of any PHY Errors like Loss of activity or Babble Error resulting in the PHY remaining in RX state for more than one frame boundary. This is a self clearing bit and core clears this bit after all the necessary logic is reset in the core. **Values:** 0x0 (RESET_INACTIVE): No Reset 0x1 (RESET_ACTIVE): PIU FS Dedicated Controller Soft Reset
[0]	**CSftRst**	**Core Soft Reset** Resets the hclk and phy_clock domains as follows: Clears the interrupts and all the CSR registers except the following register bits: GUSBCFG.TxEndDelay DCFG.DevSpd DCTL.SftDiscon All module state machines All module state machines (except the AHB Slave Unit) are reset to the IDLE state, and all the transmit FIFOs and the receive FIFO are flushed. Any transactions on the AHB Master are terminated as soon as possible, after gracefully completing the last data phase of an AHB transfer. Any transactions on the USB are terminated immediately. The application can write to this bit any time it wants to reset the core. This is a self-clearing bit and the core clears this bit after all the necessary logic is reset in the core, which can take several clocks, depending on the current state of the core. Once this bit is cleared software must wait at least 3 PHY clocks before doing any access to the PHY domain (synchronization delay). Software must also must check that bit 31 of this register is 1 (AHB Master is IDLE) before starting any operation. Typically software reset is used during software development and also when you dynamically change the

		PHY selection bits in the USB configuration registers listed above. When you change the PHY, the corresponding clock for the PHY is selected and used in the PHY domain. Once a new clock is selected, the PHY domain has to be reset for proper operation. **Values:** 0x0 (NOTACTIVE): No reset 0x1 (ACTIVE): Resets hclk and phy_clock domains

GINTSTS Register (GINTSTS)

Register	Offset	R/W	Description	Reset Value
GINTSTS	HSUSBD_BA+0x014	R/W	Interrupt Register	0x5400_0020

31	30	29	28	27	26	25	24
WkUpInt	Reserved			LPM_Int	Reserved		
23	22	21	20	19	18	17	16
ResetDet	FetSusp	Reserved		OEPInt	IEPInt	Reserved	
15	14	13	12	11	10	9	8
EOPF	ISOOutDrop	EnumDone	USBRst	USBSusp	ErlySusp	Reserved	
7	6	5	4	3	2	1	0
GOUTNakEff	GINNakEff	Reserved	RxFLvl	Sof	Reserved		

Bits	Description	
[31]	**WkUpInt**	**Resume/Remote Wakeup Detected Interrupt** Wakeup Interrupt during Suspend(L2) or LPM(L1) state. During Suspend(L2): This interrupt is asserted only when Host Initiated Resume is detected on USB. During LPM(L1): This interrupt is asserted for either Host Initiated Resume or Device Initiated Remote Wakeup on USB. **Values:** 0x0 (INACTIVE): Not active 0x1 (ACTIVE): Resume or Remote Wakeup Detected Interrupt
[30:28]	**Reserved**	**Reserved.**
[27]	**LPM_Int**	**LPM Transaction Received Interrupt** This interrupt is asserted when the device receives an LPM transaction and responds with a non-ERRORed response has completed LPM transactions for the programmed number of times (GLPMCFG.RetryCnt) **Values:** 0x0 (INACTIVE): Not Active 0x1 (ACTIVE): LPM Transaction Received Interrupt
[26:24]	**Reserved**	**Reserved.**
[23]	**ResetDet**	**Reset detected Interrupt** In Device mode, this interrupt is asserted when a reset is detected on the USB in partial power-down mode when the device is in Suspend. **Values:** 0x0 (INACTIVE): Not active 0x1 (ACTIVE): Reset detected Interrupt
[22]	**FetSusp**	**Data Fetch Suspended** This interrupt indicates that the core has stopped fetching data. For IN endpoints due to the unavailability of TxFIFO space or Request Queue space. This interrupt is used by the application for an endpoint mismatch algorithm. For example, after detecting an endpoint mismatch, the application:

		Sets a Global non-periodic IN NAK handshake Disables IN endpoints Flushes the FIFO Determines the token sequence from the IN Token Sequence Learning Queue Re-enables the endpoints Clears the Global non-periodic IN NAK handshake If the Global non-periodic IN NAK is cleared, the core has not yet fetched data for the IN endpoint, and the IN token is received. The core generates an 'IN token received when FIFO empty' interrupt. It then sends the host a NAK response. To avoid this scenario, the application can check the GINTSTS.FetSusp interrupt, which ensures that the FIFO is full before clearing a Global NAK handshake. Alternatively, the application can mask the IN token received when FIFO empty interrupt when clearing a Global IN NAK handshake. **Values:** 0x0 (INACTIVE): Not active 0x1 (ACTIVE): Data Fetch Suspended
[21:20]	**Reserved**	**Reserved.**
[19]	**OEPInt**	**OUT Endpoints Interrupt (Read Only)** The controller sets this bit to indicate that an interrupt is pending on one of the OUT endpoints of the core. The application must read the Device All Endpoints Interrupt (DAINT) register to determine the exact number of the OUT endpoint on which the interrupt occurred, and then read the corresponding Device OUT Endpoint-n Interrupt (DOEPINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the corresponding DOEPINTn register to clear this bit. **Values:** 0x0 (INACTIVE): Not active 0x1 (ACTIVE): OUT Endpoints Interrupt
[18]	**IEPInt**	**IN Endpoints Interrupt (Read Only)** The core sets this bit to indicate that an interrupt is pending on one of the IN endpoints of the core. The application must read the Device All Endpoints Interrupt (DAINT) register to determine the exact number of the IN endpoint on Device IN Endpoint-n Interrupt (DIEPINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the corresponding DIEPINTn register to clear this bit. **Values:** 0x0 (INACTIVE): Not active 0x1 (ACTIVE): IN Endpoints Interrupt
[17:16]	**Reserved**	**Reserved.**
[15]	**EOPF**	**End of Periodic Frame Interrupt** Indicates that the period specified in the Periodic Frame Interval field of the Device Configuration register (DCFG.PerFrInt) has been reached in the current microframe. In case of Non-Ignore Frame Number Scatter/Gather (Descriptor DMA) mode, the controller internally handles the following scenarios based on EOPF: **Read Flush:** At the EOPF, the controller checks if there are any pending packets in the FIFO corresponding to the current (micro)Frame. If there are any pending packets, then the controller initiates read flush, due to which the read pointer is updated to the starting location of the next micro-frame packet. If there are no pending packets corresponding to the current (micro)Frame, the controller does not take any action. **Write Flush:** At the EOPF, if the controller is still fetching the current micro-frame data, then the controller stops pushing data into the TXFIFO but keeps fetching the complete packet from the System Memory. After completing the scheduled packet size fetch, the controller updates the Status Quadlet Fields (Transmit Status to BUFFLUSH) and closes the Descriptor. During the descriptor close, the controller initiates write flush, due to which the write pointer is updated to the starting location of the next micro-frame packet. Because the controller stops pushing the packet to the TxFIFO after EOPF, to bring the write pointer to the starting location of the next micro-frame, write flush is done. **Values:**

		0x0 (INACTIVE): Not active 0x1 (ACTIVE): End of Periodic Frame Interrupt
[14]	**ISOOutDrop**	**Isochronous OUT Packet Dropped Interrupt** The controller sets this bit when it fails to write an isochronous OUT packet into the RxFIFO because the RxFIFO does not have enough space to accommodate a maximum packet size packet for the isochronous OUT endpoint. **Values:** 0x0 (INACTIVE): Not active 0x1 (ACTIVE): Isochronous OUT Packet Dropped Interrupt
[13]	**EnumDone**	Enumeration Done The core sets this bit to indicate that speed enumeration is complete. The application must read the Device Status (DSTS) register to obtain the enumerated speed. **Values:** 0x0 (INACTIVE): Not active 0x1 (ACTIVE): Enumeration Done
[12]	**USBRst**	**USB Reset** The controller sets this bit to indicate that a reset is detected on the USB. **Values:** 0x0 (INACTIVE): Not active 0x1 (ACTIVE): USB Reset
[11]	**USBSusp**	**USB Suspend** The controller sets this bit to indicate that a suspend was detected on the USB. The controller enters the Suspended state when there is no activity on the linestate signal for an extended period of time. **Values:** 0x0 (INACTIVE): Not Active 0x1 (ACTIVE): USB Suspend
[10]	**ErlySusp**	**Early Suspend** The controller sets this bit to indicate that an Idle state has been detected on the USB for 3 ms. **Values:** 0x0 (INACTIVE): No Idle state detected 0x1 (ACTIVE): 3ms of Idle state detected
[9:8]	**Reserved**	**Reserved.**
[7]	**GOUTNakEff**	**Global OUT NAK Effective (Read Only)** Indicates that the Set Global OUT NAK bit in the Device Control register (DCTL.SGOUTNak), Set by the application, has taken effect in the core. This bit can be cleared by writing the Clear Global OUT NAK bit in the Device Control register (DCTL.CGOUTNak). **Values:** 0x0 (INACTIVE): Not Active 0x1 (ACTIVE): Global OUT NAK Effective
[6]	**GINNakEff**	**Global IN Non-periodic NAK Effective (Read Only)** Indicates that the Set Global Non-periodic IN NAK bit in the Device Control register (DCTL.SGNPInNak) set by the application, has taken effect in the core. That is, the core has sampled the Global IN NAK bit Set by the application. This bit can be cleared by clearing the Clear Global Non-periodic IN NAK bit in the Device Control register (DCTL.CGNPInNak). This interrupt does not necessarily mean that a NAK handshake is sent out on the USB. The STALL bit takes precedence over the NAK bit. **Values:** 0x0 (INACTIVE): Global Non-periodic IN NAK not active 0x1 (ACTIVE): Set Global Non-periodic IN NAK bit

[5]	**Reserved**	**Reserved.**
[4]	**RxFLvl**	**RxFIFO Non-Empty (Read Only)** Indicates that there is at least one packet pending to be read from the RxFIFO. **Values:** 0x0 (INACTIVE): Rx Fifo is empty 0x1 (ACTIVE): Rx Fifo is not empty
[3]	**Sof**	**Start of Frame** In Device mode, the controller sets this bit to indicate that an SOF token has been received on the USB. The application can read the Device Status register to get the current (micro)Frame number. This interrupt is seen only when the core is operating at either HS or FS. This bit can be set only by the core and the application must write 1 to clear it. **Note:** This register may return 1'b1 if read immediately after power-on reset. If the register bit reads 1'b1 immediately after power-on reset, it does not indicate that an SOF has been received. The read value of this interrupt is valid only after a valid connection between host and device is established. If the bit is set after power on reset the application can clear the bit. **Values:** 0x0 (INTACTIVE): No Start of Frame 0x1 (ACTIVE): Start of Frame
[2:0]	**Reserved**	**Reserved.**

GINTMSK Register (GINTMSK)

Register	Offset	R/W	Description	Reset Value
GINTMSK	HSUSBD_BA+0x018	R/W	Interrupt Mask Register	0x0000_0000

31	30	29	28	27	26	25	24
WkUpIntMsk	Reserved			LPM_IntMsk	Reserved		
23	22	21	20	19	18	17	16
ResetDetMsk	FetSuspMsk	Reserved		OEPIntMsk	IEPIntMsk	EPMisMsk	Reserved
15	14	13	12	11	10	9	8
EOPFMsk	ISOOutDropMsk	EnumDoneMsk	USBRstMsk	USBSuspMsk	ErlySuspMsk	Reserved	
7	6	5	4	3	2	1	0
GOUTNakEffMsk	GINNakEffMsk	Reserved	RxFLvlMsk	SofMsk	OTGIntMsk	Reserved	

Bits	Description	
[31]	**WkUpIntMsk**	**Resume/Remote Wakeup Detected Interrupt Mask** The WakeUp bit is used for LPM state wake up in a way similar to that of wake up in suspend state. **Values:** 0x0 (MASK): Resume or Remote Wakeup Detected Interrupt Mask 0x1 (NOMASK): Unmask Resume Remote Wakeup Detected Interrupt
[30:28]	**Reserved**	**Reserved.**
[27]	**LPM_IntMsk**	**LPM Transaction Received Interrupt** LPM Transaction received interrupt Mask **Values:** 0x0 (MASK): LPM Transaction received interrupt Mask 0x1 (NOMASK): No LPM Transaction received interrupt Mask
[26:24]	**Reserved**	**Reserved.**
[23]	**ResetDetMsk**	**Reset detected Interrupt Mask** **Values:** 0x0 (MASK): Reset detected Interrupt Mask 0x1 (NOMASK): No Reset detected Interrupt Mask
[22]	**FetSuspMsk**	**Data Fetch Suspended Mask** **Values:** 0x0 (MASK): Data Fetch Suspended Mask 0x1 (NOMASK): No Data Fetch Suspended Mask
[21:20]	**Reserved**	Reserved.
[19]	**OEPIntMsk**	OUT Endpoints Interrupt Mask **Values:** 0x0 (MASK): OUT Endpoints Interrupt Mask 0x1 (NOMASK): No OUT Endpoints Interrupt Mask

[18]	**IEPIntMsk**	IN Endpoints Interrupt Mask **Values:** 0x0 (MASK): IN Endpoints Interrupt Mask 0x1 (NOMASK): No IN Endpoints Interrupt Mask
[17]	**EPMisMsk**	Endpoint Mismatch Interrupt Mask **Values:** 0x0 (MASK): Endpoint Mismatch Interrupt Mask 0x1 (NOMASK): No Endpoint Mismatch Interrupt Mask
[16]	**Reserved**	**Reserved.**
[15]	**EOPFMsk**	End of Periodic Frame Interrupt Mask **Values:** 0x0 (MASK): End of Periodic Frame Interrupt Mask 0x1 (NOMASK): No End of Periodic Frame Interrupt Mask
[14]	**ISOOutDropMsk**	Isochronous OUT Packet Dropped Interrupt Mask **Values:** 0x0 (MASK): Isochronous OUT Packet Dropped Interrupt Mask 0x1 (NOMASK): No Isochronous OUT Packet Dropped Interrupt Mask
[13]	**EnumDoneMsk**	Enumeration Done Mask **Values:** 0x0 (MASK): Enumeration Done Mask 0x1 (NOMASK): No Enumeration Done Mask
[12]	**USBRstMsk**	USB Reset Mask **Values:** 0x0 (MASK): USB Reset Mask 0x1 (NOMASK): No USB Reset Mask
[11]	**USBSuspMsk**	USB Suspend Mask **Values:** 0x0 (MASK): USB Suspend Mask 0x1 (NOMASK): No USB Suspend Mask
[10]	**ErlySuspMsk**	Early Suspend Mask **Values:** 0x0 (MASK): Early Suspend Mask 0x1 (NOMASK): No Early Suspend Mask
[9:8]	**Reserved**	**Reserved.**
[7]	**GOUTNakEffMsk**	Global OUT NAK Effective Mask **Values:**

		0x0 (MASK): Global OUT NAK Effective Mask 0x1 (NOMASK): No Global OUT NAK Effective Mask
[6]	**GINNakEffMsk**	Global Non-periodic IN NAK Effective Mask **Values:** 0x0 (MASK): Global Non-periodic IN NAK Effective Mask 0x1 (NOMASK): No Global Non-periodic IN NAK Effective Mask
[5]	**Reserved**	**Reserved.**
[4]	**RxFLvlMsk**	**Receive FIFO Non-Empty Mask** **Values:** 0x0 (MASK): Receive FIFO Non-Empty Mask 0x1 (NOMASK): No Receive FIFO Non-Empty Mask
[3]	**SofMsk**	**Start of Frame Mask** **Values:** 0x0 (MASK): Start of Frame Mask 0x1 (NOMASK): No Start of Frame Mask
[2]	**OTGIntMsk**	**OTG Interrupt Mask** **Values:** 0x0 (MASK): OTG Interrupt Mask 0x1 (NOMASK): No OTG Interrupt Mask
[1:0]	**Reserved**	Reserved.

GRXSTSR Register (GRXSTSR)

Register	Offset	R/W	Description	Reset Value
GRXSTSR	HSUSBD_BA+0x01c	R	Receive Status Debug Read Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							FN
23	22	21	20	19	18	17	16
FN			PktSts				DPID
15	14	13	12	11	10	9	8
DPID	BCnt						
7	6	5	4	3	2	1	0
BCnt				EPNum			

Bits	Description	
[31:25]	**Reserved**	**Reserved.**
[24:21]	**FN**	Frame Number This is the least significant 4 bits of the (micro)Frame number in which the packet is received on the USB. This field is supported only when isochronous OUT endpoints are supported.
[20:17]	**PktSts**	**Packet Status** Indicates the status of the received packet 4'b0001: Global OUT NAK (triggers an interrupt) 4'b0010: OUT data packet received 4'b0011: OUT transfer completed (triggers an interrupt) 4'b0100: SETUP transaction completed (triggers an interrupt) 4'b0110: SETUP data packet received Others: Reserved **Reset:**4'h0 **Values:** 0x1 (OUTNAK): Global OUT NAK (triggers an interrupt) 0x2 (INOUTDPRX): OUT data packet received 0x3 (INOUTTRCOM): IN or OUT transfer completed (triggers an interrupt) 0x4 (DSETUPCOM): SETUP transaction completed (triggers an interrupt) 0x6 (DSETUPRX): SETUP data packet received
[16:15]	**DPID**	**Data PID** Indicates the Data PID of the received OUT data packet 2'b00: DATA0 2'b10: DATA1 2'b01: DATA2 2'b11: MDATA **Reset:** 2'h0 **Values:**

		0x0 (DATA0): DATA0 0x2 (DATA1): DATA1 0x1 (DATA2): DATA2 0x3 (MDATA): MDATA
[14:4]	**BCnt**	**Byte Count** indicates the byte count of the received data packet.
[3:0]	**EPNum**	**Endpoint Number** Indicates the endpoint number to which the current received packet belongs.

GRXSTSP Register (GRXSTSP)

Register	Offset	R/W	Description	Reset Value
GRXSTSP	HSUSBD_BA+0x020	R	Receive Status Read/Pop Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							FN
23	22	21	20	19	18	17	16
FN			PktSts				DPID
15	14	13	12	11	10	9	8
DPID	BCnt						
7	6	5	4	3	2	1	0
BCnt				EPNum			

Bits	Description	
[31:25]	**Reserved**	Reserved.
[24:21]	**FN**	Frame Number This is the least significant 4 bits of the (micro)Frame number in which the packet is received on the USB. This field is supported only when isochronous OUT endpoints are supported.
[20:17]	**PktSts**	**Packet Status** Indicates the status of the received packet 4'b0001: Global OUT NAK (triggers an interrupt) 4'b0010: OUT data packet received 4'b0011: OUT transfer completed (triggers an interrupt) 4'b0100: SETUP transaction completed (triggers an interrupt) 4'b0110: SETUP data packet received Others: Reserved **Reset:**4'h0 **Values:** 0x1 (OUTNAK): Global OUT NAK (triggers an interrupt) 0x2 (INOUTDPRX): OUT data packet received 0x3 (INOUTTRCOM): IN or OUT transfer completed (triggers an interrupt) 0x4 (DSETUPCOM): SETUP transaction completed (triggers an interrupt)
[16:15]	**DPID**	**Data PID** Indicates the Data PID of the received OUT data packet 2'b00: DATA0 2'b10: DATA1 2'b01: DATA2 2'b11: MDATA **Reset:** 2'h0 **Values:** 0x0 (DATA0): DATA0

		0x2 (DATA1): DATA1 0x1 (DATA2): DATA2 0x3 (MDATA): MDATA
[14:4]	**BCnt**	**Byte Count** indicates the byte count of the received data packet.
[3:0]	**EPNum**	**Endpoint Number** Indicates the endpoint number to which the current received packet belongs. **Values:** 0x0 (CHEP0): Channel or EndPoint 0 0x1 (CHEP1): Channel or EndPoint 1 0x2 (CHEP2): Channel or EndPoint 2 0x3 (CHEP3): Channel or EndPoint 3 0x4 (CHEP4): Channel or EndPoint 4 0x5 (CHEP5): Channel or EndPoint 5 0x6 (CHEP6): Channel or EndPoint 6 0x7 (CHEP7): Channel or EndPoint 7 0x8 (CHEP8): Channel or EndPoint 8

GRXFSIZ Register (GRXFSIZ)

Register	Offset	R/W	Description	Reset Value
GRXFSIZ	HSUSBD_BA+0x024	R/W	Receive FIFO Size Register	0x0000_0100

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**RxFDep**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RxFDep							

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8:0]	**RxFDep**	**RxFIFO Depth** This value is in terms of 32-bit words. Minimum value is 16 Maximum value is 32,768 The power-on reset value of this register is specified as the Largest Rx Data FIFO Depth. You can write a new value in this field. Programmed values must not exceed the power-on value.

GNPTXFSIZ Register (GNPTXFSIZ)

Register	Offset	R/W	Description	Reset Value
GNPTXFSIZ	HSUSBD_BA+0x028	R/W	Non-periodic Transmit FIFO Size Register	0x0020_0100

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		**INEPTxF0Dep**					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**INEPTxF0StAddr**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INEPTxF0StAddr							

Bits	Description	
[31:22]	**Reserved**	Reserved.
[21:16]	**INEPTxF0Dep**	**IN Endpoint TxFIFO 0 Depth** This value is in terms of 32-bit words. Minimum value is 16 Maximum value is 32,768 The application can write a new value in this field. The power-on reset value of this field is specified as Largest IN Endpoint FIFO 0 Depth.
[15:9]	**Reserved**	**Reserved.**
[8:0]	**INEPTxF0StAddr**	**IN Endpoint FIFO0 Transmit RAM Start Address** This field contains the memory start address for IN Endpoint Transmit FIFO# 0. Programmed values must not exceed the power-on value.

GLPMCFG Register (GLPMCFG)

Register	Offset	R/W	Description	Reset Value
GLPMCFG	HSUSBD_BA+0x054	R/W	LPM Config Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved			LPM_EnBESL	Reserved			
23	22	21	20	19	18	17	16
LPM_Accept_Ctrl				Reserved			L1ResumeOK
15	14	13	12	11	10	9	8
SlpSts	CoreL1Res		HIRD_Thres				
7	6	5	4	3	2	1	0
EnblSlpM	bRemoteWake	HIRD				AppL1Res	LPMCap

Bits	Description	
[31:29]	**Reserved**	**Reserved.**
[28]	**LPM_EnBESL**	**LPM Enable BESL** This bit enables the BESL feature as defined in LPM Errata 1'b0: The core works as per USB 2.0 Link Power Management Addendum Engineering Change Notice to the USB 2.0 specification as of July 16, 2007 1'b1: The core works as per the LPM Errata **Values:** 0x0 (DISABLED): BESL is disabled 0x1 (ENABLED): BESL is enabled as defined in LPM Errata
[27:24]	**Reserved**	**Reserved.**
[23:20]	**LPM_Accept_Ctrl**	**Device Mode:** LPM Accept Control **LPM_Accept_Ctrl[0]:** The application can use this bit to accept an LPM token even if data is present in the Interrupt endpoint TxFIFO. 1'b0: Reject (NYET) LPM token when data is present in the TxFIFO for Interrupt endpoints. 1'b1: Accept(ACK) LPM token when data is present in the TxFIFO for Interrupt endpoints. **Note:** This bit is applicable only for Dedicated TxFIFO configurations (OTG_EN_DED_TX_FIFO=1). **LPM_Accept_Ctrl[1]:** The application can use this bit to reject an LPM token (NYET) between multiple stages of a single control transfer. 1'b0: Accept(ACK) LPM token during Setup, Data, and Status stage of a control transfer. 1'b1: Reject(NYET) LPM token during Setup, Data, and Status stage of a control transfer. **LPM_Accept_Ctrl[2]:** The application can use this bit to accept an LPM token even if data is present in the ISOC endpoint TxFIFO. 1'b0: Reject (NYET) LPM token when data is present in the TxFIFO for ISOC endpoints. 1'b1: Accept(ACK) LPM token when data is present in the TxFIFO for ISOC endpoints. **Note:** This bit is applicable only for Dedicated TxFIFO configurations (OTG_EN_DED_TX_FIFO=1). **LPM_Accept_Ctrl[3]:** The application can use this bit to accept an LPM token even if data is present in the BULK endpoint TxFIFO. 1'b0: Reject (NYET) LPM token, when data is present in the TxFIFO for Bulk endpoints. 1'b1: Accept(ACK) LPM token, when data is present in the TxFIFO for Bulk endpoints.

		Note: This bit is applicable only for Dedicated TxFIFO configurations (OTG_EN_DED_TX_FIFO=1).
[19:17]	**Reserved**	**Reserved.**
[16]	**L1ResumeOK**	**Sleep State Resume OK (Read Only)** Indicates that the application or host can start resume from Sleep state. This bit is valid in LPM sleep (L1) state. It is set in sleep mode after a delay of 50 micro sec (TL1Residency). The bit is reset when SlpSts is 0. 1'b0: The application/core cannot start resume from Sleep state. 1'b1: The application/core can start resume from Sleep state. **Values:** 0x0 (NOTOK): The application/core cannot start Resume from Sleep state 0x1 (OK): The application/core can start Resume from Sleep state
[15]	**SlpSts**	**Port Sleep Status (Read Only)** This bit is set as long as a Sleep condition is present on the USB bus. The core enters the Sleep state when an ACK response is sent to an LPM transaction and the TL1TokenRetry timer has expired. To stop the PHY clock, the application must set the Port Clock Stop bit, which asserts the PHY Suspend input signal. The application must rely on SlpSts and not ACK in CoreL1Res to confirm transition into sleep. The core comes out of sleep: When there is any activity on the USB line_state When the application writes to the Remote Wakeup Signaling bit in the Device Control register (DCTL.RmtWkUpSig) or when the application resets or soft-disconnects the device. **Values:** 0x0 (CORE_NOT_IN_L1): In Device mode, this bit indicates core is not in L1 0x1 (CORE_IN_L1): In Device mode, the core enters the Sleep state when an ACK response is sent to an LPM transaction
[14:13]	**CoreL1Res**	**LPM Response (Read Only)** The response of the core to LPM transaction received is reflected in these two bits. **Values:** 0x0 (LPMRESP1): ERROR : No handshake response 0x1 (LPMRESP2): STALL response 0x2 (LPMRESP3): NYET response 0x3 (LPMRESP4): ACK response
[12:8]	**HIRD_Thres**	**BESL/HIRD Threshold** EnBESL = 1'b0: The core puts the PHY into deep low power mode in L1 (by core asserting L1SuspendM) when HIRD value is greater than or equal to the value defined in this field HIRD_Thres[3:0] and HIRD_Thres[4] is set to 1b1. EnBESL = 1'b1: The core puts the PHY into deep low power mode in L1 (by core asserting L1SuspendM) when BESL value is greater than or equal to the value defined in this field BESL_Thres[3:0] and BESL_Thres [4] is set to 1'b1. DCTL.DeepSleepBESLReject = 1'b1: In device initiated resume, the core expects the Host to resume service to the device within the BESL value corresponding to L1 exit time specified in HIRD_Thres[3:0]. The Device sends a NYET response when the received HIRD in LPM token is greater than HIRD threshold.
[7]	**EnblSlpM**	**Enable utmi_sleep_n** The application uses this bit to control utmi_sleep_n assertion to the PHY in the L1 state. 1'b0: utmi_sleep_n assertion from the core is not transferred to the external PHY. 1'b1: utmi_sleep_n assertion from the core is transferred to the external PHY when utmi_l1_suspend_n cannot be asserted. **Values:** 0x0 (DISABLED): utmi_sleep_n assertion from the core is not transferred to the external PHY 0x1 (ENABLED): utmi_sleep_n assertion from the core is transferred to the external PHY when utmi_l1_suspend_n cannot be asserted

[6]	**bRemoteWake**	**RemoteWakeEnable (Read Only)** This field is read only. It is updated with the Received LPM Token bRemoteWake bmAttribute when an ACK/NYET/STALL response is sent to an LPM transaction. **Values:** 0x0 (DISABLED): Remote Wakeup is disabled 0x1 (ENABLED): In device mode, this field takes the value of remote wake up
[5:2]	**HIRD**	**Host-Initiated Resume Duration (Read Only)** **EnBESL = 1'b0.** Host Initiated Resume Duration. This field is read only and is updated with the Received LPM Token HIRD bmAttribute when an ACK/NYET/STALL response is sent to an LPM transaction. If HIRD[3:0], 4'b0000, THIRD(us) = 50. 4'b0001, THIRD(us) = 125. 4'b0010, THIRD(us) = 200. 4'b0011, THIRD(us) = 275. 4'b0100, THIRD(us) = 350. 4'b0101, THIRD(us) = 425. 4'b0110, THIRD(us) = 500. 4'b0111, THIRD(us) = 575. 4'b1000, THIRD(us) = 650. 4'b1001, THIRD(us) = 725. 4'b1010, THIRD(us) = 800. 4'b1011, THIRD(us) = 875. 4'b1100, THIRD(us) = 950. 4'b1101, THIRD(us) = 1025. 4'b1110, THIRD(us) = 1100. 4'b1111, THIRD(us) = 1175. **EnBESL = 1'b1.** Best Effort Service Latency (BESL). This field is updated with the Received LPM Token BESL bmAttribute when an ACK/NYET/STALL response is sent to an LPM transaction. If BESL[3:0], 4'b0000, TBESL(us) = 125. 4'b0001, TBESL(us) = 150. 4'b0010, TBESL(us) = 200. 4'b0011, TBESL(us) = 300. 4'b0100, TBESL(us) = 400. 4'b0101, TBESL(us) = 500. 4'b0110, TBESL(us) = 1000. 4'b0111, TBESL(us) = 2000. 4'b1000, TBESL(us) = 3000. 4'b1001, TBESL(us) = 4000. 4'b1010, TBESL(us) = 5000. 4'b1011, TBESL(us) = 6000. 4'b1100, TBESL(us) = 7000. 4'b1101, TBESL(us) = 8000. 4'b1110, TBESL(us) = 9000. 4'b1111, TBESL(us) = 10000.
[1]	**AppL1Res**	**LPM response programmed by application** Handshake response to LPM token pre-programmed by device application software. The response depends on GLPMCFG.LPMCap. If GLPMCFG.LPMCap is 1'b0, the core operates as a non-LPM-capable Device and does not respond to any LPM transactions. If GLPMCFG.LPMCap is 1'b1, the core responds as follows:

		1: ACK Even though an ACK is pre-programmed, the core responds with an ACK only on a successful LPM transaction. The LPM transaction is successful if: There are no PID/CRC5 errors in both the EXT token and the LPM token (else ERROR) A valid bLinkState = 0001B (L1) is received in the LPM transaction (else STALL). No data is pending in the Transmit queue (else NYET) 0: NYET The pre-programmed software bit is overridden for response to LPM token when: The received bLinkState is not L1 (STALL response) An error is detected in either of the LPM token packets due to corruption (ERROR response). **Values:** 0x0 (NYET_RESP): The core responds with a NYET when an error is detected in either of the LPM token packets due to corruption 0x1 (ACK_RESP): The core responds with an ACK only on a successful LPM transaction
[0]	**LPMCap**	**LPM-Capable** The application uses this bit to control the controller LPM capabilities. If the core operates as a non-LPM-capable device, it cannot respond to any LPM transactions. If GLPMCFG.LPMCap is 1'b0, the software must not set any of the remaining fields in the GLPMCFG register and these fields should hold their Reset values. 1'b0: LPM capability is not enabled. 1'b1: LPM capability is enabled. **Values:** 0x0 (DISABLED): LPM capability is not enabled 0x1 (ENABLED): LPM capability is enabled

GDFIFOCFG Register (GDFIFOCFG)

Register	Offset	R/W	Description	Reset Value
GDFIFOCFG	HSUSBD_BA+0x05c	R/W	Global DFIFO Configuration Register	0x0BB8_0C00

31	30	29	28	27	26	25	24
EPInfoBaseAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
EPInfoBaseAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
GDFIFOCfg							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GDFIFOCfg							

Bits	Description	
[31:16]	**EPInfoBaseAddr**	**EPInfoBaseAddr** This field provides the start address of the EP info controller. **Value After Reset:** 0xbb8
[15:0]	**GDFIFOCfg**	**GDFIFOCfg** This field is for dynamic programming of the DFIFO Size. This value takes effect only when the application programs a non zero value to this register. The value programmed must conform to the guidelines described in 'FIFO RAM Allocation'. The core does not have any corrective logic if the FIFO sizes are programmed incorrectly. **Value After Reset:** 0xc00

GREFCLK Register (GREFCLK)

Register	Offset	R/W	Description	Reset Value
GREFCLK	HSUSBD_BA+0x064	R/W	ref_clk Control Register	0x0000_0000

31	30	29	28	27	26	25	24
REFCLKPER							
23	22	21	20	19	18	17	16
REFCLKPER							
15	14	13	12	11	10	9	8
REFCLKPER	RefclkMode	Reserved				SOF_CNT_WKUP_ALERT	
7	6	5	4	3	2	1	0
SOF_CNT_WKUP_ALERT							

Bits	Description	
[31:15]	**REFCLKPER**	**This bit indicates the period of ref_clk in terms of pico seconds** **Note:** The default value of this field is 'd0. The period of ref_clk should be an integer multiple of 125us. The minimum frequency supported is 12 MHz. Other supported frequencies are 16, 17, 19.2, 20, 24, 30, and 40 MHz.
[14]	**RefclkMode**	**This bit is used to enable or disable ref_clk mode of operation** **Note:** The default value of this field is 'd0. When this field is disabled, DCTL.ServInt cannot be set to 1. **Values:** 0x1 (ENABLE): Controller uses ref_clk to run internal micro-frame timers 0x0 (DISABLE): Controller uses phy_clk to run internal micro-frame timers
[13:10]	**Reserved**	**Reserved.**
[9:0]	**SOF_CNT_WKUP_ALERT**	**This bit indicates the number of SOF's after which the controller should generate an interrupt if the device had been in L1 state until that period** The interrupt is used by software to initiate remote wakeup in the controller in order to sync to the uF number in the host.

GINTMSK2 Register (GINTMSK2)

Register	Offset	R/W	Description	Reset Value
GINTMSK2	HSUSBD_BA+0x068	R/W	Interrupt Mask Register 2	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**WkUpAlertIntMsk**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**WkUpAlertIntMsk**	**Mode: Device** Remote WakeUp Alert Interrupt Mask This interrupt is used to alert the application to initiate Remote WakeUp sequence. **Values**: 0x0 (MASK): Mask Remote WakeUp Alert Interrupt 0x1 (NOMASK): Unmask Remote WakeUp Alert Interrupt

GINTSTS2 Register (GINTSTS2)

Register	Offset	R/W	Description	Reset Value
GINTSTS2	HSUSBD_BA+0x06c	R/W	Interrupt Register 2	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**WkUpAlertInt**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**WkUpAlertInt**	**Mode: Device** Remote WakeUp Alert Interrupt Mask This interrupt is used to alert the application to initiate Remote WakeUp sequence. Values: • 0x0 (INACTIVE): Not Active • 0x1 (ACTIVE): Remote WakeUp Alert Interrupt detected

DIEPTXF1 Register (DIEPTXF1)

Register	Offset	R/W	Description	Reset Value
DIEPTXF1	HSUSBD_BA+0x104	R/W	Device IN Endpoint Transmit FIFO Size Register 1	0x0100_0120

31	30	29	28	27	26	25	24
Reserved							**INEPnTxFDep**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
INEPnTxFDep							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**INEPnTxFStAddr**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INEPnTxFStAddr							

Bits	Description	
[31:25]	**Reserved**	Reserved.
[24:16]	**INEPnTxFDep**	**IN Endpoint TxFIFO Depth** This value is in terms of 32-bit words. Minimum value is 16 Maximum value is 32,768 The power-on reset value of this register is specified as the Largest IN Endpoint FIFO number Depth. Programmed values must not exceed the power-on value.
[15:9]	**Reserved**	**Reserved.**
[8:0]	**INEPnTxFStAddr**	**IN Endpoint FIFOn Transmit RAM Start Address** This field contains the memory start address for IN endpoint Transmit FIFO. The power-on reset value of this register is specified as the Largest Rx Data FIFO Depth.

DIEPTXF2 Register (DIEPTXF2)

Register	Offset	R/W	Description	Reset Value
DIEPTXF2	HSUSBD_BA+0x108	R/W	Device IN Endpoint Transmit FIFO Size Register 2	0x0100_0220

31	30	29	28	27	26	25	24
Reserved							**INEPnTxFDep**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
INEPnTxFDep							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**INEPnTxFStAddr**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INEPnTxFStAddr							

Bits	Description	
[31:25]	**Reserved**	Reserved.
[24:16]	**INEPnTxFDep**	**IN Endpoint TxFIFO Depth** This value is in terms of 32-bit words. Minimum value is 16 Maximum value is 32,768 The power-on reset value of this register is specified as the Largest IN Endpoint FIFO number Depth. Programmed values must not exceed the power-on value.
[15:10]	**Reserved**	**Reserved.**
[9:0]	**INEPnTxFStAddr**	**IN Endpoint FIFOn Transmit RAM Start Address** This field contains the memory start address for IN endpoint Transmit FIFO. The power-on reset value of this register is specified as the Largest Rx Data FIFO Depth.

DIEPTXF3 Register (DIEPTXF3)

Register	Offset	R/W	Description	Reset Value
DIEPTXF3	HSUSBD_BA+0x10c	R/W	Device IN Endpoint Transmit FIFO Size Register 3	0x0100_0320

31	30	29	28	27	26	25	24
Reserved							**INEPnTxFDep**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
INEPnTxFDep							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**INEPnTxFStAddr**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INEPnTxFStAddr							

Bits	Description	
[31:25]	**Reserved**	Reserved.
[24:16]	**INEPnTxFDep**	**IN Endpoint TxFIFO Depth** This value is in terms of 32-bit words. Minimum value is 16 Maximum value is 32,768 The power-on reset value of this register is specified as the Largest IN Endpoint FIFO number Depth. Programmed values must not exceed the power-on value.
[15:10]		**Reserved.**
[9:0]	**INEPnTxFStAddr**	**IN Endpoint FIFOn Transmit RAM Start Address** This field contains the memory start address for IN endpoint Transmit FIFO. The power-on reset value of this register is specified as the Largest Rx Data FIFO Depth. **Value After Reset:** 0x320

DIEPTXF4 Register (DIEPTXF4)

Register	Offset	R/W	Description	Reset Value
DIEPTXF4	HSUSBD_BA+0x110	R/W	Device IN Endpoint Transmit FIFO Size Register 4	0x0100_0420

31	30	29	28	27	26	25	24
Reserved							INEPnTxFDep
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
INEPnTxFDep							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					INEPnTxFStAddr		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INEPnTxFStAddr							

Bits	Description	
[31:25]	**Reserved**	Reserved.
[24:16]	**INEPnTxFDep**	**IN Endpoint TxFIFO Depth** This value is in terms of 32-bit words. Minimum value is 16 Maximum value is 32,768 The power-on reset value of this register is specified as the Largest IN Endpoint FIFO number Depth. Programmed values must not exceed the power-on value
[15:11]	**Reserved**	**Reserved.**
[10:0]	**INEPnTxFStAddr**	**IN Endpoint FIFOn Transmit RAM Start Address** This field contains the memory start address for IN endpoint Transmit FIFO. The power-on reset value of this register is specified as the Largest Rx Data FIFO Depth.

DIEPTXF5 Register (DIEPTXF5)

Register	Offset	R/W	Description	Reset Value
DIEPTXF5	HSUSBD_BA+0x114	R/W	Device IN Endpoint Transmit FIFO Size Register 5	0x0100_0520

31	30	29	28	27	26	25	24
Reserved							INEPnTxFDep
23	22	21	20	19	18	17	16
INEPnTxFDep							
15	14	13	12	11	10	9	8
Reserved					INEPnTxFStAddr		
7	6	5	4	3	2	1	0
INEPnTxFStAddr							

Bits	Description	
[31:25]	**Reserved**	Reserved.
[24:16]	**INEPnTxFDep**	**IN Endpoint TxFIFO Depth** This value is in terms of 32-bit words. Minimum value is 16 Maximum value is 32,768 The power-on reset value of this register is specified as the Largest IN Endpoint FIFO number Depth. Programmed values must not exceed the power-on value.
[15:11]	**Reserved**	**Reserved.**
[10:0]	**INEPnTxFStAddr**	**IN Endpoint FIFOn Transmit RAM Start Address** This field contains the memory start address for IN endpoint Transmit FIFO. The power-on reset value of this register is specified as the Largest Rx Data FIFO Depth.

DIEPTXF6 Register (DIEPTXF6)

Register	Offset	R/W	Description	Reset Value
DIEPTXF6	HSUSBD_BA+0x118	R/W	Device IN Endpoint Transmit FIFO Size Register 6	0x0100_0620

31	30	29	28	27	26	25	24
Reserved							INEPnTxFDep
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
INEPnTxFDep							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					INEPnTxFStAddr		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INEPnTxFStAddr							

Bits	Description	
[31:25]	**Reserved**	**Reserved.**
[24:16]	**INEPnTxFDep**	**IN Endpoint TxFIFO Depth** This value is in terms of 32-bit words. Minimum value is 16 Maximum value is 32,768 The power-on reset value of this register is specified as the Largest IN Endpoint FIFO number Depth. Programmed values must not exceed the power-on value.
[15:11]	**Reserved**	**Reserved.**
[10:0]	**INEPnTxFStAddr**	**IN Endpoint FIFOn Transmit RAM Start Address** This field contains the memory start address for IN endpoint Transmit FIFO. The power-on reset value of this register is specified as the Largest Rx Data FIFO Depth.

DIEPTXF7 Register (DIEPTXF7)

Register	Offset	R/W	Description	Reset Value
DIEPTXF7	HSUSBD_BA+0x11c	R/W	Device IN Endpoint Transmit FIFO Size Register 7	0x0100_0720

31	30	29	28	27	26	25	24
Reserved							INEPnTxFDep
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
INEPnTxFDep							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					INEPnTxFStAddr		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INEPnTxFStAddr							

Bits	Description	
[31:25]	**Reserved**	**Reserved.**
[24:16]	**INEPnTxFDep**	**IN Endpoint TxFIFO Depth** This value is in terms of 32-bit words. Minimum value is 16 Maximum value is 32,768 The power-on reset value of this register is specified as the Largest IN Endpoint FIFO number Depth. Programmed values must not exceed the power-on value.
[15:11]	**Reserved**	**Reserved.**
[10:0]	**INEPnTxFStAddr**	**IN Endpoint FIFOn Transmit RAM Start Address** This field contains the memory start address for IN endpoint Transmit FIFO. The power-on reset value of this register is specified as the Largest Rx Data FIFO Depth.

DIEPTXF8 Register (DIEPTXF8)

Register	Offset	R/W	Description	Reset Value
DIEPTXF8	HSUSBD_BA+0x120	R/W	Device IN Endpoint Transmit FIFO Size Register 8	0x0100_0820

31	30	29	28	27	26	25	24
Reserved							INEPnTxFDep
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
INEPnTxFDep							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				INEPnTxFStAddr			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INEPnTxFStAddr							

Bits	Description	
[31:25]	**Reserved**	**Reserved.**
[24:16]	**INEPnTxFDep**	**IN Endpoint TxFIFO Depth** This value is in terms of 32-bit words. Minimum value is 16 Maximum value is 32,768 The power-on reset value of this register is specified as the Largest IN Endpoint FIFO number Depth. Programmed values must not exceed the power-on value.
[15:12]	**Reserved**	**Reserved.**
[11:0]	**INEPnTxFStAddr**	**IN Endpoint FIFOn Transmit RAM Start Address** This field contains the memory start address for IN endpoint Transmit FIFO. The power-on reset value of this register is specified as the Largest Rx Data FIFO Depth.

DCFG Register (DCFG)

Register	Offset	R/W	Description	Reset Value
DCFG	HSUSBD_BA+0x800	R/W	Device Configuration Register	0x0812_0000

31	30	29	28	27	26	25	24
ResValid						PerSchIntvl	
23	22	21	20	19	18	17	16
DescDMA	Reserved					ipgisocSupt	Reserved
15	14	13	12	11	10	9	8
ErraticIntMsk	XCVRDLY	EnDevOutNak	PerFrInt		DevAddr		
7	6	5	4	3	2	1	0
DevAddr				Reserved	NZStsOUTHShk	DevSpd	

Bits	Description	
[31:26]	**ResValid**	**Resume Validation Period** This field is effective only when DCFG.Ena32 kHzSusp is set. It controls the resume period when the core resumes from suspend. The core counts for ResValid number of clock cycles to detect a valid resume when this bit is set
[25:24]	**PerSchIntvl**	**Periodic Scheduling Interval** PerSchIntvl must be programmed for Scatter/Gather DMA mode. This field specifies the amount of time the Internal DMA engine must allocate for fetching periodic IN endpoint data. Based on the number of periodic endpoints, this value must be specified as 25,50 or 75% of (micro)Frame. When any periodic endpoints are active, the internal DMA engine allocates the specified amount of time in fetching periodic IN endpoint data . When no periodic endpoints are active, Then the internal DMA engine services non-periodic endpoints, ignoring this field. After the specified time within a (micro)Frame, the DMA switches to fetching for non-periodic endpoints. 2'b00: 25% of (micro)Frame. 2'b01: 50% of (micro)Frame. 2'b10: 75% of (micro)Frame. 2'b11: Reserved. Reset: 2'b00 **Values:** 0x0 (MF25): 25% of (micro)Frame 0x1 (MF50): 50% of (micro)Frame 0x2 (MF75): 75% of (micro)Frame 0x3 (RESERVED): Reserved
[23]	**DescDMA**	**Enable Scatter/gather DMA in device mode** The application can Set this bit during initialization to enable the Scatter/Gather DMA operation. **Note:** This bit must be modified only once after a reset. The following combinations are available for programming: GAHBCFG.DMAEn=0,DCFG.DescDMA=0 => Invalid. GAHBCFG.DMAEn=0,DCFG.DescDMA=1 => Invalid.

		GAHBCFG.DMAEn=1,DCFG.DescDMA=0 => Invalid. GAHBCFG.DMAEn=1,DCFG.DescDMA=1 => Scatter/Gather DMA mode. **Values:** 0x0 (DISABLED): Disable Scatter/Gather DMA 0x1 (ENABLED): Enable Scatter/Gather DMA
[22:18]	**Reserved**	**Reserved.**
[17]	**ipgisocSupt**	**Worst-Case Inter-Packet Gap ISOC OUT Support** This bit indicates that the controller supports the worst-case scenario of Rx followed by Rx Inter Packet Gap (IPG) (32 bit times) as per the UTMI Specification for any token following an ISOC OUT token. Without this support, when any token follows an ISOC OUT token with the worst-case IPG, the controller will not detect the followed token. The worst-case IPG of the controller without this support depends on the AHB and PHY Clock frequency. **Values:** 0x0 (DISABLED): Worst-Case Inter-Packet Gap ISOC OUT Support is disabled 0x1 (ENABLED): Worst-Case Inter-Packet Gap ISOC OUT Support is enabled
[16]	**Reserved**	**Reserved.**
[15]	**ErraticIntMsk**	**Erratic Error Interrupt Mask** **Values:** 0x0 (NOMASK): Early suspend interrupt is generated on erratic error 0x1 (MASK): Mask early suspend interrupt on erratic error
[14]	**XCVRDLY**	**XCVRDLY** Enables or disables delay between xcvr_sel and txvalid during device chirp **Values:** 0x0 (DISABLE): No delay between xcvr_sel and txvalid during Device chirp 0x1 (ENABLE): Enable delay between xcvr_sel and txvalid during Device chirp
[13]	**EnDevOutNak**	**Enable Device OUT NAK** This bit enables setting NAK for Bulk OUT endpoints after the transfer is completed for Device mode Descriptor DMA 1'b0 : The core does not set NAK after Bulk OUT transfer complete 1'b1 : The core sets NAK after Bulk OUT transfer complete This bit is one time programmable after reset like any other DCFG register bits. **Values:** 0x0 (DISABLED): The core does not set NAK after Bulk OUT transfer complete 0x1 (ENABLED): The core sets NAK after Bulk OUT transfer complete
[12:11]	**PerFrInt**	**Periodic Frame Interval** Indicates the time within a (micro)Frame at which the application must be notified using the End Of Periodic Frame Interrupt. This can be used to determine If all the isochronous traffic for that (micro)Frame is complete. 2'b00: 80% of the (micro)Frame interval 2'b01: 85% of the (micro)Frame interval 2'b10: 90% of the (micro)Frame interval 2'b11: 95% of the (micro)Frame interval **Values:** 0x0 (EOPF80): 80% of the (micro)Frame interval 0x1 (EOPF85): 85% of the (micro)Frame interval 0x2 (EOPF90): 90% of the (micro)Frame interval 0x3 (EOPF95): 95% of the (micro)Frame interval

[10:4]	**DevAddr**	**Device Address** The application must program this field after every SetAddress control command.
[3]	**Reserved**	Reserved.
[2]	**NZStsOUTHShk**	**Non-Zero-Length Status OUT Handshake** The application can use this field to select the handshake the core sends on receiving a nonzero-length data packet during the OUT transaction of a control transfer's Status stage. 1'b1: Send a STALL handshake on a nonzero-length status OUT transaction and do not send the received OUT packet to the application. 1'b0: Send the received OUT packet to the application (zerolength or nonzero-length) and send a handshake based on the NAK and STALL bits for the endpoint in the Device Endpoint Control register. **Values:** 0x0 (SENDOUT): Send the received OUT packet to the application (zero-length or non-zero length) and send a handshake based on NAK and STALL bits for the endpoint in the Devce Endpoint Control Register 0x1 (SENDSTALL): Send a STALL handshake on a nonzero-length status OUT transaction and do not send the received OUT packet to the application
[1:0]	**DevSpd**	**Device Speed** Indicates the speed at which the application requires the core to enumerate, or the maximum speed the application can support. However, the actual bus speed is determined only after the connect sequence is completed, and is based on the speed of the USB host to which the core is connected. **Values:** 0x0 (USBHS20): High speed USB 2.0 0x1 (USBFS20): Full speed USB 2.0 0x2 (Reserved0): Reserved 0x3 (Reserved1): Reserved

DCTL Register (DCTL)

Register	Offset	R/W	Description	Reset Value
DCTL	HSUSBD_BA+0x804	R/W	Device Control Register	0x0000_0002

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
R5				ServInt	DeepSleepBESLReject	EnContOnBNA	NakOnBble
15	14	13	12	11	10	9	8
IgnrFrmNum	GMC		Reserved	PWROnPrgDone	CGOUTNak	SGOUTNak	CGNPInNak
7	6	5	4	3	2	1	0
SGNPInNak	TstCtl			GOUTNakSts	GNPINNakSts	SftDiscon	RmtWkUpSig

Bits	Description	
[31:20]	**Reserved**	**Reserved.**
[19]	**ServInt**	**Service Interval based scheduling for Isochronous IN Endpoints** **Note:** This bit is applicable only in device mode and when Scatter/Gather DMA mode is used. This feature should not be enabled along with DCTL.IgnrFrmNum. When this bit is enabled, the frame number field in the ISOC IN descriptor structure is interpreted as the last frame of the service interval. In Scatter/Gather DMA mode, if this bit is enabled, the pending packets are flushed by the controller at the last frame of the service interval. **Values:** 0x0 (DISABLED): The controller behavior depends on DCTL.IgnrFrmNum field. 0x1 (ENABLED): The controller can transmit the packets in any frame of the service interval.
[18]	**DeepSleepBESLReject**	**DeepSleepBESLReject** 1: Deep Sleep BESL Reject feature is enabled 0: Deep Sleep BESL Reject feature is disabled Core rejects LPM request with HIRD value greater than HIRD threshold programmed. NYET response is sent for LPM tokens with HIRD value greater than HIRD threshold. By default, the Deep Sleep BESL Reject feature is disabled. **Values:** 0x0 (DISABLED): Deep Sleep BESL Reject feature is disabled 0x1 (ENABLED): Deep Sleep BESL Reject feature is enabled
[17]	**EnContOnBNA**	**Enable Continue on BNA** This bit enables the core to continue on BNA for Bulk OUT endpoints. With this feature enabled, when a Bulk OUT or INTR OUT endpoint receives a BNA interrupt the core starts processing the descriptor that caused the BNA interrupt after the endpoint re-enables the endpoint. 1'b0: After receiving BNA interrupt,the core disables the endpoint. When the endpoint is re-enabled by the application, the core starts processing from the DOEPDMA descriptor. 1'b1: After receiving BNA interrupt, the core disables the endpoint. When the endpoint is re-enabled by the application, the core starts processing from the descriptor that received the BNA interrupt. It is a one-time programmable after reset bit like any other DCTL register bits. **Values:**

		0x0 (DISABLED): Core disables the endpoint after receiving BNA interrupt. When application re-enables the endpoint, core starts processing from the DOEPDMA descriptor 0x1 (ENABLED): Core disables the endpoint after receiving BNA interrupt. When application re-enables the endpoint, core starts processing from the descriptor that received the BNA interrupt.
[16]	**NakOnBble**	**NAK on Babble Error** Set NAK automatically on babble (NakOnBble). The core sets NAK automatically for the endpoint on which babble is received. **Values:** 0x0 (DISABLED): Disable NAK on Babble Error 0x1 (ENABLED): NAK on Babble Error
[15]	**IgnrFrmNum**	**Ignore Frame Number Feature for Isochronous Endpoints** **Note:** Do not program IgnrFrmNum bit to 1'b1 when the core is operating in threshold mode. **Note:** When Scatter/Gather DMA mode is enabled this feature is not applicable to High Speed, High bandwidth transfers. When this bit is enabled, there must be only one packet per descriptor. 0: The core transmits the packets only in the frame number in which they are intended to be transmitted. 1: The core ignores the frame number, sending packets immediately as the packets are ready. In Scatter/Gather DMA mode, if this bit is enabled, the packets are not flushed when a ISOC IN token is received for an elapsed frame. **Values:** 0x0 (DISABLED): The core transmits the packets only in the frame number in which they are intended to be transmitted. 0x1 (ENABLED): The core ignores the frame number, sending packets immediately as the packets are ready.
[14:13]	**GMC**	**Global Multi Count** GMC must be programmed only once after initialization. Applicable only for Scatter/Gather DMA mode. This indicates the number of packets to be serviced for that end point before moving to the next end point. It is only for non-periodic endpoints. 2'b00: Invalid. 2'b01: 1 packet. 2'b10: 2 packets. 2'b11: 3 packets. **Values:** 0x0 (NOTVALID): Invalid 0x1 (ONEPACKET): 1 packet 0x2 (TWOPACKET): 2 packets 0x3 (THREEPACKET): 3 packets
[12]	**Reserved**	**Reserved.**
[11]	**PWROnPrgDone**	**Power-On Programming Done** The application uses this bit to indicate that register programming is completed after a wake-up from Power-down mode. **Values:** 0x0 (NOTDONE): Power-On Programming not done 0x1 (DONE): Power-On Programming Done
[10]	**CGOUTNak**	**Clear Global OUT NAK** A write to this field clears the Global OUT NAK. **Values:** 0x0 (DISABLED): Disable Clear Global OUT NAK

		0x1 (ENABLED): Clear Global OUT NAK
[9]	**SGOUTNak**	**Set Global OUT NAK** A write to this field sets the Global OUT NAK. The application uses this bit to send a NAK handshake on all OUT endpoints. The application must set the this bit only after making sure that the Global OUT NAK Effective bit in the Core Interrupt Register (GINTSTS.GOUTNakEff) is cleared. **Values:** 0x0 (DISABLED): Disable Global OUT NAK 0x1 (ENABLED): Set Global OUT NAK
[8]	**CGNPInNak**	**Clear Global Non-periodic IN NAK** A write to this field clears the Global Non-periodic IN NAK. **Values:** 0x0 (DISABLE): Disable Global Non-periodic IN NAK 0x1 (ENABLE): Clear Global Non-periodic IN NAK
[7]	**SGNPInNak**	**Set Global Non-periodic IN NAK** A write to this field sets the Global Non-periodic IN NAK.The application uses this bit to send a NAK handshake on all non-periodic IN endpoints. The application must Set this bit only after making sure that the Global IN NAK Effective bit in the Core Interrupt Register (GINTSTS.GINNakEff) is cleared **Values:** 0x0 (DISABLE): Disable Global Non-periodic IN NAK 0x1 (ENABLE): Set Global Non-periodic IN NAK
[6:4]	**TstCtl**	**Test Control** 3'b000: Test mode disabled 3'b001: Test_J mode 3'b010: Test_K mode 3'b011: Test_SE0_NAK mode 3'b100: Test_Packet mode 3'b101: Test_Force_Enable Others: Reserved **Values:** 0x0 (DISABLED): Test mode disabled 0x1 (TESTJ): Test_J mode 0x2 (TESTK): Test_K mode 0x3 (TESTSN): Test_SE0_NAK mode 0x4 (TESTPM): Test_Packet mode 0x5 (TESTFE): Test_force_Enable
[3]	**GOUTNakSts**	**Global OUT NAK Status (Read Only)** 1'b0: A handshake is sent based on the FIFO Status and the NAK and STALL bit settings. 1'b1: No data is written to the RxFIFO, irrespective of space availability. Sends a NAK handshake on all packets, except on SETUP transactions. All isochronous OUT packets are dropped. **Values:** 0x0 (INACTIVE): A handshake is sent based on the FIFO Status and the NAK and STALL bit settings. 0x1 (ACTIVE): No data is written to the RxFIFO, irrespective of space availability. Sends a NAK handshake on all packets, except on SETUP transactions. All isochronous OUT packets are dropped.
[2]	**GNPINNakSts**	**Global Non-periodic IN NAK Status (Read Only)** 1'b0: A handshake is sent out based on the data availability in the transmit FIFO.

		1'b1: A NAK handshake is sent out on all non-periodic IN endpoints, irrespective of the data availability in the transmit FIFO. **Values:** 0x0 (INACTIVE): A handshake is sent out based on the data availability in the transmit FIFO 0x1 (ACTIVE): A NAK handshake is sent out on all non-periodic IN endpoints, irrespective of the data availability in the transmit FIFO.
[1]	**SftDiscon**	**Soft Disconnect** The application uses this bit to signal the controller to do a soft disconnect. As long as this bit is set, the host does not see that the device is connected, and the device does not receive signals on the USB. The core stays in the disconnected state until the application clears this bit. 1'b0: Normal operation. When this bit is cleared after a soft disconnect, the core drives the phy_opmode_o signal on the UTMI+ to 2'b00, which generates a device connect event to the USB host. When the device is reconnected, the USB host restarts device enumeration. 1'b1: The core drives the phy_opmode_o signal on the UTMI+ to 2'b01, which generates a device disconnect event to the USB host. **Note:** This bit can be also used for ULPI/FS Serial interfaces. This bit is not impacted by a soft reset. **Values:** 0x0 (NODISCONNECT): The core drives the phy_opmode_o signal on the UTMI+ to 2'b00, which generates a device connect event to the USB host 0x1 (DISCONNECT): The core drives the phy_opmode_o signal on the UTMI+ to 2'b01, which generates a device disconnect event to the USB host
[0]	**RmtWkUpSig**	**Remote Wakeup Signaling** When the application sets this bit, the core initiates remote signaling to wake up the USB host. The application must Set this bit to instruct the core to exit the Suspend state. As specified in the USB 2.0 specification, the application must clear this bit 1-15 ms after setting it. If LPM is enabled and the core is in the L1 (Sleep) state, when the application sets this bit, the core initiates L1 remote signaling to wake up the USB host. The application must set this bit to instruct the core to exit the Sleep state. As specified in the LPM specification, the hardware automatically clears this bit 50 microseconds (TL1DevDrvResume) after being set by the application. The application must not set this bit when GLPMCFG bRemoteWake from the previous LPM transaction is zero. **Values:** 0x0 (DISABLEDRMWKUP): Core does not send Remote Wakeup Signaling 0x1 (ENABLERMWKUP): Core sends Remote Wakeup Signaling

DSTS Register (DSTS)

Register	Offset	R/W	Description	Reset Value
DSTS	HSUSBD_BA+0x808	R	Device Status Register	0x0000_0002

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
DevLnSts		SOFFN					
15	14	13	12	11	10	9	8
SOFFN							
7	6	5	4	3	2	1	0
Reserved				ErrticErr	EnumSpd		SuspSts

Bits	Description	
[31:24]	**Reserved**	**Reserved.**
[23:22]	**DevLnSts**	**Device Line Status** Indicates the current logic level USB data lines DevLnSts[1]: Logic level of D+ DevLnSts[0]: Logic level of D-
[21:8]	**SOFFN**	**Frame or Microframe Number of the Received SOF** When the core is operating at high speed, this field contains a microframe number. When the core is operating at full or low speed, this field contains a Frame number. **Note:** This register may return a non-zero value if read immediately after power-on reset. In case the register bit reads non-zero immediately after power-on reset, it does not indicate that SOF has been received from the host. The read value of this interrupt is valid only after a valid connection between host and device is established.
[7:4]	**Reserved**	**Reserved.**
[3]	**ErrticErr**	**Erratic Error** The core sets this bit to report any erratic errors (phy_rxvalid_i/phy_rxvldh_i or phy_rxactive_i is asserted for at least 2 ms, due to PHY error). Due to erratic errors, the core goes into Suspended state and an interrupt is generated to the application with Early Suspend bit of the Core Interrupt register (GINTSTS.ErlySusp). If the early suspend is asserted due to an erratic error, the application can only perform a soft disconnect recover. **Values:** 0x0 (INACTIVE): No Erratic Error 0x1 (ACTIVE): Erratic Error
[2:1]	**EnumSpd**	**Enumerated Speed** Indicates the speed at which the controller has come up after speed detection through a connect or reset sequence. 2'b00: High speed 2'b01: Full speed **Values:** 0x0 (HS3060): High speed 0x1 (FS3060): Full speed

		0x2 (Reserved0): Reserved 0x3 (Reserved1): Reserved
[0]	**SuspSts**	**Suspend Status** In Device mode, this bit is set as long as a Suspend condition is detected on the USB. The core enters the Suspend state when there is no activity on the phy_line_state_i signal for an extended period of time. The core comes out of the suspend under the following conditions : If there is any activity on the phy_line_state_i signal, or If the application writes to the Remote Wakeup Signaling bit in the Device Control register (DCTL.RmtWkUpSig). When the core comes out of the suspend, this bit is set to 1'b0. **Values:** 0x0 (INACTIVE): No suspend state 0x1 (ACTIVE): Suspend state

DIEPMSK Register (DIEPMSK)

Register	Offset	R/W	Description	Reset Value
DIEPMSK	HSUSBD_BA+0x810	R/W	Device IN Endpoint Common Interrupt Mask Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		NAKMsk	Reserved			BNAInIntrMsk	TxfifoUndrnMsk
7	6	5	4	3	2	1	0
Reserved	INEPNakEffMsk	INTknEPMisMsk	INTknTXFEmpMsk	TimeOUTMsk	AHBErrMsk	EPDisbldMsk	XferComplMsk

Bits	Description	
[31:14]	**Reserved**	**Reserved.**
[13]	**NAKMsk**	**NAK interrupt Mask** **Values:** 0x0 (MASK): Mask NAK Interrupt 0x1 (NOMASK): No Mask NAK Interrupt
[12:10]	**Reserved**	**Reserved.**
[9]	**BNAInIntrMsk**	**BNA interrupt Mask** **Values:** 0x0 (MASK): Mask BNA Interrupt 0x1 (NOMASK): No BNA Interrupt Mask
[8]	**TxfifoUndrnMsk**	**Fifo Underrun Mask** **Values:** 0x0 (MASK): Mask Fifo Underrun Interrupt 0x1 (NOMASK): No Fifo Underrun Interrupt Mask
[7]	**Reserved**	**Reserved.**
[6]	**INEPNakEffMsk**	**IN Endpoint NAK Effective Mask** **Values:** 0x0 (MASK): Mask IN Endpoint NAK Effective Interrupt 0x1 (NOMASK): No IN Endpoint NAK Effective Interrupt Mask
[5]	**INTknEPMisMsk**	**IN Token received with EP Mismatch Mask** **Values:** 0x0 (MASK): Mask IN Token received with EP Mismatch Interrupt 0x1 (NOMASK): No Mask IN Token received with EP Mismatch Interrupt
[4]	**INTknTXFEmpMsk**	**IN Token Received When TxFIFO Empty Mask** **Values:**

		0x0 (MASK): Mask IN Token Received When TxFIFO Empty Interrupt 0x1 (NOMASK): No IN Token Received When TxFIFO Empty Interrupt
[3]	**TimeOUTMsk**	**Timeout Condition Mask** **Values:** 0x0 (MASK): Mask Timeout Condition Interrupt 0x1 (NOMASK): No Timeout Condition Interrupt Mask
[2]	**AHBErrMsk**	**AHB Error Mask** **Values:** 0x0 (MASK): Mask AHB Error Interrupt 0x1 (NOMASK): No AHB Error Interrupt Mask
[1]	**EPDisbldMsk**	**Endpoint Disabled Interrupt Mask** **Values:** 0x0 (MASK): Mask Endpoint Disabled Interrupt 0x1 (NOMASK): No Endpoint Disabled Interrupt Mask
[0]	**XferComplMsk**	**Transfer Completed Interrupt Mask** **Values:** 0x0 (MASK): Mask Transfer Completed Interrupt 0x1 (NOMASK): No Transfer Completed Interrupt Mask

DOEPMSK Register (DOEPMSK)

Register	Offset	R/W	Description	Reset Value
DOEPMSK	HSUSBD_BA+0x814	R/W	Device OUT Endpoint Common Interrupt Mask Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**NYETMsk**	**NAKMsk**	**BbleErrMsk**	**Reserved**		**BnaOutIntrMsk**	**OutPktErrMsk**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**Back2BackSETup**	**StsPhseRcvdMsk**	**OUTTknEPdisMsk**	**SetUPMsk**	**AHBErrMsk**	**EPDisbldMsk**	**XferComplMsk**

Bits	Description	
[31:15]	**Reserved**	**Reserved.**
[14]	**NYETMsk**	**NYET interrupt Mask** **Values:** 0x0 (MASK): Mask NYET Interrupt 0x1 (NOMASK): No NYET Interrupt Mask
[13]	**NAKMsk**	**NAK interrupt Mask** **Values:** 0x0 (MASK): Mask NAK Interrupt 0x1 (NOMASK): No NAK Interrupt Mask
[12]	**BbleErrMsk**	**Babble Error interrupt Mask** **Values:** 0x0 (MASK): Mask Babble Error Interrupt 0x1 (NOMASK): No Babble Error Interrupt Mask
[11:10]	**Reserved**	**Reserved.**
[9]	**BnaOutIntrMsk**	**BNA interrupt Mask** **Values:** 0x0 (MASK): Mask BNA Interrupt 0x1 (NOMASK): No BNA Interrupt Mask
[8]	**OutPktErrMsk**	**OUT Packet Error Mask** **Values:** 0x0 (MASK): Mask OUT Packet Error Interrupt 0x1 (NOMASK): No OUT Packet Error Interrupt Mask
[7]	**Reserved**	**Reserved.**
[6]	**Back2BackSETup**	**Back-to-Back SETUP Packets Received Mask** Applies to control OUT endpoints only. **Values:**

		0x0 (MASK): Mask Back-to-Back SETUP Packets Received Interrupt 0x1 (NOMASK): No Back-to-Back SETUP Packets Received Interrupt Mask
[5]	**StsPhseRcvdMsk**	**Status Phase Received Mask** Applies to control OUT endpoints only. **Values:** 0x0 (MASK): Status Phase Received Mask 0x1 (NOMASK): No Status Phase Received Mask
[4]	**OUTTknEPdisMsk**	**OUT Token Received when Endpoint Disabled Mask** Applies to control OUT endpoints only. **Values:** 0x0 (MASK): Mask OUT Token Received when Endpoint Disabled Interrupt 0x1 (NOMASK): No OUT Token Received when Endpoint Disabled Interrupt Mask
[3]	**SetUPMsk**	**SETUP Phase Done Mask** Applies to control endpoints only. **Values:** 0x0 (MASK): Mask SETUP Phase Done Interrupt 0x1 (NOMASK): No SETUP Phase Done Interrupt Mask
[2]	**AHBErrMsk**	**AHB Error** **Values:** 0x0 (MASK): Mask AHB Error Interrupt 0x1 (NOMASK): No AHB Error Interrupt Mask
[1]	**EPDisbldMsk**	**Endpoint Disabled Interrupt Mask** **Values:** 0x0 (MASK): Mask Endpoint Disabled Interrupt 0x1 (NOMASK): No Endpoint Disabled Interrupt Mask
[0]	**XferComplMsk**	**Transfer Completed Interrupt Mask** **Values:** 0x0 (MASK): Mask Transfer Completed Interrupt 0x1 (NOMASK): No Transfer Completed Interrupt Mask

DAINT Register (DAINT)

Register	Offset	R/W	Description	Reset Value
DAINT	HSUSBD_BA+0x818	R	Device All Endpoints Interrupt Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							OutEPInt8
23	22	21	20	19	18	17	16
OutEPInt7	OutEPInt6	OutEPInt5	OutEPInt4	OutEPInt3	OutEPInt2	OutEPInt1	OutEPInt0
15	14	13	12	11	10	9	8
Reserved							InEpInt8
7	6	5	4	3	2	1	0
InEpInt7	InEpInt6	InEpInt5	InEpInt4	InEpInt3	InEpInt2	InEpInt1	InEpInt0

Bits	Description	
[31:25]	**Reserved**	**Reserved.**
[24]	**OutEPInt8**	**OUT Endpoint 8 Interrupt Bit**
[23]	**OutEPInt7**	**OUT Endpoint 7 Interrupt Bit**
[22]	**OutEPInt6**	**OUT Endpoint 6 Interrupt Bit**
[21]	**OutEPInt5**	**OUT Endpoint 5 Interrupt Bit**
[20]	**OutEPInt4**	**OUT Endpoint 4 Interrupt Bit**
[19]	**OutEPInt3**	**OUT Endpoint 3 Interrupt Bit**
[18]	**OutEPInt2**	**OUT Endpoint 2 Interrupt Bit**
[17]	**OutEPInt1**	**OUT Endpoint 1 Interrupt Bit**
[16]	**OutEPInt0**	**OUT Endpoint 0 Interrupt Bit** **Values:** 0x0 (INACTIVE): No Interrupt 0x1 (ACTIVE): OUT Endpoint 0 Interrupt
[15:9]	**Reserved**	**Reserved.**
[8]	**InEpInt8**	**IN Endpoint 8 Interrupt Bit**
[7]	**InEpInt7**	**IN Endpoint 7 Interrupt Bit**
[6]	**InEpInt6**	**IN Endpoint 6 Interrupt Bit**
[5]	**InEpInt5**	**IN Endpoint 5 Interrupt Bit**
[4]	**InEpInt4**	**IN Endpoint 4 Interrupt Bit**
[3]	**InEpInt3**	**IN Endpoint 3 Interrupt Bit**
[2]	**InEpInt2**	**IN Endpoint 2 Interrupt Bit**
[1]	**InEpInt1**	**IN Endpoint 1 Interrupt Bit**

[0]	**InEpInt0**	**IN Endpoint 0 Interrupt Bit** **Values:** 0x0 (INACTIVE): No Interrupt 0x1 (ACTIVE): IN Endpoint 0 Interrupt

DAINTMSK Register (DAINTMSK)

Register	Offset	R/W	Description	Reset Value
DAINTMSK	HSUSBD_BA+0x81c	R/W	Device All Endpoints Interrupt Mask Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							**OutEPMsk8**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
OutEPMsk7	**OutEPMsk6**	**OutEPMsk5**	**OutEPMsk4**	**OutEPMsk3**	**OutEPMsk2**	**OutEPMsk1**	**OutEPMsk0**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**InEpMsk8**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
InEpMsk7	**InEpMsk6**	**InEpMsk5**	**InEpMsk4**	**InEpMsk3**	**InEpMsk2**	**InEpMsk1**	**InEpMsk0**

Bits	Description	
[31:25]	**Reserved**	**Reserved.**
[24]	**OutEPMsk8**	**OUT Endpoint 8 Interrupt mask Bit**
[23]	**OutEPMsk7**	**OUT Endpoint 7 Interrupt mask Bit**
[22]	**OutEPMsk6**	**OUT Endpoint 6 Interrupt mask Bit**
[21]	**OutEPMsk5**	**OUT Endpoint 5 Interrupt mask Bit**
[20]	**OutEPMsk4**	**OUT Endpoint 4 Interrupt mask Bit**
[19]	**OutEPMsk3**	**OUT Endpoint 3 Interrupt mask Bit**
[18]	**OutEPMsk2**	**OUT Endpoint 2 Interrupt mask Bit**
[17]	**OutEPMsk1**	**OUT Endpoint 1 Interrupt mask Bit**
[16]	**OutEPMsk0**	**OUT Endpoint 0 Interrupt mask Bit** **Values:** 0x0 (MASK): OUT Endpoint 0 Interrupt mask 0x1 (NOMASK): No Interrupt mask
[15:9]	**Reserved**	**Reserved.**
[8]	**InEpMsk8**	**IN Endpoint 8 Interrupt mask Bit**
[7]	**InEpMsk7**	**IN Endpoint 7 Interrupt mask Bit**
[6]	**InEpMsk6**	**IN Endpoint 6 Interrupt mask Bit**
[5]	**InEpMsk5**	**IN Endpoint 5 Interrupt mask Bit**
[4]	**InEpMsk4**	**IN Endpoint 4 Interrupt mask Bit**
[3]	**InEpMsk3**	**IN Endpoint 3 Interrupt mask Bit**
[2]	**InEpMsk2**	**IN Endpoint 2 Interrupt mask Bit**
[1]	**InEpMsk1**	**IN Endpoint 1 Interrupt mask Bit**

[0]	**InEpMsk0**	**IN Endpoint 0 Interrupt mask Bit** **Values:** 0x0 (MASK): IN Endpoint 0 Interrupt mask 0x1 (NOMASK): No Interrupt mask

DTHRCTL Register (DTHRCTL)

Register	Offset	R/W	Description	Reset Value
DTHRCTL	HSUSBD_BA+0x830	R/W	Device Threshold Control Register	0x0C10_0020

31	30	29	28	27	26	25	24
Reserved				ArbPrkEn	Reserved	RxThrLen	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RxThrLen							RxThrEn
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			AHBThrRatio		TxThrLen		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TxThrLen						ISOThrEn	NonISOThrEn

Bits	Description	
[31:28]	**Reserved**	**Reserved.**
[27]	**ArbPrkEn**	**Arbiter Parking Enable** This bit controls internal DMA arbiter parking for IN endpoints. If thresholding is enabled and this bit is set to one, then the arbiter parks on the IN endpoint for which there is a token received on the USB. This is done to avoid getting into underrun conditions. By default, arbiter parking is enabled. **Values:** 0x0 (DISABLED): Disable DMA arbiter parking 0x1 (ENABLED): Enable DMA arbiter parking for IN endpoints
[26]	**Reserved**	**Reserved.**
[25:17]	**RxThrLen**	**Receive Threshold Length** This field specifies Receive thresholding size in DWORDS. This field also specifies the amount of data received on the USB before the core can start transmitting on the AHB. The threshold length has to be at least eight DWORDS. The recommended value for ThrLen is to be the same as the programmed AHB Burst Length (GAHBCFG.HBstLen).
[16]	**RxThrEn**	**Receive Threshold Enable** When this bit is set, the core enables thresholding in the receive direction. **Note:** We recommends that you do not enable RxThrEn, because it may cause issues in the RxFIFO especially during error conditions such as RxError and Babble. **Values:** 0x0 (DISABLED): Disable thresholding 0x1 (ENABLED): Enable thresholding in the receive direction
[15:13]	**Reserved**	**Reserved.**
[12:11]	**AHBThrRatio**	**AHB Threshold Ratio** These bits define the ratio between the AHB threshold and the MAC threshold for the transmit path only. The AHB threshold always remains less than or equal to the USB threshold, because this does not increase overhead. Both the AHB and the MAC threshold must be DWORD-aligned. The application needs to program TxThrLen and the AHBThrRatio to make the AHB Threshold value DWORD aligned. If the AHB threshold value is not DWORD aligned, the core might not behave correctly. When programming the TxThrLen and AHBThrRatio, the application must ensure that the minimum AHB threshold value does not go below 8 DWORDS to meet the USB turnaround time requirements.

		2'b00: AHB threshold = MAC threshold. 2'b01: AHB threshold = MAC threshold / 2. 2'b10: AHB threshold = MAC threshold / 4. 2'b11: AHB threshold = MAC threshold / 8. **Values:** 0x0 (THRESZERO): AHB threshold = MAC threshold. 0x1 (THRESONE): AHB threshold = MAC threshold /2. 0x2 (THRESTWO): AHB threshold = MAC threshold /4. 0x3 (THRESTHREE): AHB threshold = MAC threshold /8.
[10:2]	**TxThrLen**	**Transmit Threshold Length** This field specifies Transmit thresholding size in DWORDS. This also forms the MAC threshold and specifies the amount of data in bytes to be in the corresponding endpoint transmit FIFO, before the core can start transmit on the USB. The threshold length has to be at least eight DWORDS when the value of AHBThrRatio is 2'h00. In case the AHBThrRatio is non zero the application needs to ensure that the AHB Threshold value does not go below the recommended eight DWORD. This field controls both isochronous and non-isochronous IN endpoint thresholds. The recommended value for ThrLen is to be the same as the programmed AHB Burst Length (GAHBCFG.HBstLen). **Note:** When OTG_ARCHITECTURE=2, the reset value of this register field is 8.
[1]	**ISOThrEn**	**ISO IN Endpoints Threshold Enable** When this bit is Set, the core enables thresholding for isochronous IN endpoints. **Values:** 0x0 (DISABLED): No thresholding 0x1 (ENABLED): Enables thresholding for isochronous IN endpoints
[0]	**NonISOThrEn**	**Non-ISO IN Endpoints Threshold Enable** When this bit is Set, the core enables thresholding for Non Isochronous IN endpoints. **Values:** 0x0 (DISABLED): No thresholding 0x1 (ENABLED): Enable thresholding for non-isochronous IN endpoints

DIEPEMPMSK Register (DIEPEMPMSK)

Register	Offset	R/W	Description	Reset Value
DIEPEMPMSK	HSUSBD_BA+0x834	R/W	Device IN Endpoint FIFO Empty Interrupt Mask Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
InEpTxfEmpMsk							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
InEpTxfEmpMsk							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**InEpTxfEmpMsk**	**IN EP Tx FIFO Empty Interrupt Mask Bits** These bits acts as mask bits for DIEPINTn.TxFEmp interrupt, one bit per IN Endpoint: Bit 0 for IN EP 0, bit 8 for IN EP 8 Values: 0x1 (EP0_MASK): Mask IN EP0 Tx FIFO Empty Interrupt 0x2 (EP1_MASK): Mask IN EP1 Tx FIFO Empty Interrupt 0x4 (EP2_MASK): Mask IN EP2 Tx FIFO Empty Interrupt 0x8 (EP3_MASK): Mask IN EP3 Tx FIFO Empty Interrupt 0x10 (EP4_MASK): Mask IN EP4 Tx FIFO Empty Interrupt 0x20 (EP5_MASK): Mask IN EP5 Tx FIFO Empty Interrupt 0x40 (EP6_MASK): Mask IN EP6 Tx FIFO Empty Interrupt 0x80 (EP7_MASK): Mask IN EP7 Tx FIFO Empty Interrupt 0x100 (EP8_MASK): Mask IN EP8 Tx FIFO Empty Interrupt

DIEPCTL0 Register (DIEPCTL0)

Register	Offset	R/W	Description	Reset Value
DIEPCTL0	HSUSBD_BA+0x900	R/W	Device Control IN Endpoint 0 Control Register	0x0000_8000

31	30	29	28	27	26	25	24
EPEna	EPDis	Reserved		SNAK	CNAK	TxFNum	
23	22	21	20	19	18	17	16
TxFNum		Stall	Reserved	EPType		NAKSts	Reserved
15	14	13	12	11	10	9	8
USBActEP	Reserved						
7	6	5	4	3	2	1	0
Reserved						MPS	

Bits	Description	
[31]	**EPEna**	**Endpoint Enable** When Scatter/Gather DMA mode is enabled for IN endpoints, this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. **Values:** 0x0 (INACTIVE): No action 0x1 (ACTIVE): Enable Endpoint
[30]	**EPDis**	**Endpoint Disable** The application sets this bit to stop transmitting data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled Interrupt. The application must Set this bit only if Endpoint Enable is already set for this endpoint. **Values:** 0x0 (INACTIVE): No action 0x1 (ACTIVE): Disabled Endpoint
[29:28]	**Reserved**	**Reserved.**
[27]	**SNAK**	**Set NAK** **A** write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint The core can also set this bit for an endpoint after a SETUP packet is received on that endpoint **Values:** 0x0 (NOSET): No action 0x1 (SET): Set NAK
[26]	**CNAK**	**Clear NAK** A write to this bit clears the NAK bit for the endpoint **Values:** 0x0 (NOCLEAR): No action 0x1 (CLEAR): Clear NAK
[25:22]	**TxFNum**	**TxFIFO Number** For Dedicated FIFO operation, this value is set to the FIFO number that is assigned to IN Endpoint.

		Values: 0x0 (TXFIFO0): Tx FIFO 0 0x1 (TXFIFO1): Tx FIFO 1 0x2 (TXFIFO2): Tx FIFO 2 0x3 (TXFIFO3): Tx FIFO 3 0x4 (TXFIFO4): Tx FIFO 4 0x5 (TXFIFO5): Tx FIFO 5 0x6 (TXFIFO6): Tx FIFO 6 0x7 (TXFIFO7): Tx FIFO 7 0x8 (TXFIFO8): Tx FIFO 8
[21]	**Stall**	**STALL Handshake** The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Nonperiodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. **Values:** 0x0 (INACTIVE): No Stall 0x1 (ACTIVE): Stall Handshake
[20]	**Reserved**	**Reserved.**
[19:18]	**EPType**	**Endpoint Type (Read Only)** Hardcoded to 00 for control. **Values:** 0x0 (ACTIVE): Endpoint Control 0
[17]	**NAKSts**	**NAK Status (Read Only)** Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status 1'b1: The core is transmitting NAK handshakes on this endpoint. When this bit is set, either by the application or core, the core stops transmitting data, even If there is data available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (INACTIVE): The core is transmitting non-NAK handshakes based on the FIFO status 0x1 (ACTIVE): The core is transmitting NAK handshakes on this endpoint
[16]	**Reserved**	**Reserved.**
[15]	**USBActEP**	**USB Active Endpoint (Read Only)** This bit is always SET to 1, indicating that control endpoint 0 is always active in all configurations and interfaces. **Values:** 0x1 (ACTIVE0): Control endpoint is always active
[14:2]	**Reserved**	**Reserved.**
[1:0]	**MPS**	**Maximum Packet Size** Applies to IN and OUT endpoints. The application must program this field with the maximum packet size for the current logical endpoint. 2'b00: 64 bytes 2'b01: 32 bytes 2'b10: 16 bytes

		2'b11: 8 bytes **Values:** 0x0 (BYTES64): 64 bytes 0x1 (BYTES32): 32 bytes 0x2 (BYTES16): 16 bytes 0x3 (BYTES8): 8 bytes

DIEPINT0 Register (DIEPINT0)

Register	Offset	R/W	Description	Reset Value
DIEPINT0	HSUSBD_BA+0x908	R/W	Device IN Endpoint 0 Interrupt Register	0x0000_0080

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	NYETIntrpt	NAKIntrpt	BbleErr	Reserved		BNAIntr	TxfifoUndrn
7	6	5	4	3	2	1	0
TxFEmp	INEPNakEff	INTknEPMis	INTknTXFEmp	Reserved	AHBErr	EPDisbld	XferCompl

Bits	Description	
[31:15]	**Reserved**	**Reserved.**
[14]	**NYETIntrpt**	**NYET Interrupt** The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint. **Values:** 0x0 (INACTIVE): No interrupt 0x1 (ACTIVE): NYET Interrupt
[13]	**NAKIntrpt**	**NAK Interrupt** The core generates this interrupt when a NAK is transmitted or received by the device. <brIn case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo. **Values:** 0x0 (INACTIVE): No interrupt 0x1 (ACTIVE): NAK Interrupt
[12]	**BbleErr**	**NAK Interrupt** The core generates this interrupt when babble is received for the endpoint. **Values:** 0x0 (INACTIVE): No interrupt 0x1 (ACTIVE): BbleErr interrupt
[11:10]	**Reserved**	**Reserved.**
[9]	**BNAIntr**	**Buffer Not Available Interrupt** The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as DMA done. **Values:** 0x0 (INACTIVE): No BNA interrupt 0x1 (ACTIVE): BNA interrupt
[8]	**TxfifoUndrn**	**Fifo Underrun** Applies to IN endpoints only.

		The core generates this interrupt when it detects a transmit FIFO underrun condition in threshold mode for this endpoint. **Values:** 0x0 (INACTIVE): No Fifo Underrun interrupt 0x1 (ACTIVE): Fifo Underrun interrupt
[7]	**TxFEmp**	**Transmit FIFO Empty (Read Only)** This bit is valid only for IN Endpoints This interrupt is asserted when the TxFIFO for this endpoint is either half or completely empty. The half or completely empty status is determined by the TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.NPTxFEmpLvl)). **Values:** 0x0 (INACTIVE): No Transmit FIFO Empty interrupt 0x1 (ACTIVE): Transmit FIFO Empty interrupt
[6]	**INEPNakEff**	**IN Endpoint NAK Effective** Applies to periodic IN endpoints only. This bit can be cleared when the application clears the IN endpoint NAK by writing to DIEPCTLn.CNAK. This interrupt indicates that the core has sampled the NAK bit Set (either by the application or by the core). The interrupt indicates that the IN endpoint NAK bit Set by the application has taken effect in the core. This interrupt does not guarantee that a NAK handshake is sent on the USB. A STALL bit takes priority over a NAK bit. **Values:** 0x0 (INACTIVE): No IN Endpoint NAK Effective interrupt 0x1 (ACTIVE): IN Endpoint NAK Effective interrupt
[5]	**INTknEPMis**	**IN Token Received with EP Mismatch** Applies to non-periodic IN endpoints only. Indicates that the data in the top of the non-periodic TxFIFO belongs to an endpoint other than the one for which the IN token was received. This interrupt is asserted on the endpoint for which the IN token was received. **Values:** 0x0 (INACTIVE): No IN Token Received with EP Mismatch interrupt 0x1 (ACTIVE): IN Token Received with EP Mismatch interrupt
[4]	**INTknTXFEmp**	**IN Token Received When TxFIFO is Empty** Applies to non-periodic IN endpoints only. Indicates that an IN token was received when the associated TxFIFO (periodic/non-periodic) was empty. This interrupt is asserted on the endpoint for which the IN token was received. **Values:** 0x0 (INACTIVE): No IN Token Received when TxFIFO Empty interrupt 0x1 (ACTIVE): IN Token Received when TxFIFO Empty Interrupt
[3]	**Reserved**	**Reserved.**
[2]	**AHBErr**	**AHB Error** Applies to IN and OUT endpoints. When there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address. **Values:** 0x0 (INACTIVE): No AHB Error Interrupt 0x1 (ACTIVE): AHB Error interrupt
[1]	**EPDisbld**	**Endpoint Disabled Interrupt**

		Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request. **Values:** 0x0 (INACTIVE): No Endpoint Disabled Interrupt 0x1 (ACTIVE): Endpoint Disabled Interrupt
[0]	**XferCompl**	**Transfer Completed Interrupt** Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is set. **Values:** 0x0 (INACTIVE): No Transfer Complete Interrupt 0x1 (ACTIVE): Transfer Completed Interrupt

DIEPDMA0 Register (DIEPDMA0)

Register	Offset	R/W	Description	Reset Value
DIEPDMA0	HSUSBD_BA+0x914	R/W	Device IN Endpoint 0 DMA Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMAAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMAAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMAAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMAAddr							

Bits	Description	
[31:0]	**DMAAddr**	**DMAAddr** This field indicates the base pointer for the descriptor list.

DTXFSTS0 Register (DTXFSTS0)

Register	Offset	R/W	Description	Reset Value
DTXFSTS0	HSUSBD_BA+0x918	R	Device IN Endpoint Transmit FIFO Status Register 0	0x0000_0020

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
INEPTxFSpcAvail							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INEPTxFSpcAvail							

Bits	Description	
[31:16]	**Reserved**	**Reserved.**
[15:0]	**INEPTxFSpcAvail**	**IN Endpoint TxFIFO Space Avail** Indicates the amount of free space available in the Endpoint TxFIFO. Values are in terms of 32-bit words. 16'h0: Endpoint TxFIFO is full 16'h1: 1 word available 16'h2: 2 words available 16'hn: n words available (where 0 n 32,768) 16'h8000: 32,768 words available Others: Reserved

DIEPDMAB0 Register (DIEPDMAB0)

Register	Offset	R/W	Description	Reset Value
DIEPDMAB0	HSUSBD_BA+0x91c	R	Device IN Endpoint 16 Buffer Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMABufferAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMABufferAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMABufferAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMABufferAddr							

Bits	Description	
[31:0]	**DMABufferAddr**	**Holds the current buffer address This register is updated as and when the data transfer for the corresponding end point is in progress.**

DIEPCTL1 Register (DIEPCTL1)

Register	Offset	R/W	Description	Reset Value
DIEPCTL1	HSUSBD_BA+0x920	R/W	Device Control IN Endpoint 1 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
EPEna	EPDis	Reserved		SNAK	CNAK	TxFNum	
23	22	21	20	19	18	17	16
TxFNum		Stall	Reserved	EPType		NAKSts	DPID
15	14	13	12	11	10	9	8
USBActEP	Reserved				MPS		
7	6	5	4	3	2	1	0
MPS							

Bits	Description	
[31]	**EPEna**	**Endpoint Enable** Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled, For IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. For OUT endpoint it indicates that the descriptor structure and data buffer to receive data is setup. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done Endpoint Disabled Transfer Completed **Note:** For control endpoints in DMA mode, this bit must be set to be able to transfer SETUP data packets in memory. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Enable Endpoint
[30]	**EPDis**	**Endpoint Disable** Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Disable Endpoint
[29:28]	**Reserved**	**Reserved.**
[27]	**SNAK**	**Set NAK** A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also Set this bit for an endpoint after a SETUP packet is received on that endpoint. **Values:**

		0x0 (INACTIVE): No Set NAK 0x1 (ACTIVE): Set NAK
[26]	**CNAK**	**Clear NAK** A write to this bit clears the NAK bit for the endpoint. **Values:** 0x0 (INACTIVE): No Clear NAK 0x1 (ACTIVE): Clear NAK
[25:22]	**TxFNum**	**TxFIFO Number** Dedicated FIFO Operation: These bits specify the FIFO number associated with this endpoint. Each active IN endpoint must be programmed to a separate FIFO number. This field is valid only for IN endpoints. **Values:** 0x0 (TXFIFO0): Tx FIFO 0 0x1 (TXFIFO1): Tx FIFO 1 0x2 (TXFIFO2): Tx FIFO 2 0x3 (TXFIFO3): Tx FIFO 3 0x4 (TXFIFO4): Tx FIFO 4 0x5 (TXFIFO5): Tx FIFO 5 0x6 (TXFIFO6): Tx FIFO 6 0x7 (TXFIFO7): Tx FIFO 7 0x8 (TXFIFO8): Tx FIFO 8
[21]	**Stall**	**STALL Handshake** Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core. Applies to control endpoints only. The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (INACTIVE): STALL All non-active tokens 0x1 (ACTIVE): STALL All Active Tokens
[20]	**Reserved**	**Reserved.**
[19:18]	**EPType**	**Endpoint Type** This is the transfer type supported by this logical endpoint 2'b00: Control 2'b01: Isochronous 2'b10: Bulk 2'b11: Interrupt **Values:** 0x0 (CONTROL): Control 0x1 (ISOCHRONOUS): Isochronous 0x2 (BULK): Bulk 0x3 (INTERRUP): Interrupt
[17]	**NAKSts**	**NAK Status (Read Only)** Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status.

		1'b1: The core is transmitting NAK handshakes on this endpoint. When either the application or the core sets this bit: The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet. For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the TxFIFO. For isochronous IN endpoints: The core sends out a zero-length data packet, even if there data is available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (NONNAK): The core is transmitting non-NAK handshakes based on the FIFO status 0x1 (NAK): The core is transmitting NAK handshakes on this endpoint
[16]	**DPID**	**Endpoint Data PID (Read Only)** Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. The applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID. 1'b0: DATA0 1'b1: DATA1 **Values:** 0x0 (DATA0EVENFRM): DATA0 or Even Frame 0x1 (DATA1ODDFRM): DATA1 or Odd Frame
[15]	**USBActEP**	**USB Active Endpoint** Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints (other than EP 0) after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit. **Values:** 0x0 (DISABLED): Not Active 0x1 (ENABLED): USB Active Endpoint
[14:11]	**Reserved**	**Reserved.**
[10:0]	**MPS**	**Maximum Packet Size** The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.

DIEPINT1 Register (DIEPINT1)

Register	Offset	R/W	Description	Reset Value
DIEPINT1	HSUSBD_BA+0x928	R/W	Device IN Endpoint 1 Interrupt Register	0x0000_0080

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	NYETIntrpt	NAKIntrpt	BbleErr	Reserved		BNAIntr	TxfifoUndrn
7	6	5	4	3	2	1	0
TxFEmp	INEPNakEff	INTknEPMis	INTknTXFEmp	Reserved	AHBErr	EPDisbld	XferCompl

Bits	Description	
[31:15]	**Reserved**	**Reserved.**
[14]	**NYETIntrpt**	**NYET Interrupt** The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint. **Values:** 0x0 (INACTIVE): No NYET interrupt 0x1 (ACTIVE): NYET Interrupt
[13]	**NAKIntrpt**	**NAK Interrupt** The core generates this interrupt when a NAK is transmitted or received by the device. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo. **Values:** 0x0 (INACTIVE): No NAK interrupt 0x1 (ACTIVE): NAK Interrupt
[12]	**BbleErr**	**NAK Interrupt** The core generates this interrupt when babble is received for the endpoint. **Values:** 0x0 (INACTIVE): No interrupt 0x1 (ACTIVE): BbleErr interrupt
[11:10]	**Reserved**	**Reserved.**
[9]	**BNAIntr**	**Buffer Not Available Interrupt** The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as Host busy or DMA done. **Values:** 0x0 (INACTIVE): No BNA interrupt 0x1 (ACTIVE): BNA interrupt
[8]	**TxfifoUndrn**	**Fifo Underrun** Applies to IN endpoints Only

		This bit is valid only If thresholding is enabled. The core generates this interrupt when it detects a transmit FIFO underrun condition for this endpoint. **Values:** 0x0 (INACTIVE): No Tx FIFO Underrun interrupt 0x1 (ACTIVE): TxFIFO Underrun interrupt
[7]	**TxFEmp**	**Transmit FIFO Empty (Read Only)** This bit is valid only for IN endpoints This interrupt is asserted when the TxFIFO for this endpoint is either half or completely empty. The half or completely empty status is determined by the TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.NPTxFEmpLvl)). **Values:** 0x0 (INACTIVE): No Transmit FIFO Empty interrupt 0x1 (ACTIVE): Transmit FIFO Empty interrupt
[6]	**INEPNakEff**	**IN Endpoint NAK Effective** Applies to periodic IN endpoints only. This bit can be cleared when the application clears the IN endpoint NAK by writing to DIEPCTLn.CNAK. This interrupt indicates that the core has sampled the NAK bit Set (either by the application or by the core). The interrupt indicates that the IN endpoint NAK bit Set by the application has taken effect in the core. This interrupt does not guarantee that a NAK handshake is sent on the USB. A STALL bit takes priority over a NAK bit. **Values:** 0x0 (INACTIVE): No Endpoint NAK Effective interrupt 0x1 (ACTIVE): IN Endpoint NAK Effective interrupt
[5]	**INTknEPMis**	**IN Token Received with EP Mismatch** Applies to non-periodic IN endpoints only. Indicates that the data in the top of the non-periodic TxFIFO belongs to an endpoint other than the one for which the IN token was received. This interrupt is asserted on the endpoint for which the IN token was received. **Values:** 0x0 (INACTIVE): No IN Token Received with EP Mismatch interrupt 0x1 (ACTIVE): IN Token Received with EP Mismatch interrupt
[4]	**INTknTXFEmp**	**IN Token Received When TxFIFO is Empty** Applies to non-periodic IN endpoints only. Indicates that an IN token was received when the associated TxFIFO (periodic/non-periodic) was empty. This interrupt is asserted on the endpoint for which the IN token was received. **Values:** 0x0 (INACTIVE): No IN Token Received interrupt 0x1 (ACTIVE): IN Token Received Interrupt
[3]	**Reserved**	**Reserved.**
[2]	**AHBErr**	**AHB Error** Applies to IN and OUT endpoints. When there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address. **Values:** 0x0 (INACTIVE): No AHB Error Interrupt 0x1 (ACTIVE): AHB Error interrupt
[1]	**EPDisbld**	**Endpoint Disabled Interrupt**

		Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request. **Values:** 0x0 (INACTIVE): No Endpoint Disabled Interrupt 0x1 (ACTIVE): Endpoint Disabled Interrupt
[0]	**XferCompl**	**Transfer Completed Interrupt** Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is set. **Values:** 0x0 (INACTIVE): No Transfer Complete Interrupt 0x1 (ACTIVE): Transfer Complete Interrupt

DIEPDMA1 Register (DIEPDMA1)

Register	Offset	R/W	Description	Reset Value
DIEPDMA1	HSUSBD_BA+0x934	R/W	Device IN Endpoint 1 DMA Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMAAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMAAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMAAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMAAddr							

Bits	Description	
[31:0]	**DMAAddr**	**Holds the start address of the external memory for storing or fetching endpoint data** **Note:** For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address. This field indicates the base pointer for the descriptor list.

DTXFSTS1 Register (DTXFSTS1)

Register	Offset	R/W	Description	Reset Value
DTXFSTS1	HSUSBD_BA+0x938	R	Device IN Endpoint Transmit FIFO Status Register 1	0x0000_0020

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
INEPTxFSpcAvail							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INEPTxFSpcAvail							

Bits	Description	
[31:16]	**Reserved**	**Reserved.**
[15:0]	**INEPTxFSpcAvail**	**IN Endpoint TxFIFO Space Avail** Indicates the amount of free space available in the Endpoint TxFIFO. Values are in terms of 32-bit words. 16'h0: Endpoint TxFIFO is full 16'h1: 1 word available 16'h2: 2 words available 16'hn: n words available (where 0 n 32,768) 16'h8000: 32,768 words available Others: Reserved

DIEPDMAB1 Register (DIEPDMAB1)

Register	Offset	R/W	Description	Reset Value
DIEPDMAB1	HSUSBD_BA+0x93c	R	Device IN Endpoint 1 Buffer Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMABufferAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMABufferAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMABufferAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMABufferAddr							

Bits	Description	
[31:0]	**DMABufferAddr**	**Holds the current buffer address This register is updated as and when the data transfer for the corresponding end point is in progress**

DIEPCTL2 Register (DIEPCTL2)

Register	Offset	R/W	Description	Reset Value
DIEPCTL2	HSUSBD_BA+0x940	R/W	Device Control IN Endpoint 2 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
EPEna	EPDis	Reserved		SNAK	CNAK	TxFNum	
23	22	21	20	19	18	17	16
TxFNum		Stall	Reserved	EPType		NAKSts	DPID
15	14	13	12	11	10	9	8
USBActEP	Reserved				MPS		
7	6	5	4	3	2	1	0
MPS							

Bits	Description	
[31]	**EPEna**	**Endpoint Enable** Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled, For IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. For OUT endpoint it indicates that the descriptor structure and data buffer to receive data is setup. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done Endpoint Disabled Transfer Completed **Note:** For control endpoints in DMA mode, this bit must be set to be able to transfer SETUP data packets in memory. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Enable Endpoint
[30]	**EPDis**	**Endpoint Disable** Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Disable Endpoint
[29:28]	**Reserved**	**Reserved.**
[27]	**SNAK**	**Set NAK** A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also Set this bit for an endpoint after a SETUP packet is received on that endpoint. **Values:**

		0x0 (INACTIVE): No Set NAK 0x1 (ACTIVE): Set NAK
[26]	**CNAK**	**Clear NAK** A write to this bit clears the NAK bit for the endpoint. **Values:** 0x0 (INACTIVE): No Clear NAK 0x1 (ACTIVE): Clear NAK
[25:22]	**TxFNum**	**TxFIFO Number** Dedicated FIFO Operation: These bits specify the FIFO number associated with this endpoint. Each active IN endpoint must be programmed to a separate FIFO number. This field is valid only for IN endpoints. **Values:** 0x0 (TXFIFO0): Tx FIFO 0 0x1 (TXFIFO1): Tx FIFO 1 0x2 (TXFIFO2): Tx FIFO 2 0x3 (TXFIFO3): Tx FIFO 3 0x4 (TXFIFO4): Tx FIFO 4 0x5 (TXFIFO5): Tx FIFO 5 0x6 (TXFIFO6): Tx FIFO 6 0x7 (TXFIFO7): Tx FIFO 7 0x8 (TXFIFO8): Tx FIFO 8
[21]	**Stall**	**STALL Handshake** Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core. Applies to control endpoints only. The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (INACTIVE): STALL All non-active tokens 0x1 (ACTIVE): STALL All Active Tokens
[20]	**Reserved**	**Reserved.**
[19:18]	**EPType**	**Endpoint Type** This is the transfer type supported by this logical endpoint 2'b00: Control 2'b01: Isochronous 2'b10: Bulk 2'b11: Interrupt **Values:** 0x0 (CONTROL): Control 0x1 (ISOCHRONOUS): Isochronous 0x2 (BULK): Bulk 0x3 (INTERRUP): Interrupt
[17]	**NAKSts**	**NAK Status (Read Only)** Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status.

		1'b1: The core is transmitting NAK handshakes on this endpoint. When either the application or the core sets this bit: The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet. For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the TxFIFO. For isochronous IN endpoints: The core sends out a zero-length data packet, even if there data is available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (NONNAK): The core is transmitting non-NAK handshakes based on the FIFO status 0x1 (NAK): The core is transmitting NAK handshakes on this endpoint
[16]	**DPID**	**Endpoint Data PID (Read Only)** Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. The applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID. 1'b0: DATA0 1'b1: DATA1 **Values:** 0x0 (DATA0EVENFRM): DATA0 or Even Frame 0x1 (DATA1ODDFRM): DATA1 or Odd Frame
[15]	**USBActEP**	**USB Active Endpoint** Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints (other than EP 0) after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit. **Values:** 0x0 (DISABLED): Not Active 0x1 (ENABLED): USB Active Endpoint
[14:11]	**Reserved**	**Reserved.**
[10:0]	**MPS**	**Maximum Packet Size** The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.

DIEPINT2 Register (DIEPINT2)

Register	Offset	R/W	Description	Reset Value
DIEPINT2	HSUSBD_BA+0x948	R/W	Device IN Endpoint 2 Interrupt Register	0x0000_0080

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	NYETIntrpt	NAKIntrpt	BbleErr	Reserved		BNAIntr	TxfifoUndrn
7	6	5	4	3	2	1	0
TxFEmp	INEPNakEff	INTknEPMis	INTknTXFEmp	Reserved	AHBErr	EPDisbld	XferCompl

Bits	Description	
[31:15]	**Reserved**	**Reserved.**
[14]	**NYETIntrpt**	**NYET Interrupt** The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint. **Values:** 0x0 (INACTIVE): No NYET interrupt 0x1 (ACTIVE): NYET Interrupt
[13]	**NAKIntrpt**	**NAK Interrupt** The core generates this interrupt when a NAK is transmitted or received by the device. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo. **Values:** 0x0 (INACTIVE): No NAK interrupt 0x1 (ACTIVE): NAK Interrupt
[12]	**BbleErr**	**NAK Interrupt** The core generates this interrupt when babble is received for the endpoint. **Values:** 0x0 (INACTIVE): No interrupt 0x1 (ACTIVE): BbleErr interrupt
[11:10]	**Reserved**	**Reserved.**
[9]	**BNAIntr**	**Buffer Not Available Interrupt** The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as Host busy or DMA done. **Values:** 0x0 (INACTIVE): No BNA interrupt 0x1 (ACTIVE): BNA interrupt
[8]	**TxfifoUndrn**	**Fifo Underrun** Applies to IN endpoints Only

		This bit is valid only If thresholding is enabled. The core generates this interrupt when it detects a transmit FIFO underrun condition for this endpoint. **Values:** 0x0 (INACTIVE): No Tx FIFO Underrun interrupt 0x1 (ACTIVE): TxFIFO Underrun interrupt
[7]	**TxFEmp**	**Transmit FIFO Empty (Read Only)** This bit is valid only for IN endpoints This interrupt is asserted when the TxFIFO for this endpoint is either half or completely empty. The half or completely empty status is determined by the TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.NPTxFEmpLvl)). **Values:** 0x0 (INACTIVE): No Transmit FIFO Empty interrupt 0x1 (ACTIVE): Transmit FIFO Empty interrupt
[6]	**INEPNakEff**	**IN Endpoint NAK Effective** Applies to periodic IN endpoints only. This bit can be cleared when the application clears the IN endpoint NAK by writing to DIEPCTLn.CNAK. This interrupt indicates that the core has sampled the NAK bit Set (either by the application or by the core). The interrupt indicates that the IN endpoint NAK bit Set by the application has taken effect in the core. This interrupt does not guarantee that a NAK handshake is sent on the USB. A STALL bit takes priority over a NAK bit. **Values:** 0x0 (INACTIVE): No Endpoint NAK Effective interrupt 0x1 (ACTIVE): IN Endpoint NAK Effective interrupt
[5]	**INTknEPMis**	**IN Token Received with EP Mismatch** Applies to non-periodic IN endpoints only. Indicates that the data in the top of the non-periodic TxFIFO belongs to an endpoint other than the one for which the IN token was received. This interrupt is asserted on the endpoint for which the IN token was received. **Values:** 0x0 (INACTIVE): No IN Token Received with EP Mismatch interrupt 0x1 (ACTIVE): IN Token Received with EP Mismatch interrupt
[4]	**INTknTXFEmp**	**IN Token Received When TxFIFO is Empty** Applies to non-periodic IN endpoints only. Indicates that an IN token was received when the associated TxFIFO (periodic/non-periodic) was empty. This interrupt is asserted on the endpoint for which the IN token was received. **Values:** 0x0 (INACTIVE): No IN Token Received interrupt 0x1 (ACTIVE): IN Token Received Interrupt
[3]	**Reserved**	**Reserved.**
[2]	**AHBErr**	**AHB Error** Applies to IN and OUT endpoints. When there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address. **Values:** 0x0 (INACTIVE): No AHB Error Interrupt 0x1 (ACTIVE): AHB Error interrupt
[1]	**EPDisbld**	**Endpoint Disabled Interrupt**

		Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request. **Values:** 0x0 (INACTIVE): No Endpoint Disabled Interrupt 0x1 (ACTIVE): Endpoint Disabled Interrupt
[0]	**XferCompl**	**Transfer Completed Interrupt** Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is set. **Values:** 0x0 (INACTIVE): No Transfer Complete Interrupt 0x1 (ACTIVE): Transfer Complete Interrupt

DIEPDMA2 Register (DIEPDMA2)

Register	Offset	R/W	Description	Reset Value
DIEPDMA2	HSUSBD_BA+0x954	R/W	Device IN Endpoint 2 DMA Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMAAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMAAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMAAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMAAddr							

Bits	Description	
[31:0]	**DMAAddr**	**Holds the start address of the external memory for storing or fetching endpoint data** **Note: For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten.** **This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address.** **This field indicates the base pointer for the descriptor list.**

DTXFSTS2 Register (DTXFSTS2)

Register	Offset	R/W	Description	Reset Value
DTXFSTS2	HSUSBD_BA+0x958	R	Device IN Endpoint Transmit FIFO Status Register 2	0x0000_0020

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
INEPTxFSpcAvail							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INEPTxFSpcAvail							

Bits	Description	
[31:16]	**Reserved**	**Reserved.**
[15:0]	**INEPTxFSpcAvail**	**IN Endpoint TxFIFO Space Avail** Indicates the amount of free space available in the Endpoint TxFIFO. Values are in terms of 32-bit words. 16'h0: Endpoint TxFIFO is full 16'h1: 1 word available 16'h2: 2 words available 16'hn: n words available (where 0 n 32,768) 16'h8000: 32,768 words available Others: Reserved

DIEPDMAB2 Register (DIEPDMAB2)

Register	Offset	R/W	Description	Reset Value
DIEPDMAB2	HSUSBD_BA+0x95c	R	Device IN Endpoint 2 Buffer Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMABufferAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMABufferAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMABufferAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMABufferAddr							

Bits	Description	
[31:0]	**DMABufferAddr**	**Holds the current buffer address This register is updated as and when the data transfer for the corresponding end point is in progress**

DIEPCTL3 Register (DIEPCTL3)

Register	Offset	R/W	Description	Reset Value
DIEPCTL3	HSUSBD_BA+0x960	R/W	Device Control IN Endpoint 3 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
EPEna	EPDis	Reserved		SNAK	CNAK	TxFNum	
23	22	21	20	19	18	17	16
TxFNum		Stall	Reserved	EPType		NAKSts	DPID
15	14	13	12	11	10	9	8
USBActEP	Reserved				MPS		
7	6	5	4	3	2	1	0
MPS							

Bits	Description	
[31]	**EPEna**	**Endpoint Enable** Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled, For IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. For OUT endpoint it indicates that the descriptor structure and data buffer to receive data is setup. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done Endpoint Disabled Transfer Completed **Note:** For control endpoints in DMA mode, this bit must be set to be able to transfer SETUP data packets in memory. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Enable Endpoint
[30]	**EPDis**	**Endpoint Disable** Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Disable Endpoint
[29:28]		**Reserved.**
[27]	**SNAK**	**Set NAK** A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also Set this bit for an endpoint after a SETUP packet is received on that endpoint. **Values:**

		0x0 (INACTIVE): No Set NAK 0x1 (ACTIVE): Set NAK
[26]	**CNAK**	**Clear NAK** A write to this bit clears the NAK bit for the endpoint. **Values:** 0x0 (INACTIVE): No Clear NAK 0x1 (ACTIVE): Clear NAK
[25:22]	**TxFNum**	**TxFIFO Number** Dedicated FIFO Operation: These bits specify the FIFO number associated with this endpoint. Each active IN endpoint must be programmed to a separate FIFO number. This field is valid only for IN endpoints. **Values:** 0x0 (TXFIFO0): Tx FIFO 0 0x1 (TXFIFO1): Tx FIFO 1 0x2 (TXFIFO2): Tx FIFO 2 0x3 (TXFIFO3): Tx FIFO 3 0x4 (TXFIFO4): Tx FIFO 4 0x5 (TXFIFO5): Tx FIFO 5 0x6 (TXFIFO6): Tx FIFO 6 0x7 (TXFIFO7): Tx FIFO 7 0x8 (TXFIFO8): Tx FIFO 8
[21]	**Stall**	**STALL Handshake** Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core. Applies to control endpoints only. The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (INACTIVE): STALL All non-active tokens 0x1 (ACTIVE): STALL All Active Tokens
[20]	**Reserved**	**Reserved.**
[19:18]	**EPType**	**Endpoint Type** This is the transfer type supported by this logical endpoint 2'b00: Control 2'b01: Isochronous 2'b10: Bulk 2'b11: Interrupt **Values:** 0x0 (CONTROL): Control 0x1 (ISOCHRONOUS): Isochronous 0x2 (BULK): Bulk 0x3 (INTERRUP): Interrupt
[17]	**NAKSts**	**NAK Status (Read Only)** Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status.

		1'b1: The core is transmitting NAK handshakes on this endpoint. When either the application or the core sets this bit: The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet. For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the TxFIFO. For isochronous IN endpoints: The core sends out a zero-length data packet, even if there data is available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (NONNAK): The core is transmitting non-NAK handshakes based on the FIFO status 0x1 (NAK): The core is transmitting NAK handshakes on this endpoint
[16]	**DPID**	**Endpoint Data PID (Read Only)** Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. The applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID. 1'b0: DATA0 1'b1: DATA1 **Values:** 0x0 (DATA0EVENFRM): DATA0 or Even Frame 0x1 (DATA1ODDFRM): DATA1 or Odd Frame
[15]	**USBActEP**	**USB Active Endpoint** Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints (other than EP 0) after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit. **Values:** 0x0 (DISABLED): Not Active 0x1 (ENABLED): USB Active Endpoint
[14:11]	**Reserved**	**Reserved.**
[10:0]	**MPS**	**Maximum Packet Size** The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.

DIEPINT3 Register (DIEPINT3)

Register	Offset	R/W	Description	Reset Value
DIEPINT3	HSUSBD_BA+0x968	R/W	Device IN Endpoint 3 Interrupt Register	0x0000_0080

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	NYETIntrpt	NAKIntrpt	BbleErr	Reserved		BNAIntr	TxfifoUndrn
7	6	5	4	3	2	1	0
TxFEmp	INEPNakEff	INTknEPMis	INTknTXFEmp	Reserved	AHBErr	EPDisbld	XferCompl

Bits	Description	
[31:15]	**Reserved**	**Reserved.**
[14]	**NYETIntrpt**	**NYET Interrupt** The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint. **Values:** 0x0 (INACTIVE): No NYET interrupt 0x1 (ACTIVE): NYET Interrupt
[13]	**NAKIntrpt**	**NAK Interrupt** The core generates this interrupt when a NAK is transmitted or received by the device. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo. **Values:** 0x0 (INACTIVE): No NAK interrupt 0x1 (ACTIVE): NAK Interrupt
[12]	**BbleErr**	**NAK Interrupt** The core generates this interrupt when babble is received for the endpoint. **Values:** 0x0 (INACTIVE): No interrupt 0x1 (ACTIVE): BbleErr interrupt
[11:10]	**Reserved**	**Reserved.**
[9]	**BNAIntr**	**Buffer Not Available Interrupt** The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as Host busy or DMA done. **Values:** 0x0 (INACTIVE): No BNA interrupt 0x1 (ACTIVE): BNA interrupt
[8]	**TxfifoUndrn**	**Fifo Underrun** Applies to IN endpoints Only

		This bit is valid only If thresholding is enabled. The core generates this interrupt when it detects a transmit FIFO underrun condition for this endpoint. **Values:** 0x0 (INACTIVE): No Tx FIFO Underrun interrupt 0x1 (ACTIVE): TxFIFO Underrun interrupt
[7]	**TxFEmp**	**Transmit FIFO Empty (Read Only)** This bit is valid only for IN endpoints This interrupt is asserted when the TxFIFO for this endpoint is either half or completely empty. The half or completely empty status is determined by the TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.NPTxFEmpLvl)). **Values:** 0x0 (INACTIVE): No Transmit FIFO Empty interrupt 0x1 (ACTIVE): Transmit FIFO Empty interrupt
[6]	**INEPNakEff**	**IN Endpoint NAK Effective** Applies to periodic IN endpoints only. This bit can be cleared when the application clears the IN endpoint NAK by writing to DIEPCTLn.CNAK. This interrupt indicates that the core has sampled the NAK bit Set (either by the application or by the core). The interrupt indicates that the IN endpoint NAK bit Set by the application has taken effect in the core. This interrupt does not guarantee that a NAK handshake is sent on the USB. A STALL bit takes priority over a NAK bit. **Values:** 0x0 (INACTIVE): No Endpoint NAK Effective interrupt 0x1 (ACTIVE): IN Endpoint NAK Effective interrupt
[5]	**INTknEPMis**	**IN Token Received with EP Mismatch** Applies to non-periodic IN endpoints only. Indicates that the data in the top of the non-periodic TxFIFO belongs to an endpoint other than the one for which the IN token was received. This interrupt is asserted on the endpoint for which the IN token was received. **Values:** 0x0 (INACTIVE): No IN Token Received with EP Mismatch interrupt 0x1 (ACTIVE): IN Token Received with EP Mismatch interrupt
[4]	**INTknTXFEmp**	**IN Token Received When TxFIFO is Empty** Applies to non-periodic IN endpoints only. Indicates that an IN token was received when the associated TxFIFO (periodic/non-periodic) was empty. This interrupt is asserted on the endpoint for which the IN token was received. **Values:** 0x0 (INACTIVE): No IN Token Received interrupt 0x1 (ACTIVE): IN Token Received Interrupt
[3]	**Reserved**	**Reserved.**
[2]	**AHBErr**	**AHB Error** Applies to IN and OUT endpoints. When there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address. **Values:** 0x0 (INACTIVE): No AHB Error Interrupt 0x1 (ACTIVE): AHB Error interrupt
[1]	**EPDisbld**	**Endpoint Disabled Interrupt**

		Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request. **Values:** 0x0 (INACTIVE): No Endpoint Disabled Interrupt 0x1 (ACTIVE): Endpoint Disabled Interrupt
[0]	**XferCompl**	**Transfer Completed Interrupt** Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is set. **Values:** 0x0 (INACTIVE): No Transfer Complete Interrupt 0x1 (ACTIVE): Transfer Complete Interrupt

DIEPDMA3 Register (DIEPDMA3)

Register	Offset	R/W	Description	Reset Value
DIEPDMA3	HSUSBD_BA+0x974	R/W	Device IN Endpoint 3 DMA Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMAAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMAAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMAAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMAAddr							

Bits	Description	
[31:0]	**DMAAddr**	**Holds the start address of the external memory for storing or fetching endpoint data** **Note:** For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address. This field indicates the base pointer for the descriptor list.

DTXFSTS3 Register (DTXFSTS3)

Register	Offset	R/W	Description	Reset Value
DTXFSTS3	HSUSBD_BA+0x978	R	Device IN Endpoint Transmit FIFO Status Register 3	0x0000_0020

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
INEPTxFSpcAvail							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INEPTxFSpcAvail							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**INEPTxFSpcAvail**	**IN Endpoint TxFIFO Space Avail** Indicates the amount of free space available in the Endpoint TxFIFO. Values are in terms of 32-bit words. 16'h0: Endpoint TxFIFO is full 16'h1: 1 word available 16'h2: 2 words available 16'hn: n words available (where 0 n 32,768) 16'h8000: 32,768 words available Others: Reserved

DIEPDMAB3 Register (DIEPDMAB3)

Register	Offset	R/W	Description	Reset Value
DIEPDMAB3	HSUSBD_BA+0x97c	R	Device IN Endpoint 3 Buffer Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMABufferAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMABufferAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMABufferAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMABufferAddr							

Bits	Description	
[31:0]	**DMABufferAddr**	**Holds the current buffer address This register is updated as and when the data transfer for the corresponding end point is in progress**

DIEPCTL4 Register (DIEPCTL4)

Register	Offset	R/W	Description	Reset Value
DIEPCTL4	HSUSBD_BA+0x980	R/W	Device Control IN Endpoint 4 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
EPEna	EPDis	Reserved		SNAK	CNAK	TxFNum	
23	22	21	20	19	18	17	16
TxFNum		Stall	Reserved	EPType		NAKSts	DPID
15	14	13	12	11	10	9	8
USBActEP	Reserved				MPS		
7	6	5	4	3	2	1	0
MPS							

Bits	Description	
[31]	**EPEna**	**Endpoint Enable** Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled, For IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. For OUT endpoint it indicates that the descriptor structure and data buffer to receive data is setup. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done Endpoint Disabled Transfer Completed **Note:** For control endpoints in DMA mode, this bit must be set to be able to transfer SETUP data packets in memory. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Enable Endpoint
[30]	**EPDis**	**Endpoint Disable** Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Disable Endpoint
[29:28]		**Reserved.**
[27]	**SNAK**	**Set NAK** A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also Set this bit for an endpoint after a SETUP packet is received on that endpoint. **Values:**

		0x0 (INACTIVE): No Set NAK 0x1 (ACTIVE): Set NAK
[26]	**CNAK**	**Clear NAK** A write to this bit clears the NAK bit for the endpoint. **Values:** 0x0 (INACTIVE): No Clear NAK 0x1 (ACTIVE): Clear NAK
[25:22]	**TxFNum**	**TxFIFO Number** Dedicated FIFO Operation: These bits specify the FIFO number associated with this endpoint. Each active IN endpoint must be programmed to a separate FIFO number. This field is valid only for IN endpoints. **Values:** 0x0 (TXFIFO0): Tx FIFO 0 0x1 (TXFIFO1): Tx FIFO 1 0x2 (TXFIFO2): Tx FIFO 2 0x3 (TXFIFO3): Tx FIFO 3 0x4 (TXFIFO4): Tx FIFO 4 0x5 (TXFIFO5): Tx FIFO 5 0x6 (TXFIFO6): Tx FIFO 6 0x7 (TXFIFO7): Tx FIFO 7 0x8 (TXFIFO8): Tx FIFO 8
[21]	**Stall**	**STALL Handshake** Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core. Applies to control endpoints only. The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (INACTIVE): STALL All non-active tokens 0x1 (ACTIVE): STALL All Active Tokens
[20]	**Reserved**	**Reserved.**
[19:18]	**EPType**	**Endpoint Type** This is the transfer type supported by this logical endpoint 2'b00: Control 2'b01: Isochronous 2'b10: Bulk 2'b11: Interrupt **Values:** 0x0 (CONTROL): Control 0x1 (ISOCHRONOUS): Isochronous 0x2 (BULK): Bulk 0x3 (INTERRUP): Interrupt
[17]	**NAKSts**	**NAK Status (Read Only)** Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status.

		1'b1: The core is transmitting NAK handshakes on this endpoint. When either the application or the core sets this bit: The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet. For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the TxFIFO. For isochronous IN endpoints: The core sends out a zero-length data packet, even if there data is available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (NONNAK): The core is transmitting non-NAK handshakes based on the FIFO status 0x1 (NAK): The core is transmitting NAK handshakes on this endpoint
[16]	**DPID**	**Endpoint Data PID (Read Only)** Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. The applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID. 1'b0: DATA0 1'b1: DATA1 **Values:** 0x0 (DATA0EVENFRM): DATA0 or Even Frame 0x1 (DATA1ODDFRM): DATA1 or Odd Frame
[15]	**USBActEP**	**USB Active Endpoint** Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints (other than EP 0) after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit. **Values:** 0x0 (DISABLED): Not Active 0x1 (ENABLED): USB Active Endpoint
[14:11]	**Reserved**	**Reserved.**
[10:0]	**MPS**	**Maximum Packet Size** The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.

DIEPINT4 Register (DIEPINT4)

Register	Offset	R/W	Description	Reset Value
DIEPINT4	HSUSBD_BA+0x988	R/W	Device IN Endpoint 4 Interrupt Register	0x0000_0080

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	NYETIntrpt	NAKIntrpt	BbleErr	Reserved		BNAIntr	TxfifoUndrn
7	6	5	4	3	2	1	0
TxFEmp	INEPNakEff	INTknEPMis	INTknTXFEmp	Reserved	AHBErr	EPDisbld	XferCompl

Bits	Description	
[31:15]	**Reserved**	**Reserved.**
[14]	**NYETIntrpt**	**NYET Interrupt** The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint. **Values:** 0x0 (INACTIVE): No NYET interrupt 0x1 (ACTIVE): NYET Interrupt
[13]	**NAKIntrpt**	**NAK Interrupt** The core generates this interrupt when a NAK is transmitted or received by the device. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo. **Values:** 0x0 (INACTIVE): No NAK interrupt 0x1 (ACTIVE): NAK Interrupt
[12]	**BbleErr**	**NAK Interrupt** The core generates this interrupt when babble is received for the endpoint. **Values:** 0x0 (INACTIVE): No interrupt 0x1 (ACTIVE): BbleErr interrupt
[11:10]	**Reserved**	**Reserved.**
[9]	**BNAIntr**	**Buffer Not Available Interrupt** The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as Host busy or DMA done. **Values:** 0x0 (INACTIVE): No BNA interrupt 0x1 (ACTIVE): BNA interrupt
[8]	**TxfifoUndrn**	**Fifo Underrun** Applies to IN endpoints Only

		This bit is valid only If thresholding is enabled. The core generates this interrupt when it detects a transmit FIFO underrun condition for this endpoint. **Values:** 0x0 (INACTIVE): No Tx FIFO Underrun interrupt 0x1 (ACTIVE): TxFIFO Underrun interrupt
[7]	**TxFEmp**	**Transmit FIFO Empty (Read Only)** This bit is valid only for IN endpoints This interrupt is asserted when the TxFIFO for this endpoint is either half or completely empty. The half or completely empty status is determined by the TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.NPTxFEmpLvl)). **Values:** 0x0 (INACTIVE): No Transmit FIFO Empty interrupt 0x1 (ACTIVE): Transmit FIFO Empty interrupt
[6]	**INEPNakEff**	**IN Endpoint NAK Effective** Applies to periodic IN endpoints only. This bit can be cleared when the application clears the IN endpoint NAK by writing to DIEPCTLn.CNAK. This interrupt indicates that the core has sampled the NAK bit Set (either by the application or by the core). The interrupt indicates that the IN endpoint NAK bit Set by the application has taken effect in the core. This interrupt does not guarantee that a NAK handshake is sent on the USB. A STALL bit takes priority over a NAK bit. **Values:** 0x0 (INACTIVE): No Endpoint NAK Effective interrupt 0x1 (ACTIVE): IN Endpoint NAK Effective interrupt
[5]	**INTknEPMis**	**IN Token Received with EP Mismatch** Applies to non-periodic IN endpoints only. Indicates that the data in the top of the non-periodic TxFIFO belongs to an endpoint other than the one for which the IN token was received. This interrupt is asserted on the endpoint for which the IN token was received. **Values:** 0x0 (INACTIVE): No IN Token Received with EP Mismatch interrupt 0x1 (ACTIVE): IN Token Received with EP Mismatch interrupt
[4]	**INTknTXFEmp**	**IN Token Received When TxFIFO is Empty** Applies to non-periodic IN endpoints only. Indicates that an IN token was received when the associated TxFIFO (periodic/non-periodic) was empty. This interrupt is asserted on the endpoint for which the IN token was received. **Values:** 0x0 (INACTIVE): No IN Token Received interrupt 0x1 (ACTIVE): IN Token Received Interrupt
[3]	**Reserved**	**Reserved.**
[2]	**AHBErr**	**AHB Error** Applies to IN and OUT endpoints. When there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address. **Values:** 0x0 (INACTIVE): No AHB Error Interrupt 0x1 (ACTIVE): AHB Error interrupt
[1]	**EPDisbld**	**Endpoint Disabled Interrupt**

		Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request. **Values:** 0x0 (INACTIVE): No Endpoint Disabled Interrupt 0x1 (ACTIVE): Endpoint Disabled Interrupt
[0]	**XferCompl**	**Transfer Completed Interrupt** Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is set. **Values:** 0x0 (INACTIVE): No Transfer Complete Interrupt 0x1 (ACTIVE): Transfer Complete Interrupt

DIEPDMA4 Register (DIEPDMA4)

Register	Offset	R/W	Description	Reset Value
DIEPDMA4	HSUSBD_BA+0x994	R/W	Device IN Endpoint 4 DMA Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMAAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMAAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMAAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMAAddr							

Bits	Description	
[31:0]	**DMAAddr**	**Holds the start address of the external memory for storing or fetching endpoint data** **Note:** For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address. This field indicates the base pointer for the descriptor list.

DTXFSTS4 Register (DTXFSTS4)

Register	Offset	R/W	Description	Reset Value
DTXFSTS4	HSUSBD_BA+0x998	R	Device IN Endpoint Transmit FIFO Status Register 4	0x0000_0020

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
INEPTxFSpcAvail							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INEPTxFSpcAvail							

Bits	Description	
[31:16]	**Reserved**	**Reserved.**
[15:0]	**INEPTxFSpcAvail**	**IN Endpoint TxFIFO Space Avail** Indicates the amount of free space available in the Endpoint TxFIFO. Values are in terms of 32-bit words. 16'h0: Endpoint TxFIFO is full 16'h1: 1 word available 16'h2: 2 words available 16'hn: n words available (where 0 n 32,768) 16'h8000: 32,768 words available Others: Reserved

DIEPDMAB4 Register (DIEPDMAB4)

Register	Offset	R/W	Description	Reset Value
DIEPDMAB4	HSUSBD_BA+0x99c	R	Device IN Endpoint 4 Buffer Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMABufferAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMABufferAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMABufferAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMABufferAddr							

Bits	Description	
[31:0]	**DMABufferAddr**	**Holds the current buffer address This register is updated as and when the data transfer for the corresponding end point is in progress**

DIEPCTL5 Register (DIEPCTL5)

Register	Offset	R/W	Description	Reset Value
DIEPCTL5	HSUSBD_BA+0x9a0	R/W	Device Control IN Endpoint 5 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
EPEna	EPDis	Reserved		SNAK	CNAK	TxFNum	
23	22	21	20	19	18	17	16
TxFNum		Stall	Reserved	EPType		NAKSts	DPID
15	14	13	12	11	10	9	8
USBActEP	Reserved				MPS		
7	6	5	4	3	2	1	0
MPS							

Bits	Description	
[31]	**EPEna**	**Endpoint Enable** Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled, For IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. For OUT endpoint it indicates that the descriptor structure and data buffer to receive data is setup. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done Endpoint Disabled Transfer Completed **Note:** For control endpoints in DMA mode, this bit must be set to be able to transfer SETUP data packets in memory. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Enable Endpoint
[30]	**EPDis**	**Endpoint Disable** Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Disable Endpoint
[29:28]	**Reserved**	**Reserved.**
[27]	**SNAK**	**Set NAK** A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also Set this bit for an endpoint after a SETUP packet is received on that endpoint. **Values:**

		0x0 (INACTIVE): No Set NAK 0x1 (ACTIVE): Set NAK
[26]	**CNAK**	**Clear NAK** A write to this bit clears the NAK bit for the endpoint. **Values:** 0x0 (INACTIVE): No Clear NAK 0x1 (ACTIVE): Clear NAK
[25:22]	**TxFNum**	**TxFIFO Number** Dedicated FIFO Operation: These bits specify the FIFO number associated with this endpoint. Each active IN endpoint must be programmed to a separate FIFO number. This field is valid only for IN endpoints. **Values:** 0x0 (TXFIFO0): Tx FIFO 0 0x1 (TXFIFO1): Tx FIFO 1 0x2 (TXFIFO2): Tx FIFO 2 0x3 (TXFIFO3): Tx FIFO 3 0x4 (TXFIFO4): Tx FIFO 4 0x5 (TXFIFO5): Tx FIFO 5 0x6 (TXFIFO6): Tx FIFO 6 0x7 (TXFIFO7): Tx FIFO 7 0x8 (TXFIFO8): Tx FIFO 8
[21]	**Stall**	**STALL Handshake** Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core. Applies to control endpoints only. The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (INACTIVE): STALL All non-active tokens 0x1 (ACTIVE): STALL All Active Tokens
[20]	**Reserved**	**Reserved.**
[19:18]	**EPType**	**Endpoint Type** This is the transfer type supported by this logical endpoint 2'b00: Control 2'b01: Isochronous 2'b10: Bulk 2'b11: Interrupt **Values:** 0x0 (CONTROL): Control 0x1 (ISOCHRONOUS): Isochronous 0x2 (BULK): Bulk 0x3 (INTERRUP): Interrupt
[17]	**NAKSts**	**NAK Status (Read Only)** Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status.

		1'b1: The core is transmitting NAK handshakes on this endpoint. When either the application or the core sets this bit: The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet. For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the TxFIFO. For isochronous IN endpoints: The core sends out a zero-length data packet, even if there data is available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (NONNAK): The core is transmitting non-NAK handshakes based on the FIFO status 0x1 (NAK): The core is transmitting NAK handshakes on this endpoint
[16]	**DPID**	**Endpoint Data PID (Read Only)** Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. The applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID. 1'b0: DATA0 1'b1: DATA1 **Values:** 0x0 (DATA0EVENFRM): DATA0 or Even Frame 0x1 (DATA1ODDFRM): DATA1 or Odd Frame
[15]	**USBActEP**	**USB Active Endpoint** Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints (other than EP 0) after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit. **Values:** 0x0 (DISABLED): Not Active 0x1 (ENABLED): USB Active Endpoint
[14:11]	**Reserved**	**Reserved.**
[10:0]	**MPS**	**Maximum Packet Size** The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.

DIEPINT5 Register (DIEPINT5)

Register	Offset	R/W	Description	Reset Value
DIEPINT5	HSUSBD_BA+0x9a8	R/W	Device IN Endpoint 5 Interrupt Register	0x0000_0080

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	NYETIntrpt	NAKIntrpt	BbleErr	Reserved		BNAIntr	TxfifoUndrn
7	6	5	4	3	2	1	0
TxFEmp	INEPNakEff	INTknEPMis	INTknTXFEmp	Reserved	AHBErr	EPDisbld	XferCompl

Bits	Description	
[31:15]	**Reserved**	**Reserved.**
[14]	**NYETIntrpt**	**NYET Interrupt** The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint. **Values:** 0x0 (INACTIVE): No NYET interrupt 0x1 (ACTIVE): NYET Interrupt
[13]	**NAKIntrpt**	**NAK Interrupt** The core generates this interrupt when a NAK is transmitted or received by the device. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo. **Values:** 0x0 (INACTIVE): No NAK interrupt 0x1 (ACTIVE): NAK Interrupt
[12]	**BbleErr**	**NAK Interrupt** The core generates this interrupt when babble is received for the endpoint. **Values:** 0x0 (INACTIVE): No interrupt 0x1 (ACTIVE): BbleErr interrupt
[11:10]	**Reserved**	**Reserved.**
[9]	**BNAIntr**	**Buffer Not Available Interrupt** The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as Host busy or DMA done. **Values:** 0x0 (INACTIVE): No BNA interrupt 0x1 (ACTIVE): BNA interrupt
[8]	**TxfifoUndrn**	**Fifo Underrun** Applies to IN endpoints Only

		This bit is valid only If thresholding is enabled. The core generates this interrupt when it detects a transmit FIFO underrun condition for this endpoint. **Values:** 0x0 (INACTIVE): No Tx FIFO Underrun interrupt 0x1 (ACTIVE): TxFIFO Underrun interrupt
[7]	**TxFEmp**	**Transmit FIFO Empty (Read Only)** This bit is valid only for IN endpoints This interrupt is asserted when the TxFIFO for this endpoint is either half or completely empty. The half or completely empty status is determined by the TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.NPTxFEmpLvl)). **Values:** 0x0 (INACTIVE): No Transmit FIFO Empty interrupt 0x1 (ACTIVE): Transmit FIFO Empty interrupt
[6]	**INEPNakEff**	**IN Endpoint NAK Effective** Applies to periodic IN endpoints only. This bit can be cleared when the application clears the IN endpoint NAK by writing to DIEPCTLn.CNAK. This interrupt indicates that the core has sampled the NAK bit Set (either by the application or by the core). The interrupt indicates that the IN endpoint NAK bit Set by the application has taken effect in the core. This interrupt does not guarantee that a NAK handshake is sent on the USB. A STALL bit takes priority over a NAK bit. **Values:** 0x0 (INACTIVE): No Endpoint NAK Effective interrupt 0x1 (ACTIVE): IN Endpoint NAK Effective interrupt
[5]	**INTknEPMis**	**IN Token Received with EP Mismatch** Applies to non-periodic IN endpoints only. Indicates that the data in the top of the non-periodic TxFIFO belongs to an endpoint other than the one for which the IN token was received. This interrupt is asserted on the endpoint for which the IN token was received. **Values:** 0x0 (INACTIVE): No IN Token Received with EP Mismatch interrupt 0x1 (ACTIVE): IN Token Received with EP Mismatch interrupt
[4]	**INTknTXFEmp**	**IN Token Received When TxFIFO is Empty** Applies to non-periodic IN endpoints only. Indicates that an IN token was received when the associated TxFIFO (periodic/non-periodic) was empty. This interrupt is asserted on the endpoint for which the IN token was received. **Values:** 0x0 (INACTIVE): No IN Token Received interrupt 0x1 (ACTIVE): IN Token Received Interrupt
[3]	**Reserved**	**Reserved.**
[2]	**AHBErr**	**AHB Error** Applies to IN and OUT endpoints. When there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address. **Values:** 0x0 (INACTIVE): No AHB Error Interrupt 0x1 (ACTIVE): AHB Error interrupt
[1]	**EPDisbld**	**Endpoint Disabled Interrupt**

		Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request. **Values:** 0x0 (INACTIVE): No Endpoint Disabled Interrupt 0x1 (ACTIVE): Endpoint Disabled Interrupt
[0]	**XferCompl**	**Transfer Completed Interrupt** Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is set. **Values:** 0x0 (INACTIVE): No Transfer Complete Interrupt 0x1 (ACTIVE): Transfer Complete Interrupt

DIEPDMA5 Register (DIEPDMA5)

Register	Offset	R/W	Description	Reset Value
DIEPDMA5	HSUSBD_BA+0x9b4	R/W	Device IN Endpoint 5 DMA Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMAAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMAAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMAAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMAAddr							

Bits	Description	
[31:0]	**DMAAddr**	**Holds the start address of the external memory for storing or fetching endpoint data** **Note:** For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address. This field indicates the base pointer for the descriptor list.

DTXFSTS5 Register (DTXFSTS5)

Register	Offset	R/W	Description	Reset Value
DTXFSTS5	HSUSBD_BA+0x9b8	R	Device IN Endpoint Transmit FIFO Status Register 5	0x0000_0020

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
INEPTxFSpcAvail							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INEPTxFSpcAvail							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**INEPTxFSpcAvail**	**IN Endpoint TxFIFO Space Avail** Indicates the amount of free space available in the Endpoint TxFIFO. Values are in terms of 32-bit words. 16'h0: Endpoint TxFIFO is full 16'h1: 1 word available 16'h2: 2 words available 16'hn: n words available (where 0 n 32,768) 16'h8000: 32,768 words available Others: Reserved

DIEPDMAB5 Register (DIEPDMAB5)

Register	Offset	R/W	Description	Reset Value
DIEPDMAB5	HSUSBD_BA+0x9bc	R	Device IN Endpoint 5 Buffer Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMABufferAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMABufferAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMABufferAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMABufferAddr							

Bits	Description	
[31:0]	**DMABufferAddr**	**Holds the current buffer address This register is updated as and when the data transfer for the corresponding end point is in progress**

DIEPCTL6 Register (DIEPCTL6)

Register	Offset	R/W	Description	Reset Value
DIEPCTL6	HSUSBD_BA+0x9c0	R/W	Device Control IN Endpoint 6 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
EPEna	EPDis	Reserved		SNAK	CNAK	TxFNum	
23	22	21	20	19	18	17	16
TxFNum		Stall	Reserved	EPType		NAKSts	DPID
15	14	13	12	11	10	9	8
USBActEP	Reserved				MPS		
7	6	5	4	3	2	1	0
MPS							

Bits	Description	
[31]	**EPEna**	**Endpoint Enable** Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled, For IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. For OUT endpoint it indicates that the descriptor structure and data buffer to receive data is setup. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done Endpoint Disabled Transfer Completed **Note:** For control endpoints in DMA mode, this bit must be set to be able to transfer SETUP data packets in memory. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Enable Endpoint
[30]	**EPDis**	**Endpoint Disable** Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Disable Endpoint
[29:28]	**Reserved**	**Reserved.**
[27]	**SNAK**	**Set NAK** A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also Set this bit for an endpoint after a SETUP packet is received on that endpoint. **Values:**

		0x0 (INACTIVE): No Set NAK 0x1 (ACTIVE): Set NAK
[26]	**CNAK**	**Clear NAK** A write to this bit clears the NAK bit for the endpoint. **Values:** 0x0 (INACTIVE): No Clear NAK 0x1 (ACTIVE): Clear NAK
[25:22]	**TxFNum**	**TxFIFO Number** Dedicated FIFO Operation: These bits specify the FIFO number associated with this endpoint. Each active IN endpoint must be programmed to a separate FIFO number. This field is valid only for IN endpoints. **Values:** 0x0 (TXFIFO0): Tx FIFO 0 0x1 (TXFIFO1): Tx FIFO 1 0x2 (TXFIFO2): Tx FIFO 2 0x3 (TXFIFO3): Tx FIFO 3 0x4 (TXFIFO4): Tx FIFO 4 0x5 (TXFIFO5): Tx FIFO 5 0x6 (TXFIFO6): Tx FIFO 6 0x7 (TXFIFO7): Tx FIFO 7 0x8 (TXFIFO8): Tx FIFO 8
[21]	**Stall**	**STALL Handshake** Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core. Applies to control endpoints only. The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (INACTIVE): STALL All non-active tokens 0x1 (ACTIVE): STALL All Active Tokens
[20]	**Reserved**	**Reserved.**
[19:18]	**EPType**	**Endpoint Type** This is the transfer type supported by this logical endpoint 2'b00: Control 2'b01: Isochronous 2'b10: Bulk 2'b11: Interrupt **Values:** 0x0 (CONTROL): Control 0x1 (ISOCHRONOUS): Isochronous 0x2 (BULK): Bulk 0x3 (INTERRUP): Interrupt
[17]	**NAKSts**	**NAK Status (Read Only)** Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status.

		1'b1: The core is transmitting NAK handshakes on this endpoint. When either the application or the core sets this bit: The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet. For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the TxFIFO. For isochronous IN endpoints: The core sends out a zero-length data packet, even if there data is available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (NONNAK): The core is transmitting non-NAK handshakes based on the FIFO status 0x1 (NAK): The core is transmitting NAK handshakes on this endpoint
[16]	**DPID**	**Endpoint Data PID (Read Only)** Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. The applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID. 1'b0: DATA0 1'b1: DATA1 **Values:** 0x0 (DATA0EVENFRM): DATA0 or Even Frame 0x1 (DATA1ODDFRM): DATA1 or Odd Frame
[15]	**USBActEP**	**USB Active Endpoint** Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints (other than EP 0) after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit. **Values:** 0x0 (DISABLED): Not Active 0x1 (ENABLED): USB Active Endpoint
[14:11]	**Reserved**	**Reserved.**
[10:0]	**MPS**	**Maximum Packet Size** The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.

DIEPINT6 Register (DIEPINT6)

Register	Offset	R/W	Description	Reset Value
DIEPINT6	HSUSBD_BA+0x9c8	R/W	Device IN Endpoint 6 Interrupt Register	0x0000_0080

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	NYETIntrpt	NAKIntrpt	BbleErr	Reserved		BNAIntr	TxfifoUndrn
7	6	5	4	3	2	1	0
TxFEmp	INEPNakEff	INTknEPMis	INTknTXFEmp	Reserved	AHBErr	EPDisbld	XferCompl

Bits	Description	
[31:15]	**Reserved**	**Reserved.**
[14]	**NYETIntrpt**	**NYET Interrupt** The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint. **Values:** 0x0 (INACTIVE): No NYET interrupt 0x1 (ACTIVE): NYET Interrupt
[13]	**NAKIntrpt**	**NAK Interrupt** The core generates this interrupt when a NAK is transmitted or received by the device. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo. **Values:** 0x0 (INACTIVE): No NAK interrupt 0x1 (ACTIVE): NAK Interrupt
[12]	**BbleErr**	**NAK Interrupt** The core generates this interrupt when babble is received for the endpoint. **Values:** 0x0 (INACTIVE): No interrupt 0x1 (ACTIVE): BbleErr interrupt
[11:10]	**Reserved**	**Reserved.**
[9]	**BNAIntr**	**Buffer Not Available Interrupt** The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as Host busy or DMA done. **Values:** 0x0 (INACTIVE): No BNA interrupt 0x1 (ACTIVE): BNA interrupt
[8]	**TxfifoUndrn**	**Fifo Underrun** Applies to IN endpoints Only

		This bit is valid only If thresholding is enabled. The core generates this interrupt when it detects a transmit FIFO underrun condition for this endpoint. **Values:** 0x0 (INACTIVE): No Tx FIFO Underrun interrupt 0x1 (ACTIVE): TxFIFO Underrun interrupt
[7]	**TxFEmp**	**Transmit FIFO Empty (Read Only)** This bit is valid only for IN endpoints This interrupt is asserted when the TxFIFO for this endpoint is either half or completely empty. The half or completely empty status is determined by the TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.NPTxFEmpLvl)). **Values:** 0x0 (INACTIVE): No Transmit FIFO Empty interrupt 0x1 (ACTIVE): Transmit FIFO Empty interrupt
[6]	**INEPNakEff**	**IN Endpoint NAK Effective** Applies to periodic IN endpoints only. This bit can be cleared when the application clears the IN endpoint NAK by writing to DIEPCTLn.CNAK. This interrupt indicates that the core has sampled the NAK bit Set (either by the application or by the core). The interrupt indicates that the IN endpoint NAK bit Set by the application has taken effect in the core. This interrupt does not guarantee that a NAK handshake is sent on the USB. A STALL bit takes priority over a NAK bit. **Values:** 0x0 (INACTIVE): No Endpoint NAK Effective interrupt 0x1 (ACTIVE): IN Endpoint NAK Effective interrupt
[5]	**INTknEPMis**	**IN Token Received with EP Mismatch** Applies to non-periodic IN endpoints only. Indicates that the data in the top of the non-periodic TxFIFO belongs to an endpoint other than the one for which the IN token was received. This interrupt is asserted on the endpoint for which the IN token was received. **Values:** 0x0 (INACTIVE): No IN Token Received with EP Mismatch interrupt 0x1 (ACTIVE): IN Token Received with EP Mismatch interrupt
[4]	**INTknTXFEmp**	**IN Token Received When TxFIFO is Empty** Applies to non-periodic IN endpoints only. Indicates that an IN token was received when the associated TxFIFO (periodic/non-periodic) was empty. This interrupt is asserted on the endpoint for which the IN token was received. **Values:** 0x0 (INACTIVE): No IN Token Received interrupt 0x1 (ACTIVE): IN Token Received Interrupt
[3]	**Reserved**	**Reserved.**
[2]	**AHBErr**	**AHB Error** Applies to IN and OUT endpoints. When there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address. **Values:** 0x0 (INACTIVE): No AHB Error Interrupt 0x1 (ACTIVE): AHB Error interrupt
[1]	**EPDisbld**	**Endpoint Disabled Interrupt**

		Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request. **Values:** 0x0 (INACTIVE): No Endpoint Disabled Interrupt 0x1 (ACTIVE): Endpoint Disabled Interrupt
[0]	**XferCompl**	**Transfer Completed Interrupt** Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is set. **Values:** 0x0 (INACTIVE): No Transfer Complete Interrupt 0x1 (ACTIVE): Transfer Complete Interrupt

DIEPDMA6 Register (DIEPDMA6)

Register	Offset	R/W	Description	Reset Value
DIEPDMA6	HSUSBD_BA+0x9d4	R/W	Device IN Endpoint 6 DMA Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMAAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMAAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMAAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMAAddr							

Bits	Description	
[31:0]	**DMAAddr**	**Holds the start address of the external memory for storing or fetching endpoint data** **Note:** For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address. This field indicates the base pointer for the descriptor list.

DTXFSTS6 Register (DTXFSTS6)

Register	Offset	R/W	Description	Reset Value
DTXFSTS6	HSUSBD_BA+0x9d8	R	Device IN Endpoint Transmit FIFO Status Register 6	0x0000_0020

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
INEPTxFSpcAvail							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INEPTxFSpcAvail							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**INEPTxFSpcAvail**	**IN Endpoint TxFIFO Space Avail** Indicates the amount of free space available in the Endpoint TxFIFO. Values are in terms of 32-bit words. 16'h0: Endpoint TxFIFO is full 16'h1: 1 word available 16'h2: 2 words available 16'hn: n words available (where 0 n 32,768) 16'h8000: 32,768 words available Others: Reserved

DIEPDMAB6 Register (DIEPDMAB6)

Register	Offset	R/W	Description	Reset Value
DIEPDMAB6	HSUSBD_BA+0x9dc	R	Device IN Endpoint 6 Buffer Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMABufferAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMABufferAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMABufferAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMABufferAddr							

Bits	Description	
[31:0]	**DMABufferAddr**	**Holds the current buffer address This register is updated as and when the data transfer for the corresponding end point is in progress**

DIEPCTL7 Register (DIEPCTL7)

Register	Offset	R/W	Description	Reset Value
DIEPCTL7	HSUSBD_BA+0x9e0	R/W	Device Control IN Endpoint 7 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
EPEna	EPDis	Reserved		SNAK	CNAK	TxFNum	
23	22	21	20	19	18	17	16
TxFNum		Stall	Reserved	EPType		NAKSts	DPID
15	14	13	12	11	10	9	8
USBActEP	Reserved				MPS		
7	6	5	4	3	2	1	0
MPS							

Bits	Description	
[31]	**EPEna**	**Endpoint Enable** Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled, For IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. For OUT endpoint it indicates that the descriptor structure and data buffer to receive data is setup. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done Endpoint Disabled Transfer Completed **Note:** For control endpoints in DMA mode, this bit must be set to be able to transfer SETUP data packets in memory. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Enable Endpoint
[30]	**EPDis**	**Endpoint Disable** Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Disable Endpoint
[29:28]	**Reserved**	**Reserved.**
[27]	**SNAK**	**Set NAK** A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also Set this bit for an endpoint after a SETUP packet is received on that endpoint. **Values:**

		0x0 (INACTIVE): No Set NAK 0x1 (ACTIVE): Set NAK
[26]	**CNAK**	**Clear NAK** A write to this bit clears the NAK bit for the endpoint. **Values:** 0x0 (INACTIVE): No Clear NAK 0x1 (ACTIVE): Clear NAK
[25:22]	**TxFNum**	**TxFIFO Number** Dedicated FIFO Operation: These bits specify the FIFO number associated with this endpoint. Each active IN endpoint must be programmed to a separate FIFO number. This field is valid only for IN endpoints. **Values:** 0x0 (TXFIFO0): Tx FIFO 0 0x1 (TXFIFO1): Tx FIFO 1 0x2 (TXFIFO2): Tx FIFO 2 0x3 (TXFIFO3): Tx FIFO 3 0x4 (TXFIFO4): Tx FIFO 4 0x5 (TXFIFO5): Tx FIFO 5 0x6 (TXFIFO6): Tx FIFO 6 0x7 (TXFIFO7): Tx FIFO 7 0x8 (TXFIFO8): Tx FIFO 8
[21]	**Stall**	**STALL Handshake** Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core. Applies to control endpoints only. The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (INACTIVE): STALL All non-active tokens 0x1 (ACTIVE): STALL All Active Tokens
[20]	**Reserved**	**Reserved.**
[19:18]	**EPType**	**Endpoint Type** This is the transfer type supported by this logical endpoint 2'b00: Control 2'b01: Isochronous 2'b10: Bulk 2'b11: Interrupt **Values:** 0x0 (CONTROL): Control 0x1 (ISOCHRONOUS): Isochronous 0x2 (BULK): Bulk 0x3 (INTERRUP): Interrupt
[17]	**NAKSts**	**NAK Status (Read Only)** Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status.

		1'b1: The core is transmitting NAK handshakes on this endpoint. When either the application or the core sets this bit: The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet. For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the TxFIFO. For isochronous IN endpoints: The core sends out a zero-length data packet, even if there data is available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (NONNAK): The core is transmitting non-NAK handshakes based on the FIFO status 0x1 (NAK): The core is transmitting NAK handshakes on this endpoint
[16]	**DPID**	**Endpoint Data PID (Read Only)** Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. The applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID. 1'b0: DATA0 1'b1: DATA1 **Values:** 0x0 (DATA0EVENFRM): DATA0 or Even Frame 0x1 (DATA1ODDFRM): DATA1 or Odd Frame
[15]	**USBActEP**	**USB Active Endpoint** Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints (other than EP 0) after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit. **Values:** 0x0 (DISABLED): Not Active 0x1 (ENABLED): USB Active Endpoint
[14:11]	**Reserved**	**Reserved.**
[10:0]	**MPS**	**Maximum Packet Size** The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.

DIEPINT7 Register (DIEPINT7)

Register	Offset	R/W	Description	Reset Value
DIEPINT7	HSUSBD_BA+0x9e8	R/W	Device IN Endpoint 7 Interrupt Register	0x0000_0080

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	NYETIntrpt	NAKIntrpt	BbleErr	Reserved		BNAIntr	TxfifoUndrn
7	6	5	4	3	2	1	0
TxFEmp	INEPNakEff	INTknEPMis	INTknTXFEmp	Reserved	AHBErr	EPDisbld	XferCompl

Bits	Description	
[31:15]	**Reserved**	**Reserved.**
[14]	**NYETIntrpt**	**NYET Interrupt** The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint. **Values:** 0x0 (INACTIVE): No NYET interrupt 0x1 (ACTIVE): NYET Interrupt
[13]	**NAKIntrpt**	**NAK Interrupt** The core generates this interrupt when a NAK is transmitted or received by the device. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo. **Values:** 0x0 (INACTIVE): No NAK interrupt 0x1 (ACTIVE): NAK Interrupt
[12]	**BbleErr**	**NAK Interrupt** The core generates this interrupt when babble is received for the endpoint. **Values:** 0x0 (INACTIVE): No interrupt 0x1 (ACTIVE): BbleErr interrupt
[11:10]	**Reserved**	**Reserved.**
[9]	**BNAIntr**	**Buffer Not Available Interrupt** The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as Host busy or DMA done. **Values:** 0x0 (INACTIVE): No BNA interrupt 0x1 (ACTIVE): BNA interrupt
[8]	**TxfifoUndrn**	**Fifo Underrun** Applies to IN endpoints Only

		This bit is valid only If thresholding is enabled. The core generates this interrupt when it detects a transmit FIFO underrun condition for this endpoint. **Values:** 0x0 (INACTIVE): No Tx FIFO Underrun interrupt 0x1 (ACTIVE): TxFIFO Underrun interrupt
[7]	**TxFEmp**	**Transmit FIFO Empty (Read Only)** This bit is valid only for IN endpoints This interrupt is asserted when the TxFIFO for this endpoint is either half or completely empty. The half or completely empty status is determined by the TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.NPTxFEmpLvl)). **Values:** 0x0 (INACTIVE): No Transmit FIFO Empty interrupt 0x1 (ACTIVE): Transmit FIFO Empty interrupt
[6]	**INEPNakEff**	**IN Endpoint NAK Effective** Applies to periodic IN endpoints only. This bit can be cleared when the application clears the IN endpoint NAK by writing to DIEPCTLn.CNAK. This interrupt indicates that the core has sampled the NAK bit Set (either by the application or by the core). The interrupt indicates that the IN endpoint NAK bit Set by the application has taken effect in the core. This interrupt does not guarantee that a NAK handshake is sent on the USB. A STALL bit takes priority over a NAK bit. **Values:** 0x0 (INACTIVE): No Endpoint NAK Effective interrupt 0x1 (ACTIVE): IN Endpoint NAK Effective interrupt
[5]	**INTknEPMis**	**IN Token Received with EP Mismatch** Applies to non-periodic IN endpoints only. Indicates that the data in the top of the non-periodic TxFIFO belongs to an endpoint other than the one for which the IN token was received. This interrupt is asserted on the endpoint for which the IN token was received. **Values:** 0x0 (INACTIVE): No IN Token Received with EP Mismatch interrupt 0x1 (ACTIVE): IN Token Received with EP Mismatch interrupt
[4]	**INTknTXFEmp**	**IN Token Received When TxFIFO is Empty** Applies to non-periodic IN endpoints only. Indicates that an IN token was received when the associated TxFIFO (periodic/non-periodic) was empty. This interrupt is asserted on the endpoint for which the IN token was received. **Values:** 0x0 (INACTIVE): No IN Token Received interrupt 0x1 (ACTIVE): IN Token Received Interrupt
[3]	**Reserved**	**Reserved.**
[2]	**AHBErr**	**AHB Error** Applies to IN and OUT endpoints. When there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address. **Values:** 0x0 (INACTIVE): No AHB Error Interrupt 0x1 (ACTIVE): AHB Error interrupt
[1]	**EPDisbld**	**Endpoint Disabled Interrupt**

		Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request. **Values:** 0x0 (INACTIVE): No Endpoint Disabled Interrupt 0x1 (ACTIVE): Endpoint Disabled Interrupt
[0]	**XferCompl**	**Transfer Completed Interrupt** Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is set. **Values:** 0x0 (INACTIVE): No Transfer Complete Interrupt 0x1 (ACTIVE): Transfer Complete Interrupt

DIEPDMA7 Register (DIEPDMA7)

Register	Offset	R/W	Description	Reset Value
DIEPDMA7	HSUSBD_BA+0x9f4	R/W	Device IN Endpoint 7 DMA Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMAAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMAAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMAAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMAAddr							

Bits	Description	
[31:0]	**DMAAddr**	**Holds the start address of the external memory for storing or fetching endpoint data** **Note:** For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address. This field indicates the base pointer for the descriptor list.

DTXFSTS7 Register (DTXFSTS7)

Register	Offset	R/W	Description	Reset Value
DTXFSTS7	HSUSBD_BA+0x9f8	R	Device IN Endpoint Transmit FIFO Status Register 7	0x0000_0020

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
INEPTxFSpcAvail							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INEPTxFSpcAvail							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**INEPTxFSpcAvail**	**IN Endpoint TxFIFO Space Avail** Indicates the amount of free space available in the Endpoint TxFIFO. Values are in terms of 32-bit words. • 16'h0: Endpoint TxFIFO is full • 16'h1: 1 word available • 16'h2: 2 words available • 16'hn: n words available (where 0 n 32,768) • 16'h8000: 32,768 words available • Others: Reserved

DIEPDMAB7 Register (DIEPDMAB7)

Register	Offset	R/W	Description	Reset Value
DIEPDMAB7	HSUSBD_BA+0x9fc	R	Device IN Endpoint 7 Buffer Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMABufferAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMABufferAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMABufferAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMABufferAddr							

Bits	Description	
[31:0]	**DMABufferAddr**	**Holds the current buffer address This register is updated as and when the data transfer for the corresponding end point is in progress**

DIEPCTL8 Register (DIEPCTL8)

Register	Offset	R/W	Description	Reset Value
DIEPCTL8	HSUSBD_BA+0xa00	R/W	Device Control IN Endpoint 8 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
EPEna	EPDis	Reserved		SNAK	CNAK	TxFNum	
23	22	21	20	19	18	17	16
TxFNum		Stall	Reserved	EPType		NAKSts	DPID
15	14	13	12	11	10	9	8
USBActEP	Reserved				MPS		
7	6	5	4	3	2	1	0
MPS							

Bits	Description	
[31]	**EPEna**	**Endpoint Enable** Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled, For IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. For OUT endpoint it indicates that the descriptor structure and data buffer to receive data is setup. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done Endpoint Disabled Transfer Completed **Note:** For control endpoints in DMA mode, this bit must be set to be able to transfer SETUP data packets in memory. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Enable Endpoint
[30]	**EPDis**	**Endpoint Disable** Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Disable Endpoint
[29:28]	**Reserved**	**Reserved.**
[27]	**SNAK**	**Set NAK** A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also Set this bit for an endpoint after a SETUP packet is received on that endpoint. **Values:**

		0x0 (INACTIVE): No Set NAK 0x1 (ACTIVE): Set NAK
[26]	**CNAK**	**Clear NAK** A write to this bit clears the NAK bit for the endpoint. **Values:** 0x0 (INACTIVE): No Clear NAK 0x1 (ACTIVE): Clear NAK
[25:22]	**TxFNum**	**TxFIFO Number** Dedicated FIFO Operation: These bits specify the FIFO number associated with this endpoint. Each active IN endpoint must be programmed to a separate FIFO number. This field is valid only for IN endpoints. **Values:** 0x0 (TXFIFO0): Tx FIFO 0 0x1 (TXFIFO1): Tx FIFO 1 0x2 (TXFIFO2): Tx FIFO 2 0x3 (TXFIFO3): Tx FIFO 3 0x4 (TXFIFO4): Tx FIFO 4 0x5 (TXFIFO5): Tx FIFO 5 0x6 (TXFIFO6): Tx FIFO 6 0x7 (TXFIFO7): Tx FIFO 7 0x8 (TXFIFO8): Tx FIFO 8
[21]	**Stall**	**STALL Handshake** Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core. Applies to control endpoints only. The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (INACTIVE): STALL All non-active tokens 0x1 (ACTIVE): STALL All Active Tokens
[20]	**Reserved**	**Reserved.**
[19:18]	**EPType**	**Endpoint Type** This is the transfer type supported by this logical endpoint 2'b00: Control 2'b01: Isochronous 2'b10: Bulk 2'b11: Interrupt **Values:** 0x0 (CONTROL): Control 0x1 (ISOCHRONOUS): Isochronous 0x2 (BULK): Bulk 0x3 (INTERRUP): Interrupt
[17]	**NAKSts**	**NAK Status (Read Only)** Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status.

		1'b1: The core is transmitting NAK handshakes on this endpoint. When either the application or the core sets this bit: The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet. For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the TxFIFO. For isochronous IN endpoints: The core sends out a zero-length data packet, even if there data is available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (NONNAK): The core is transmitting non-NAK handshakes based on the FIFO status 0x1 (NAK): The core is transmitting NAK handshakes on this endpoint
[16]	**DPID**	**Endpoint Data PID (Read Only)** Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. The applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID. 1'b0: DATA0 1'b1: DATA1 **Values:** 0x0 (DATA0EVENFRM): DATA0 or Even Frame 0x1 (DATA1ODDFRM): DATA1 or Odd Frame
[15]	**USBActEP**	**USB Active Endpoint** Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints (other than EP 0) after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit. **Values:** 0x0 (DISABLED): Not Active 0x1 (ENABLED): USB Active Endpoint
[14:11]	**Reserved**	**Reserved.**
[10:0]	**MPS**	**Maximum Packet Size** The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.

DIEPINT8 Register (DIEPINT8)

Register	Offset	R/W	Description	Reset Value
DIEPINT8	HSUSBD_BA+0xa08	R/W	Device IN Endpoint 8 Interrupt Register	0x0000_0080

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	NYETIntrpt	NAKIntrpt	BbleErr	Reserved		BNAIntr	TxfifoUndrn
7	6	5	4	3	2	1	0
TxFEmp	INEPNakEff	INTknEPMis	INTknTXFEmp	Reserved	AHBErr	EPDisbld	XferCompl

Bits	Description	
[31:15]	**Reserved**	**Reserved.**
[14]	**NYETIntrpt**	**NYET Interrupt** The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint. **Values:** 0x0 (INACTIVE): No NYET interrupt 0x1 (ACTIVE): NYET Interrupt
[13]	**NAKIntrpt**	**NAK Interrupt** The core generates this interrupt when a NAK is transmitted or received by the device. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo. **Values:** 0x0 (INACTIVE): No NAK interrupt 0x1 (ACTIVE): NAK Interrupt
[12]	**BbleErr**	**NAK Interrupt** The core generates this interrupt when babble is received for the endpoint. **Values:** 0x0 (INACTIVE): No interrupt 0x1 (ACTIVE): BbleErr interrupt
[11:10]	**Reserved**	**Reserved.**
[9]	**BNAIntr**	**Buffer Not Available Interrupt** The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as Host busy or DMA done. **Values:** 0x0 (INACTIVE): No BNA interrupt 0x1 (ACTIVE): BNA interrupt
[8]	**TxfifoUndrn**	**Fifo Underrun** Applies to IN endpoints Only

		This bit is valid only If thresholding is enabled. The core generates this interrupt when it detects a transmit FIFO underrun condition for this endpoint. **Values:** 0x0 (INACTIVE): No Tx FIFO Underrun interrupt 0x1 (ACTIVE): TxFIFO Underrun interrupt
[7]	**TxFEmp**	**Transmit FIFO Empty (Read Only)** This bit is valid only for IN endpoints This interrupt is asserted when the TxFIFO for this endpoint is either half or completely empty. The half or completely empty status is determined by the TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.NPTxFEmpLvl)). **Values:** 0x0 (INACTIVE): No Transmit FIFO Empty interrupt 0x1 (ACTIVE): Transmit FIFO Empty interrupt
[6]	**INEPNakEff**	**IN Endpoint NAK Effective** Applies to periodic IN endpoints only. This bit can be cleared when the application clears the IN endpoint NAK by writing to DIEPCTLn.CNAK. This interrupt indicates that the core has sampled the NAK bit Set (either by the application or by the core). The interrupt indicates that the IN endpoint NAK bit Set by the application has taken effect in the core. This interrupt does not guarantee that a NAK handshake is sent on the USB. A STALL bit takes priority over a NAK bit. **Values:** 0x0 (INACTIVE): No Endpoint NAK Effective interrupt 0x1 (ACTIVE): IN Endpoint NAK Effective interrupt
[5]	**INTknEPMis**	**IN Token Received with EP Mismatch** Applies to non-periodic IN endpoints only. Indicates that the data in the top of the non-periodic TxFIFO belongs to an endpoint other than the one for which the IN token was received. This interrupt is asserted on the endpoint for which the IN token was received. **Values:** 0x0 (INACTIVE): No IN Token Received with EP Mismatch interrupt 0x1 (ACTIVE): IN Token Received with EP Mismatch interrupt
[4]	**INTknTXFEmp**	**IN Token Received When TxFIFO is Empty** Applies to non-periodic IN endpoints only. Indicates that an IN token was received when the associated TxFIFO (periodic/non-periodic) was empty. This interrupt is asserted on the endpoint for which the IN token was received. **Values:** 0x0 (INACTIVE): No IN Token Received interrupt 0x1 (ACTIVE): IN Token Received Interrupt
[3]	**Reserved**	**Reserved.**
[2]	**AHBErr**	**AHB Error** Applies to IN and OUT endpoints. When there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address. **Values:** 0x0 (INACTIVE): No AHB Error Interrupt 0x1 (ACTIVE): AHB Error interrupt
[1]	**EPDisbld**	**Endpoint Disabled Interrupt**

		Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request. **Values:** 0x0 (INACTIVE): No Endpoint Disabled Interrupt 0x1 (ACTIVE): Endpoint Disabled Interrupt
[0]	**XferCompl**	**Transfer Completed Interrupt** Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is set. **Values:** 0x0 (INACTIVE): No Transfer Complete Interrupt 0x1 (ACTIVE): Transfer Complete Interrupt

DIEPDMA8 Register (DIEPDMA8)

Register	Offset	R/W	Description	Reset Value
DIEPDMA8	HSUSBD_BA+0xa14	R/W	Device IN Endpoint 8 DMA Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMAAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMAAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMAAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMAAddr							

Bits	Description	
[31:0]	**DMAAddr**	**Holds the start address of the external memory for storing or fetching endpoint data** **Note:** For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address. This field indicates the base pointer for the descriptor list.

DTXFSTS8 Register (DTXFSTS8)

Register	Offset	R/W	Description	Reset Value
DTXFSTS8	HSUSBD_BA+0xa18	R	Device IN Endpoint Transmit FIFO Status Register 8	0x0000_0020

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
INEPTxFSpcAvail							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INEPTxFSpcAvail							

Bits	Description	
[31:16]	**Reserved**	**Reserved.**
[15:0]	**INEPTxFSpcAvail**	**IN Endpoint TxFIFO Space Avail** Indicates the amount of free space available in the Endpoint TxFIFO. Values are in terms of 32-bit words. 16'h0: Endpoint TxFIFO is full 16'h1: 1 word available 16'h2: 2 words available 16'hn: n words available (where 0 n 32,768) 16'h8000: 32,768 words available Others: Reserved

DIEPDMAB8 Register (DIEPDMAB8)

Register	Offset	R/W	Description	Reset Value
DIEPDMAB8	HSUSBD_BA+0xa1c	R	Device IN Endpoint 8 Buffer Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMABufferAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMABufferAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMABufferAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMABufferAddr							

Bits	Description	
[31:0]	**DMABufferAddr**	**Holds the current buffer address This register is updated as and when the data transfer for the corresponding end point is in progress**

DOEPCTL0 Register (DOEPCTL0)

Register	Offset	R/W	Description	Reset Value
DOEPCTL0	HSUSBD_BA+0xb00	R/W	Device Control OUT Endpoint 0 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
EPEna	EPDis	Reserved		SNAK	CNAK	Reserved	
23	22	21	20	19	18	17	16
Reserved		Stall	Snp	EPType		NAKSts	Reserved
15	14	13	12	11	10	9	8
USBActEP	Reserved						
7	6	5	4	3	2	1	0
Reserved						MPS	

Bits	Description	
[31]	**EPEna**	**Endpoint Enable** When Scatter/Gather DMA mode is enabled, for OUT endpoints this bit indicates that the descriptor structure and data buffer to receive data is setup. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done Endpoint Disabled Transfer Completed **Note:** This bit must be set for the core to transfer SETUP data packets into memory. **Values:** 0x0 (INACTIVE): No action 0x1 (ACTIVE): Enable Endpoint
[30]	**EPDis**	**Endpoint Disable (Read Only)** The application cannot disable control OUT endpoint 0. **Values:** 0x0 (INACTIVE): No Endpoint disable
[29:28]	**Reserved**	**Reserved.**
[27]	**SNAK**	**Set NAK** A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set bit on a Transfer Completed interrupt, or after a SETUP is received on the endpoint. **Values:** 0x0 (NOSET): No action 0x1 (SET): Set NAK
[26]	**CNAK**	**Clear NAK** A write to this bit clears the NAK bit for the endpoint. **Values:** 0x0 (NOCLEAR): No action 0x1 (CLEAR): Clear NAK

[25:22]	**Reserved**	**Reserved.**
[21]	**Stall**	**STALL Handshake** The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit or Global OUT NAK is Set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (INACTIVE): No Stall 0x1 (ACTIVE): Stall Handshake
[20]	**Snp**	Reserved.
[19:18]	**EPType**	**Endpoint Type (Read Only)** Hardcoded to 2'b00 for control. **Values:** 0x0 (ACTIVE): Endpoint Control 0
[17]	**NAKSts**	**NAK Status (Read Only)** Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status. 1'b1: The core is transmitting NAK handshakes on this endpoint. When either the application or the core sets this bit, the core stops receiving data, even If there is space in the RxFIFO to accommodate the incoming packet. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (INACTIVE): The core is transmitting non-NAK handshakes based on the FIFO status 0x1 (ACTIVE): The core is transmitting NAK handshakes on this endpoint
[16]	**Reserved**	**Reserved.**
[15]	**USBActEP**	**USB Active Endpoint (Read Only)** This bit is always set to 1, indicating that a control endpoint 0 is always active in all configurations and interfaces. **Values:** 0x1 (ACTIVE): USB Active Endpoint 0
[14:2]	**Reserved**	**Reserved.**
[1:0]	**MPS**	**Maximum Packet Size (Read Only)** The maximum packet size for control OUT endpoint 0 is the same as what is programmed in control IN Endpoint 0. 2'b00: 64 bytes 2'b01: 32 bytes 2'b10: 16 bytes 2'b11: 8 bytes **Values:** 0x0 (BYTE64): 64 bytes 0x1 (BYTE32): 32 bytes 0x2 (BYTE16): 16 bytes 0x3 (BYTE8): 8 bytes **Value After Reset:** 0x

DOEPINT0 Register (DOEPINT0)

Register	Offset	R/W	Description	Reset Value
DOEPINT0	HSUSBD_BA+0xb08	R/W	Device OUT Endpoint 0 Interrupt Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	NYETIntrpt	NAKIntrpt	BbleErr	Reserved		BNAIntr	OutPktErr
7	6	5	4	3	2	1	0
Reserved	Back2BackSETup	StsPhseRcvd	OUTTknEPdis	SetUp	AHBErr	EPDisbld	XferCompl

Bits	Description	
[31:15]	**Reserved**	**Reserved.**
[14]	**NYETIntrpt**	**NYET Interrupt** The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint. **Values:** 0x0 (INACTIVE): No NYET interrupt 0x1 (ACTIVE): NYET Interrupt
[13]	**NAKIntrpt**	**NAK Interrupt** The core generates this interrupt when a NAK is transmitted or received by the device. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo. **Values:** 0x0 (INACTIVE): No NAK interrupt 0x1 (ACTIVE): NAK Interrupt
[12]	**BbleErr**	**NAK Interrupt** The core generates this interrupt when babble is received for the endpoint. **Values:** 0x0 (INACTIVE): No BbleErr interrupt 0x1 (ACTIVE): BbleErr interrupt
[11:10]	**Reserved**	**Reserved.**
[9]	**BNAIntr**	**Buffer Not Available Interrupt** The core generates this interrupt when the descriptor accessed is not ready for the core to process, such as Host busy or DMA done. **Values:** 0x0 (INACTIVE): No BNA interrupt 0x1 (ACTIVE): BNA interrupt
[8]	**OutPktErr**	**OUT Packet Error** Applies to OUT endpoints Only

		This interrupt is valid only when thresholding is enabled. This interrupt is asserted when the core detects an overflow or a CRC error for non-Isochronous OUT packet. **Values:** 0x0 (INACTIVE): No OUT Packet Error 0x1 (ACTIVE): OUT Packet Error
[7]	**Reserved**	**Reserved.**
[6]	**Back2BackSETup**	**Back-to-Back SETUP Packets Received** Applies to Control OUT endpoints only. This bit indicates that the core has received more than three back-to-back SETUP packets for this particular endpoint. For information about handling this interrupt, **Values:** 0x0 (INACTIVE): No Back-to-Back SETUP Packets Received 0x1 (ACTIVE): Back-to-Back SETUP Packets Received
[5]	**StsPhseRcvd**	**Status Phase Received for Control Write** This interrupt is valid only for Control OUT endpoints. This interrupt is generated only after the core has transferred all the data that the host has sent during the data phase of a control write transfer, to the system memory buffer. The interrupt indicates to the application that the host has switched from data phase to the status phase of a Control Write transfer. The application can use this interrupt to ACK or STALL the Status phase, after it has decoded the data phase. **Values:** 0x0 (INACTIVE): No Status Phase Received for Control Write 0x1 (ACTIVE): Status Phase Received for Control Write
[4]	**OUTTknEPdis**	**OUT Token Received When Endpoint Disabled** Applies only to control OUT endpoints. Indicates that an OUT token was received when the endpoint was not yet enabled. This interrupt is asserted on the endpoint for which the OUT token was received. **Values:** 0x0 (INACTIVE): No OUT Token Received When Endpoint Disabled 0x1 (ACTIVE): OUT Token Received When Endpoint Disabled
[3]	**SetUp**	**SETUP Phase Done** Applies to control OUT endpoints only. Indicates that the SETUP phase for the control endpoint is complete and no more back-to-back SETUP packets were received for the current control transfer. On this interrupt, the application can decode the received SETUP data packet. **Values:** 0x0 (INACTIVE): No SETUP Phase Done 0x1 (ACTIVE): SETUP Phase Done
[2]	**AHBErr**	**AHB Error** Applies to IN and OUT endpoints. This is generated when there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address. **Values:** 0x0 (INACTIVE): No AHB Error Interrupt 0x1 (ACTIVE): AHB Error interrupt
[1]	**EPDisbld**	**Endpoint Disabled Interrupt** Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request.

		Values: 0x0 (INACTIVE): No Endpoint Disabled Interrupt 0x1 (ACTIVE): Endpoint Disabled Interrupt
[0]	**XferCompl**	**Transfer Completed Interrupt** Applies to IN and OUT endpoints. For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is Set. **Values:** 0x0 (INACTIVE): No Transfer Complete Interrupt 0x1 (ACTIVE): Transfer Complete Interrupt

DOEPDMA0 Register (DOEPDMA0)

Register	Offset	R/W	Description	Reset Value
DOEPDMA0	HSUSBD_BA+0xb14	R/W	Device OUT Endpoint 0 DMA Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMAAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMAAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMAAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMAAddr							

Bits	Description	
[31:0]	**DMAAddr**	**Holds the start address of the external memory for storing or fetching endpoint data** **Note:** For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address

DOEPDMAB0 Register (DOEPDMAB0)

Register	Offset	R/W	Description	Reset Value
DOEPDMAB0	HSUSBD_BA+0xb1c	R	Device OUT Endpoint 16 Buffer Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMABufferAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMABufferAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMABufferAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMABufferAddr							

Bits	Description	
[31:0]	**DMABufferAddr**	**Holds the current buffer address This register is updated as and when the data transfer for the corresponding end point is in progress**

DOEPCTL1 Register (DOEPCTL1)

Register	Offset	R/W	Description	Reset Value
DOEPCTL1	HSUSBD_BA+0xb20	R/W	Device Control OUT Endpoint 1 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
EPEna	EPDis	SetD1PID	SetD0PID	SNAK	CNAK	Reserved	
23	22	21	20	19	18	17	16
Reserved		Stall	Snp	EPType		NAKSts	DPID
15	14	13	12	11	10	9	8
USBActEP	Reserved				MPS		
7	6	5	4	3	2	1	0
MPS							

Bits	Description	
[31]	EPEna	**Endpoint Enable** Applies to IN and OUT endpoints. For IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. For OUT endpoint it indicates that the descriptor structure and data buffer to receive data is setup. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done Endpoint Disabled Transfer Completed **Note:** For control endpoints in DMA mode, this bit must be set for the controller to transfer SETUP data packets to the memory. This bit will not be cleared on Transfer Completed interrupt of the SETUP packet. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Enable Endpoint
[30]	EPDis	**Endpoint Disable** Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Disable Endpoint
[29]	SetD1PID	**Set DATA1 PID** Applies to interrupt and bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA1. **Values:** 0x0 (DISABLED): Disables Set DATA1 PID or Do not force Odd Frame 0x1 (ENABLED): Set Endpoint Data PID to DATA1 or Sets EO_FrNum field to odd (micro)Frame

[28]	**SetD0PID**	**Set DATA0 PID** Applies to interrupt/bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA0. **Values:** 0x0 (DISABLED): Disables Set DATA0 PID or Do not force Even Frame 0x1 (ENABLED): Set Endpoint Data PID to DATA0 or Sets EO_FrNum field to odd (micro)Frame
[27]	**SNAK**	**Set NAK** A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for an endpoint after a SETUP packet is received on that endpoint. **Values:** 0x0 (INACTIVE): No Set NAK 0x1 (ACTIVE): Set NAK
[26]	**CNAK**	**Clear NAK** A write to this bit clears the NAK bit for the endpoint **Values:** 0x0 (INACTIVE): No Clear NAK 0x1 (ACTIVE): Clear NAK
[25:22]	**Reserved**	**Reserved.**
[21]	**Stall**	**STALL Handshake** Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core. Applies to control endpoints only. The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (INACTIVE): STALL All non-active tokens 0x1 (ACTIVE): STALL All Active Tokens
[20]	**Snp**	Reserved.
[19:18]	**EPType**	**Endpoint Type** This is the transfer type supported by this logical endpoint. 2'b00: Control 2'b01: Isochronous 2'b10: Bulk 2'b11: Interrupt **Values:** 0x0 (CONTROL): Control 0x1 (ISOCHRONOUS): Isochronous 0x2 (BULK): Bulk 0x3 (INTERRUPT): Interrupt
[17]	**NAKSts**	**NAK Status (Read Only)** Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status. 1'b1: The core is transmitting NAK handshakes on this endpoint.

		When either the application or the core sets this bit: The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet. For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the TxFIFO. For isochronous IN endpoints: The core sends out a zero-length data packet, even if there data is available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (NONNAK): The core is transmitting non-NAK handshakes based on the FIFO status 0x1 (NAK): The core is transmitting NAK handshakes on this endpoint
[16]	**DPID**	**Endpoint Data PID (Read Only)** Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. The applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID. 1'b0: DATA0 1'b1: DATA1 **Values:** 0x0 (INACTIVE): Endpoint Data PID not active 0x1 (ACTIVE): Endpoint Data PID active
[15]	**USBActEP**	**USB Active Endpoint** Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints (other than EP 0) after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit. **Values:** 0x0 (DISABLED): Not Active 0x1 (ENABLED): USB Active Endpoint
[14:11]	**Reserved**	**Reserved.**
[10:0]	**MPS**	**Maximum Packet Size** The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.

DOEPINT1 Register (DOEPINT1)

Register	Offset	R/W	Description	Reset Value
DOEPINT1	HSUSBD_BA+0xb28	R/W	Device OUT Endpoint 1 Interrupt Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	NYETIntrpt	NAKIntrpt	BbleErr	PktDrpSts	Reserved	BNAIntr	OutPktErr
7	6	5	4	3	2	1	0
Reserved	Back2BackSETup	StsPhseRcvd	OUTTknEPdis	SetUp	AHBErr	EPDisbld	XferCompl

Bits	Description	
[31:15]	**Reserved**	**Reserved.**
[14]	**NYETIntrpt**	**NYET Interrupt** The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint. **Values:** 0x0 (INACTIVE): No NYET interrupt 0x1 (ACTIVE): NYET Interrupt
[13]	**NAKIntrpt**	**NAK Interrupt** The core generates this interrupt when a NAK is transmitted or received by the device. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo. **Values:** 0x0 (INACTIVE): No NAK interrupt 0x1 (ACTIVE): NAK Interrupt
[12]	**BbleErr**	**NAK Interrupt** The core generates this interrupt when babble is received for the endpoint. **Values:** 0x0 (INACTIVE): No BbleErr interrupt 0x1 (ACTIVE): BbleErr interrupt
[11:10]	**Reserved**	**Reserved.**
[9]	**BNAIntr**	**Buffer Not Available Interrupt** The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as Host busy or DMA done **Values:** 0x0 (INACTIVE): No BNA interrupt 0x1 (ACTIVE): BNA interrupt
[8]	**OutPktErr**	**OUT Packet Error** Applies to OUT endpoints Only

		This interrupt is valid only when thresholding is enabled. This interrupt is asserted when the core detects an overflow or a CRC error for non-Isochronous OUT packet. **Values:** 0x0 (INACTIVE): No OUT Packet Error 0x1 (ACTIVE): OUT Packet Error
[7]	**Reserved**	**Reserved.**
[6]	**Back2BackSETup**	**Back-to-Back SETUP Packets Received** Applies to Control OUT endpoints only. This bit indicates that the core has received more than three back-to-back SETUP packets for this particular endpoint. For information about handling this interrupt, **Values:** 0x0 (INACTIVE): No Back-to-Back SETUP Packets Received 0x1 (ACTIVE): Back-to-Back SETUP Packets Received
[5]	**StsPhseRcvd**	**Status Phase Received for Control Write** This interrupt is valid only for Control OUT endpoints. This interrupt is generated only after the core has transferred all the data that the host has sent during the data phase of a control write transfer, to the system memory buffer. The interrupt indicates to the application that the host has switched from data phase to the status phase of a Control Write transfer. The application can use this interrupt to ACK or STALL the Status phase, after it has decoded the data phase. This is applicable only in Case of Scatter Gather DMA mode. **Values:** 0x0 (INACTIVE): No Status Phase Received for Control Write 0x1 (ACTIVE): Status Phase Received for Control Write
[4]	**OUTTknEPdis**	**OUT Token Received When Endpoint Disabled** Applies only to control OUT endpoints. Indicates that an OUT token was received when the endpoint was not yet enabled. This interrupt is asserted on the endpoint for which the OUT token was received. **Values:** 0x0 (INACTIVE): No OUT Token Received When Endpoint Disabled 0x1 (ACTIVE): OUT Token Received When Endpoint Disabled
[3]	**SetUp**	**SETUP Phase Done** Applies to control OUT endpoints only. Indicates that the SETUP phase for the control endpoint is complete and no more back-to-back SETUP packets were received for the current control transfer. On this interrupt, the application can decode the received SETUP data packet. **Values:** 0x0 (INACTIVE): No SETUP Phase Done 0x1 (ACTIVE): SETUP Phase Done
[2]	**AHBErr**	**AHB Error** Applies to IN and OUT endpoints. When there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address. **Values:** 0x0 (INACTIVE): No AHB Error Interrupt 0x1 (ACTIVE): AHB Error interrupt
[1]	**EPDisbld**	**Endpoint Disabled Interrupt** Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request.

		Values: 0x0 (INACTIVE): No Endpoint Disabled Interrupt 0x1 (ACTIVE): Endpoint Disabled Interrupt
[0]	**XferCompl**	**Transfer Completed Interrupt** Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is Set. **Values:** 0x0 (INACTIVE): No Transfer Complete Interrupt 0x1 (ACTIVE): Transfer Complete Interrupt

DOEPDMA1 Register (DOEPDMA1)

Register	Offset	R/W	Description	Reset Value
DOEPDMA1	HSUSBD_BA+0xb34	R/W	Device OUT Endpoint 1 DMA Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMAAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMAAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMAAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMAAddr							

Bits	Description	
[31:0]	**DMAAddr**	**Holds the start address of the external memory for storing or fetching endpoint data** **Note:** For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address.

DOEPDMAB1 Register (DOEPDMAB1)

Register	Offset	R/W	Description	Reset Value
DOEPDMAB1	HSUSBD_BA+0xb3c	R	Device OUT Endpoint 1 Buffer Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMABufferAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMABufferAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMABufferAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMABufferAddr							

Bits	Description	
[31:0]	**DMABufferAddr**	**Holds the current buffer address This register is updated as and when the data transfer for the corresponding end point is in progress.**

DOEPCTL2 Register (DOEPCTL2)

Register	Offset	R/W	Description	Reset Value
DOEPCTL2	HSUSBD_BA+0xb40	R/W	Device Control OUT Endpoint 2 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
EPEna	EPDis	SetD1PID	SetD0PID	SNAK	CNAK	Reserved	
23	22	21	20	19	18	17	16
Reserved		Stall	Snp	EPType		NAKSts	DPID
15	14	13	12	11	10	9	8
USBActEP	Reserved				MPS		
7	6	5	4	3	2	1	0
MPS							

Bits	Description	
[31]	EPEna	**Endpoint Enable** Applies to IN and OUT endpoints. For IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. For OUT endpoint it indicates that the descriptor structure and data buffer to receive data is setup. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done Endpoint Disabled Transfer Completed **Note:** For control endpoints in DMA mode, this bit must be set for the controller to transfer SETUP data packets to the memory. This bit will not be cleared on Transfer Completed interrupt of the SETUP packet. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Enable Endpoint
[30]	EPDis	**Endpoint Disable** Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Disable Endpoint
[29]	SetD1PID	**Set DATA1 PID** Applies to interrupt and bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA1. **Values:** 0x0 (DISABLED): Disables Set DATA1 PID or Do not force Odd Frame 0x1 (ENABLED): Set Endpoint Data PID to DATA1 or Sets EO_FrNum field to odd (micro)Frame

[28]	**SetD0PID**	**Set DATA0 PID** Applies to interrupt/bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA0. **Values:** 0x0 (DISABLED): Disables Set DATA0 PID or Do not force Even Frame 0x1 (ENABLED): Set Endpoint Data PID to DATA0 or Sets EO_FrNum field to odd (micro)Frame
[27]	**SNAK**	**Set NAK** A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for an endpoint after a SETUP packet is received on that endpoint. **Values:** 0x0 (INACTIVE): No Set NAK 0x1 (ACTIVE): Set NAK
[26]	**CNAK**	**Clear NAK** A write to this bit clears the NAK bit for the endpoint **Values:** 0x0 (INACTIVE): No Clear NAK 0x1 (ACTIVE): Clear NAK
[25:22]	**Reserved**	**Reserved.**
[21]	**Stall**	**STALL Handshake** Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core. Applies to control endpoints only. The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (INACTIVE): STALL All non-active tokens 0x1 (ACTIVE): STALL All Active Tokens
[20]	**Snp**	Reserved.
[19:18]	**EPType**	**Endpoint Type** This is the transfer type supported by this logical endpoint. 2'b00: Control 2'b01: Isochronous 2'b10: Bulk 2'b11: Interrupt **Values:** 0x0 (CONTROL): Control 0x1 (ISOCHRONOUS): Isochronous 0x2 (BULK): Bulk 0x3 (INTERRUPT): Interrupt
[17]	**NAKSts**	**NAK Status (Read Only)** Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status. 1'b1: The core is transmitting NAK handshakes on this endpoint.

		When either the application or the core sets this bit: The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet. For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the TxFIFO. For isochronous IN endpoints: The core sends out a zero-length data packet, even if there data is available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (NONNAK): The core is transmitting non-NAK handshakes based on the FIFO status 0x1 (NAK): The core is transmitting NAK handshakes on this endpoint
[16]	**DPID**	**Endpoint Data PID (Read Only)** Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. The applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID. 1'b0: DATA0 1'b1: DATA1 **Values:** 0x0 (INACTIVE): Endpoint Data PID not active 0x1 (ACTIVE): Endpoint Data PID active
[15]	**USBActEP**	**USB Active Endpoint** Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints (other than EP 0) after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit. **Values:** 0x0 (DISABLED): Not Active 0x1 (ENABLED): USB Active Endpoint
[14:11]	**Reserved**	**Reserved.**
[10:0]	**MPS**	**Maximum Packet Size** The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.

DOEPINT2 Register (DOEPINT2)

Register	Offset	R/W	Description	Reset Value
DOEPINT2	HSUSBD_BA+0xb48	R/W	Device OUT Endpoint 2 Interrupt Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	NYETIntrpt	NAKIntrpt	BbleErr	PktDrpSts	Reserved	BNAIntr	OutPktErr
7	6	5	4	3	2	1	0
Reserved	Back2BackSETup	StsPhseRcvd	OUTTknEPdis	SetUp	AHBErr	EPDisbld	XferCompl

Bits	Description	
[31:15]	**Reserved**	**Reserved.**
[14]	**NYETIntrpt**	**NYET Interrupt** The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint. **Values:** 0x0 (INACTIVE): No NYET interrupt 0x1 (ACTIVE): NYET Interrupt
[13]	**NAKIntrpt**	**NAK Interrupt** The core generates this interrupt when a NAK is transmitted or received by the device. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo. **Values:** 0x0 (INACTIVE): No NAK interrupt 0x1 (ACTIVE): NAK Interrupt
[12]	**BbleErr**	**NAK Interrupt** The core generates this interrupt when babble is received for the endpoint. **Values:** 0x0 (INACTIVE): No BbleErr interrupt 0x1 (ACTIVE): BbleErr interrupt
[11:10]	**Reserved**	**Reserved.**
[9]	**BNAIntr**	**Buffer Not Available Interrupt** The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as Host busy or DMA done **Values:** 0x0 (INACTIVE): No BNA interrupt 0x1 (ACTIVE): BNA interrupt
[8]	**OutPktErr**	**OUT Packet Error** Applies to OUT endpoints Only

		This interrupt is valid only when thresholding is enabled. This interrupt is asserted when the core detects an overflow or a CRC error for non-Isochronous OUT packet. **Values:** 0x0 (INACTIVE): No OUT Packet Error 0x1 (ACTIVE): OUT Packet Error
[7]	**Reserved**	**Reserved.**
[6]	**Back2BackSETup**	**Back-to-Back SETUP Packets Received** Applies to Control OUT endpoints only. This bit indicates that the core has received more than three back-to-back SETUP packets for this particular endpoint. For information about handling this interrupt, **Values:** 0x0 (INACTIVE): No Back-to-Back SETUP Packets Received 0x1 (ACTIVE): Back-to-Back SETUP Packets Received
[5]	**StsPhseRcvd**	**Status Phase Received for Control Write** This interrupt is valid only for Control OUT endpoints. This interrupt is generated only after the core has transferred all the data that the host has sent during the data phase of a control write transfer, to the system memory buffer. The interrupt indicates to the application that the host has switched from data phase to the status phase of a Control Write transfer. The application can use this interrupt to ACK or STALL the Status phase, after it has decoded the data phase. This is applicable only in Case of Scatter Gather DMA mode. **Values:** 0x0 (INACTIVE): No Status Phase Received for Control Write 0x1 (ACTIVE): Status Phase Received for Control Write
[4]	**OUTTknEPdis**	**OUT Token Received When Endpoint Disabled** Applies only to control OUT endpoints. Indicates that an OUT token was received when the endpoint was not yet enabled. This interrupt is asserted on the endpoint for which the OUT token was received. **Values:** 0x0 (INACTIVE): No OUT Token Received When Endpoint Disabled 0x1 (ACTIVE): OUT Token Received When Endpoint Disabled
[3]	**SetUp**	**SETUP Phase Done** Applies to control OUT endpoints only. Indicates that the SETUP phase for the control endpoint is complete and no more back-to-back SETUP packets were received for the current control transfer. On this interrupt, the application can decode the received SETUP data packet. **Values:** 0x0 (INACTIVE): No SETUP Phase Done 0x1 (ACTIVE): SETUP Phase Done
[2]	**AHBErr**	**AHB Error** Applies to IN and OUT endpoints. When there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address. **Values:** 0x0 (INACTIVE): No AHB Error Interrupt 0x1 (ACTIVE): AHB Error interrupt
[1]	**EPDisbld**	**Endpoint Disabled Interrupt** Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request.

		Values: 0x0 (INACTIVE): No Endpoint Disabled Interrupt 0x1 (ACTIVE): Endpoint Disabled Interrupt
[0]	**XferCompl**	**Transfer Completed Interrupt** Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is Set. **Values:** 0x0 (INACTIVE): No Transfer Complete Interrupt 0x1 (ACTIVE): Transfer Complete Interrupt

DOEPDMA2 Register (DOEPDMA2)

Register	Offset	R/W	Description	Reset Value
DOEPDMA2	HSUSBD_BA+0xb54	R/W	Device OUT Endpoint 2 DMA Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMAAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMAAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMAAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMAAddr							

Bits	Description	
[31:0]	**DMAAddr**	**Holds the start address of the external memory for storing or fetching endpoint data** **Note:** For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address.

DOEPDMAB2 Register (DOEPDMAB2)

Register	Offset	R/W	Description	Reset Value
DOEPDMAB2	HSUSBD_BA+0xb5c	R	Device OUT Endpoint 2 Buffer Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMABufferAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMABufferAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMABufferAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMABufferAddr							

Bits	Description	
[31:0]	**DMABufferAddr**	**Holds the current buffer address This register is updated as and when the data transfer for the corresponding end point is in progress.**

DOEPCTL3 Register (DOEPCTL3)

Register	Offset	R/W	Description	Reset Value
DOEPCTL3	HSUSBD_BA+0xb60	R/W	Device Control OUT Endpoint 3 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
EPEna	EPDis	SetD1PID	SetD0PID	SNAK	CNAK	Reserved	
23	22	21	20	19	18	17	16
Reserved		Stall	Snp	EPType		NAKSts	DPID
15	14	13	12	11	10	9	8
USBActEP	Reserved				MPS		
7	6	5	4	3	2	1	0
MPS							

Bits	Description	
[31]	EPEna	**Endpoint Enable** Applies to IN and OUT endpoints. For IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. For OUT endpoint it indicates that the descriptor structure and data buffer to receive data is setup. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done Endpoint Disabled Transfer Completed **Note:** For control endpoints in DMA mode, this bit must be set for the controller to transfer SETUP data packets to the memory. This bit will not be cleared on Transfer Completed interrupt of the SETUP packet. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Enable Endpoint
[30]	EPDis	**Endpoint Disable** Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Disable Endpoint
[29]	SetD1PID	**Set DATA1 PID** Applies to interrupt and bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA1. **Values:** 0x0 (DISABLED): Disables Set DATA1 PID or Do not force Odd Frame 0x1 (ENABLED): Set Endpoint Data PID to DATA1 or Sets EO_FrNum field to odd (micro)Frame

[28]	**SetD0PID**	**Set DATA0 PID** Applies to interrupt/bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA0. **Values:** 0x0 (DISABLED): Disables Set DATA0 PID or Do not force Even Frame 0x1 (ENABLED): Set Endpoint Data PID to DATA0 or Sets EO_FrNum field to odd (micro)Frame
[27]	**SNAK**	**Set NAK** A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for an endpoint after a SETUP packet is received on that endpoint. **Values:** 0x0 (INACTIVE): No Set NAK 0x1 (ACTIVE): Set NAK
[26]	**CNAK**	**Clear NAK** A write to this bit clears the NAK bit for the endpoint **Values:** 0x0 (INACTIVE): No Clear NAK 0x1 (ACTIVE): Clear NAK
[25:22]	**Reserved**	**Reserved.**
[21]	**Stall**	**STALL Handshake** Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core. Applies to control endpoints only. The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (INACTIVE): STALL All non-active tokens 0x1 (ACTIVE): STALL All Active Tokens
[20]	**Snp**	Reserved.
[19:18]	**EPType**	**Endpoint Type** This is the transfer type supported by this logical endpoint. 2'b00: Control 2'b01: Isochronous 2'b10: Bulk 2'b11: Interrupt **Values:** 0x0 (CONTROL): Control 0x1 (ISOCHRONOUS): Isochronous 0x2 (BULK): Bulk 0x3 (INTERRUPT): Interrupt
[17]	**NAKSts**	**NAK Status (Read Only)** Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status. 1'b1: The core is transmitting NAK handshakes on this endpoint.

		When either the application or the core sets this bit: The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet. For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the TxFIFO. For isochronous IN endpoints: The core sends out a zero-length data packet, even if there data is available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (NONNAK): The core is transmitting non-NAK handshakes based on the FIFO status 0x1 (NAK): The core is transmitting NAK handshakes on this endpoint
[16]	**DPID**	**Endpoint Data PID (Read Only)** Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. The applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID. 1'b0: DATA0 1'b1: DATA1 **Values:** 0x0 (INACTIVE): Endpoint Data PID not active 0x1 (ACTIVE): Endpoint Data PID active
[15]	**USBActEP**	**USB Active Endpoint** Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints (other than EP 0) after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit. **Values:** 0x0 (DISABLED): Not Active 0x1 (ENABLED): USB Active Endpoint
[14:11]	**Reserved**	**Reserved.**
[10:0]	**MPS**	**Maximum Packet Size** The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.

DOEPINT3 Register (DOEPINT3)

Register	Offset	R/W	Description	Reset Value
DOEPINT3	HSUSBD_BA+0xb68	R/W	Device OUT Endpoint 3 Interrupt Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	NYETIntrpt	NAKIntrpt	BbleErr	PktDrpSts	Reserved	BNAIntr	OutPktErr
7	6	5	4	3	2	1	0
Reserved	Back2BackSETup	StsPhseRcvd	OUTTknEPdis	SetUp	AHBErr	EPDisbld	XferCompl

Bits	Description	
[31:15]	**Reserved**	**Reserved.**
[14]	**NYETIntrpt**	**NYET Interrupt** The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint. **Values:** 0x0 (INACTIVE): No NYET interrupt 0x1 (ACTIVE): NYET Interrupt
[13]	**NAKIntrpt**	**NAK Interrupt** The core generates this interrupt when a NAK is transmitted or received by the device. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo. **Values:** 0x0 (INACTIVE): No NAK interrupt 0x1 (ACTIVE): NAK Interrupt
[12]	**BbleErr**	**NAK Interrupt** The core generates this interrupt when babble is received for the endpoint. **Values:** 0x0 (INACTIVE): No BbleErr interrupt 0x1 (ACTIVE): BbleErr interrupt
[11:10]	**Reserved**	**Reserved.**
[9]	**BNAIntr**	**Buffer Not Available Interrupt** The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as Host busy or DMA done **Values:** 0x0 (INACTIVE): No BNA interrupt 0x1 (ACTIVE): BNA interrupt
[8]	**OutPktErr**	**OUT Packet Error** Applies to OUT endpoints Only

		This interrupt is valid only when thresholding is enabled. This interrupt is asserted when the core detects an overflow or a CRC error for non-Isochronous OUT packet. **Values:** 0x0 (INACTIVE): No OUT Packet Error 0x1 (ACTIVE): OUT Packet Error
[7]	**Reserved**	**Reserved.**
[6]	**Back2BackSETup**	**Back-to-Back SETUP Packets Received** Applies to Control OUT endpoints only. This bit indicates that the core has received more than three back-to-back SETUP packets for this particular endpoint. For information about handling this interrupt, **Values:** 0x0 (INACTIVE): No Back-to-Back SETUP Packets Received 0x1 (ACTIVE): Back-to-Back SETUP Packets Received
[5]	**StsPhseRcvd**	**Status Phase Received for Control Write** This interrupt is valid only for Control OUT endpoints. This interrupt is generated only after the core has transferred all the data that the host has sent during the data phase of a control write transfer, to the system memory buffer. The interrupt indicates to the application that the host has switched from data phase to the status phase of a Control Write transfer. The application can use this interrupt to ACK or STALL the Status phase, after it has decoded the data phase. This is applicable only in Case of Scatter Gather DMA mode. **Values:** 0x0 (INACTIVE): No Status Phase Received for Control Write 0x1 (ACTIVE): Status Phase Received for Control Write
[4]	**OUTTknEPdis**	**OUT Token Received When Endpoint Disabled** Applies only to control OUT endpoints. Indicates that an OUT token was received when the endpoint was not yet enabled. This interrupt is asserted on the endpoint for which the OUT token was received. **Values:** 0x0 (INACTIVE): No OUT Token Received When Endpoint Disabled 0x1 (ACTIVE): OUT Token Received When Endpoint Disabled
[3]	**SetUp**	**SETUP Phase Done** Applies to control OUT endpoints only. Indicates that the SETUP phase for the control endpoint is complete and no more back-to-back SETUP packets were received for the current control transfer. On this interrupt, the application can decode the received SETUP data packet. **Values:** 0x0 (INACTIVE): No SETUP Phase Done 0x1 (ACTIVE): SETUP Phase Done
[2]	**AHBErr**	**AHB Error** Applies to IN and OUT endpoints. When there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address. **Values:** 0x0 (INACTIVE): No AHB Error Interrupt 0x1 (ACTIVE): AHB Error interrupt
[1]	**EPDisbld**	**Endpoint Disabled Interrupt** Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request.

		Values: 0x0 (INACTIVE): No Endpoint Disabled Interrupt 0x1 (ACTIVE): Endpoint Disabled Interrupt
[0]	**XferCompl**	**Transfer Completed Interrupt** Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is Set. **Values:** 0x0 (INACTIVE): No Transfer Complete Interrupt 0x1 (ACTIVE): Transfer Complete Interrupt

DOEPDMA3 Register (DOEPDMA3)

Register	Offset	R/W	Description	Reset Value
DOEPDMA3	HSUSBD_BA+0xb74	R/W	Device OUT Endpoint 3 DMA Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMAAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMAAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMAAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMAAddr							

Bits	Description	
[31:0]	**DMAAddr**	**Holds the start address of the external memory for storing or fetching endpoint data** **Note:** For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address.

DOEPDMAB3 Register (DOEPDMAB3)

Register	Offset	R/W	Description	Reset Value
DOEPDMAB3	HSUSBD_BA+0xb7c	R	Device OUT Endpoint 3 Buffer Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMABufferAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMABufferAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMABufferAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMABufferAddr							

Bits	Description	
[31:0]	**DMABufferAddr**	**Holds the current buffer address This register is updated as and when the data transfer for the corresponding end point is in progress.**

DOEPCTL4 Register (DOEPCTL4)

Register	Offset	R/W	Description	Reset Value
DOEPCTL4	HSUSBD_BA+0xb80	R/W	Device Control OUT Endpoint 4 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
EPEna	EPDis	SetD1PID	SetD0PID	SNAK	CNAK	Reserved	
23	22	21	20	19	18	17	16
Reserved		Stall	Snp	EPType		NAKSts	DPID
15	14	13	12	11	10	9	8
USBActEP	Reserved				MPS		
7	6	5	4	3	2	1	0
MPS							

Bits	Description	
[31]	EPEna	**Endpoint Enable** Applies to IN and OUT endpoints. For IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. For OUT endpoint it indicates that the descriptor structure and data buffer to receive data is setup. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done Endpoint Disabled Transfer Completed **Note:** For control endpoints in DMA mode, this bit must be set for the controller to transfer SETUP data packets to the memory. This bit will not be cleared on Transfer Completed interrupt of the SETUP packet. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Enable Endpoint
[30]	EPDis	**Endpoint Disable** Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Disable Endpoint
[29]	SetD1PID	**Set DATA1 PID** Applies to interrupt and bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA1. **Values:** 0x0 (DISABLED): Disables Set DATA1 PID or Do not force Odd Frame 0x1 (ENABLED): Set Endpoint Data PID to DATA1 or Sets EO_FrNum field to odd (micro)Frame

[28]	**SetD0PID**	**Set DATA0 PID** Applies to interrupt/bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA0. **Values:** 0x0 (DISABLED): Disables Set DATA0 PID or Do not force Even Frame 0x1 (ENABLED): Set Endpoint Data PID to DATA0 or Sets EO_FrNum field to odd (micro)Frame
[27]	**SNAK**	**Set NAK** A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for an endpoint after a SETUP packet is received on that endpoint. **Values:** 0x0 (INACTIVE): No Set NAK 0x1 (ACTIVE): Set NAK
[26]	**CNAK**	**Clear NAK** A write to this bit clears the NAK bit for the endpoint **Values:** 0x0 (INACTIVE): No Clear NAK 0x1 (ACTIVE): Clear NAK
[25:22]	**Reserved**	**Reserved.**
[21]	**Stall**	**STALL Handshake** Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core. Applies to control endpoints only. The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (INACTIVE): STALL All non-active tokens 0x1 (ACTIVE): STALL All Active Tokens
[20]	**Snp**	Reserved.
[19:18]	**EPType**	**Endpoint Type** This is the transfer type supported by this logical endpoint. 2'b00: Control 2'b01: Isochronous 2'b10: Bulk 2'b11: Interrupt **Values:** 0x0 (CONTROL): Control 0x1 (ISOCHRONOUS): Isochronous 0x2 (BULK): Bulk 0x3 (INTERRUPT): Interrupt
[17]	**NAKSts**	**NAK Status (Read Only)** Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status. 1'b1: The core is transmitting NAK handshakes on this endpoint.

		When either the application or the core sets this bit: The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet. For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the TxFIFO. For isochronous IN endpoints: The core sends out a zero-length data packet, even if there data is available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (NONNAK): The core is transmitting non-NAK handshakes based on the FIFO status 0x1 (NAK): The core is transmitting NAK handshakes on this endpoint
[16]	**DPID**	**Endpoint Data PID (Read Only)** Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. The applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID. 1'b0: DATA0 1'b1: DATA1 **Values:** 0x0 (INACTIVE): Endpoint Data PID not active 0x1 (ACTIVE): Endpoint Data PID active
[15]	**USBActEP**	**USB Active Endpoint** Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints (other than EP 0) after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit. **Values:** 0x0 (DISABLED): Not Active 0x1 (ENABLED): USB Active Endpoint
[14:11]	**Reserved**	**Reserved.**
[10:0]	**MPS**	**Maximum Packet Size** The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.

DOEPINT4 Register (DOEPINT4)

Register	Offset	R/W	Description	Reset Value
DOEPINT4	HSUSBD_BA+0xb88	R/W	Device OUT Endpoint 4 Interrupt Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	NYETIntrpt	NAKIntrpt	BbleErr	PktDrpSts	Reserved	BNAIntr	OutPktErr
7	6	5	4	3	2	1	0
Reserved	Back2BackSETup	StsPhseRcvd	OUTTknEPdis	SetUp	AHBErr	EPDisbld	XferCompl

Bits	Description	
[31:15]	**Reserved**	**Reserved.**
[14]	**NYETIntrpt**	**NYET Interrupt** The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint. **Values:** 0x0 (INACTIVE): No NYET interrupt 0x1 (ACTIVE): NYET Interrupt
[13]	**NAKIntrpt**	**NAK Interrupt** The core generates this interrupt when a NAK is transmitted or received by the device. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo. **Values:** 0x0 (INACTIVE): No NAK interrupt 0x1 (ACTIVE): NAK Interrupt
[12]	**BbleErr**	**NAK Interrupt** The core generates this interrupt when babble is received for the endpoint. **Values:** 0x0 (INACTIVE): No BbleErr interrupt 0x1 (ACTIVE): BbleErr interrupt
[11:10]	**Reserved**	**Reserved.**
[9]	**BNAIntr**	**Buffer Not Available Interrupt** The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as Host busy or DMA done **Values:** 0x0 (INACTIVE): No BNA interrupt 0x1 (ACTIVE): BNA interrupt
[8]	**OutPktErr**	**OUT Packet Error** Applies to OUT endpoints Only

		This interrupt is valid only when thresholding is enabled. This interrupt is asserted when the core detects an overflow or a CRC error for non-Isochronous OUT packet. **Values:** 0x0 (INACTIVE): No OUT Packet Error 0x1 (ACTIVE): OUT Packet Error
[7]	**Reserved**	**Reserved.**
[6]	**Back2BackSETup**	**Back-to-Back SETUP Packets Received** Applies to Control OUT endpoints only. This bit indicates that the core has received more than three back-to-back SETUP packets for this particular endpoint. For information about handling this interrupt, **Values:** 0x0 (INACTIVE): No Back-to-Back SETUP Packets Received 0x1 (ACTIVE): Back-to-Back SETUP Packets Received
[5]	**StsPhseRcvd**	**Status Phase Received for Control Write** This interrupt is valid only for Control OUT endpoints. This interrupt is generated only after the core has transferred all the data that the host has sent during the data phase of a control write transfer, to the system memory buffer. The interrupt indicates to the application that the host has switched from data phase to the status phase of a Control Write transfer. The application can use this interrupt to ACK or STALL the Status phase, after it has decoded the data phase. This is applicable only in Case of Scatter Gather DMA mode. **Values:** 0x0 (INACTIVE): No Status Phase Received for Control Write 0x1 (ACTIVE): Status Phase Received for Control Write
[4]	**OUTTknEPdis**	**OUT Token Received When Endpoint Disabled** Applies only to control OUT endpoints. Indicates that an OUT token was received when the endpoint was not yet enabled. This interrupt is asserted on the endpoint for which the OUT token was received. **Values:** 0x0 (INACTIVE): No OUT Token Received When Endpoint Disabled 0x1 (ACTIVE): OUT Token Received When Endpoint Disabled
[3]	**SetUp**	**SETUP Phase Done** Applies to control OUT endpoints only. Indicates that the SETUP phase for the control endpoint is complete and no more back-to-back SETUP packets were received for the current control transfer. On this interrupt, the application can decode the received SETUP data packet. **Values:** 0x0 (INACTIVE): No SETUP Phase Done 0x1 (ACTIVE): SETUP Phase Done
[2]	**AHBErr**	**AHB Error** Applies to IN and OUT endpoints. When there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address. **Values:** 0x0 (INACTIVE): No AHB Error Interrupt 0x1 (ACTIVE): AHB Error interrupt
[1]	**EPDisbld**	**Endpoint Disabled Interrupt** Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request.

		Values: 0x0 (INACTIVE): No Endpoint Disabled Interrupt 0x1 (ACTIVE): Endpoint Disabled Interrupt
[0]	**XferCompl**	**Transfer Completed Interrupt** Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is Set. **Values:** 0x0 (INACTIVE): No Transfer Complete Interrupt 0x1 (ACTIVE): Transfer Complete Interrupt

DOEPDMA4 Register (DOEPDMA4)

Register	Offset	R/W	Description	Reset Value
DOEPDMA4	HSUSBD_BA+0xb94	R/W	Device OUT Endpoint 4 DMA Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMAAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMAAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMAAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMAAddr							

Bits	Description	
[31:0]	**DMAAddr**	**Holds the start address of the external memory for storing or fetching endpoint data** **Note:** For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address.

DOEPDMAB4 Register (DOEPDMAB4)

Register	Offset	R/W	Description	Reset Value
DOEPDMAB4	HSUSBD_BA+0xb9c	R	Device OUT Endpoint 4 Buffer Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMABufferAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMABufferAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMABufferAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMABufferAddr							

Bits	Description	
[31:0]	**DMABufferAddr**	**Holds the current buffer address This register is updated as and when the data transfer for the corresponding end point is in progress.**

DOEPCTL5 Register (DOEPCTL5)

Register	Offset	R/W	Description	Reset Value
DOEPCTL5	HSUSBD_BA+0xba0	R/W	Device Control OUT Endpoint 5 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
EPEna	EPDis	SetD1PID	SetD0PID	SNAK	CNAK	Reserved	
23	22	21	20	19	18	17	16
Reserved		Stall	Snp	EPType		NAKSts	DPID
15	14	13	12	11	10	9	8
USBActEP	Reserved				MPS		
7	6	5	4	3	2	1	0
MPS							

Bits	Description	
[31]	EPEna	**Endpoint Enable** Applies to IN and OUT endpoints. For IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. For OUT endpoint it indicates that the descriptor structure and data buffer to receive data is setup. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done Endpoint Disabled Transfer Completed **Note:** For control endpoints in DMA mode, this bit must be set for the controller to transfer SETUP data packets to the memory. This bit will not be cleared on Transfer Completed interrupt of the SETUP packet. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Enable Endpoint
[30]	EPDis	**Endpoint Disable** Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Disable Endpoint
[29]	SetD1PID	**Set DATA1 PID** Applies to interrupt and bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA1. **Values:** 0x0 (DISABLED): Disables Set DATA1 PID or Do not force Odd Frame 0x1 (ENABLED): Set Endpoint Data PID to DATA1 or Sets EO_FrNum field to odd (micro)Frame

[28]	**SetD0PID**	**Set DATA0 PID** Applies to interrupt/bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA0. **Values:** 0x0 (DISABLED): Disables Set DATA0 PID or Do not force Even Frame 0x1 (ENABLED): Set Endpoint Data PID to DATA0 or Sets EO_FrNum field to odd (micro)Frame
[27]	**SNAK**	**Set NAK** A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for an endpoint after a SETUP packet is received on that endpoint. **Values:** 0x0 (INACTIVE): No Set NAK 0x1 (ACTIVE): Set NAK
[26]	**CNAK**	**Clear NAK** A write to this bit clears the NAK bit for the endpoint **Values:** 0x0 (INACTIVE): No Clear NAK 0x1 (ACTIVE): Clear NAK
[25:22]	**Reserved**	**Reserved.**
[21]	**Stall**	**STALL Handshake** Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core. Applies to control endpoints only. The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (INACTIVE): STALL All non-active tokens 0x1 (ACTIVE): STALL All Active Tokens
[20]	**Snp**	Reserved.
[19:18]	**EPType**	**Endpoint Type** This is the transfer type supported by this logical endpoint. 2'b00: Control 2'b01: Isochronous 2'b10: Bulk 2'b11: Interrupt **Values:** 0x0 (CONTROL): Control 0x1 (ISOCHRONOUS): Isochronous 0x2 (BULK): Bulk 0x3 (INTERRUPT): Interrupt
[17]	**NAKSts**	**NAK Status (Read Only)** Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status. 1'b1: The core is transmitting NAK handshakes on this endpoint.

		When either the application or the core sets this bit: The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet. For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the TxFIFO. For isochronous IN endpoints: The core sends out a zero-length data packet, even if there data is available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (NONNAK): The core is transmitting non-NAK handshakes based on the FIFO status 0x1 (NAK): The core is transmitting NAK handshakes on this endpoint
[16]	**DPID**	**Endpoint Data PID (Read Only)** Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. The applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID. 1'b0: DATA0 1'b1: DATA1 **Values:** 0x0 (INACTIVE): Endpoint Data PID not active 0x1 (ACTIVE): Endpoint Data PID active
[15]	**USBActEP**	**USB Active Endpoint** Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints (other than EP 0) after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit. **Values:** 0x0 (DISABLED): Not Active 0x1 (ENABLED): USB Active Endpoint
[14:11]	**Reserved**	**Reserved.**
[10:0]	**MPS**	**Maximum Packet Size** The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.

DOEPINT5 Register (DOEPINT5)

Register	Offset	R/W	Description	Reset Value
DOEPINT5	HSUSBD_BA+0xba8	R/W	Device OUT Endpoint 5 Interrupt Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	NYETIntrpt	NAKIntrpt	BbleErr	PktDrpSts	Reserved	BNAIntr	OutPktErr
7	6	5	4	3	2	1	0
Reserved	Back2BackSETup	StsPhseRcvd	OUTTknEPdis	SetUp	AHBErr	EPDisbld	XferCompl

Bits	Description	
[31:15]	**Reserved**	**Reserved.**
[14]	**NYETIntrpt**	**NYET Interrupt** The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint. **Values:** 0x0 (INACTIVE): No NYET interrupt 0x1 (ACTIVE): NYET Interrupt
[13]	**NAKIntrpt**	**NAK Interrupt** The core generates this interrupt when a NAK is transmitted or received by the device. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo. **Values:** 0x0 (INACTIVE): No NAK interrupt 0x1 (ACTIVE): NAK Interrupt
[12]	**BbleErr**	**NAK Interrupt** The core generates this interrupt when babble is received for the endpoint. **Values:** 0x0 (INACTIVE): No BbleErr interrupt 0x1(ACTIVE): BbleErr interrupt
[11:10]	**Reserved**	**Reserved.**
[9]	**BNAIntr**	**Buffer Not Available Interrupt** The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as Host busy or DMA done **Values:** 0x0 (INACTIVE): No BNA interrupt 0x1 (ACTIVE): BNA interrupt
[8]	**OutPktErr**	**OUT Packet Error** Applies to OUT endpoints Only

		This interrupt is valid only when thresholding is enabled. This interrupt is asserted when the core detects an overflow or a CRC error for non-Isochronous OUT packet. **Values:** 0x0 (INACTIVE): No OUT Packet Error 0x1 (ACTIVE): OUT Packet Error
[7]	**Reserved**	**Reserved.**
[6]	**Back2BackSETup**	**Back-to-Back SETUP Packets Received** Applies to Control OUT endpoints only. This bit indicates that the core has received more than three back-to-back SETUP packets for this particular endpoint. For information about handling this interrupt, **Values:** 0x0 (INACTIVE): No Back-to-Back SETUP Packets Received 0x1 (ACTIVE): Back-to-Back SETUP Packets Received
[5]	**StsPhseRcvd**	**Status Phase Received for Control Write** This interrupt is valid only for Control OUT endpoints. This interrupt is generated only after the core has transferred all the data that the host has sent during the data phase of a control write transfer, to the system memory buffer. The interrupt indicates to the application that the host has switched from data phase to the status phase of a Control Write transfer. The application can use this interrupt to ACK or STALL the Status phase, after it has decoded the data phase. This is applicable only in Case of Scatter Gather DMA mode. **Values:** 0x0 (INACTIVE): No Status Phase Received for Control Write 0x1 (ACTIVE): Status Phase Received for Control Write
[4]	**OUTTknEPdis**	**OUT Token Received When Endpoint Disabled** Applies only to control OUT endpoints. Indicates that an OUT token was received when the endpoint was not yet enabled. This interrupt is asserted on the endpoint for which the OUT token was received. **Values:** 0x0 (INACTIVE): No OUT Token Received When Endpoint Disabled 0x1 (ACTIVE): OUT Token Received When Endpoint Disabled
[3]	**SetUp**	**SETUP Phase Done** Applies to control OUT endpoints only. Indicates that the SETUP phase for the control endpoint is complete and no more back-to-back SETUP packets were received for the current control transfer. On this interrupt, the application can decode the received SETUP data packet. **Values:** 0x0 (INACTIVE): No SETUP Phase Done 0x1 (ACTIVE): SETUP Phase Done
[2]	**AHBErr**	**AHB Error** Applies to IN and OUT endpoints. When there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address. **Values:** 0x0 (INACTIVE): No AHB Error Interrupt 0x1 (ACTIVE): AHB Error interrupt
[1]	**EPDisbld**	**Endpoint Disabled Interrupt** Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request.

		Values: 0x0 (INACTIVE): No Endpoint Disabled Interrupt 0x1 (ACTIVE): Endpoint Disabled Interrupt
[0]	**XferCompl**	**Transfer Completed Interrupt** Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is Set. **Values:** 0x0 (INACTIVE): No Transfer Complete Interrupt 0x1 (ACTIVE): Transfer Complete Interrupt

DOEPDMA5 Register (DOEPDMA5)

Register	Offset	R/W	Description	Reset Value
DOEPDMA5	HSUSBD_BA+0xbb4	R/W	Device OUT Endpoint 5 DMA Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMAAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMAAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMAAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMAAddr							

Bits	Description	
[31:0]	**DMAAddr**	**Holds the start address of the external memory for storing or fetching endpoint data** **Note:** For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address.

DOEPDMAB5 Register (DOEPDMAB5)

Register	Offset	R/W	Description	Reset Value
DOEPDMAB5	HSUSBD_BA+0xbbc	R	Device OUT Endpoint 5 Buffer Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMABufferAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMABufferAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMABufferAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMABufferAddr							

Bits	Description	
[31:0]	**DMABufferAddr**	**Holds the current buffer address This register is updated as and when the data transfer for the corresponding end point is in progress.**

DOEPCTL6 Register (DOEPCTL6)

Register	Offset	R/W	Description	Reset Value
DOEPCTL6	HSUSBD_BA+0xbc0	R/W	Device Control OUT Endpoint 6 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
EPEna	EPDis	SetD1PID	SetD0PID	SNAK	CNAK	Reserved	
23	22	21	20	19	18	17	16
Reserved		Stall	Snp	EPType		NAKSts	DPID
15	14	13	12	11	10	9	8
USBActEP	Reserved				MPS		
7	6	5	4	3	2	1	0
MPS							

Bits	Description	
[31]	EPEna	**Endpoint Enable** Applies to IN and OUT endpoints. For IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. For OUT endpoint it indicates that the descriptor structure and data buffer to receive data is setup. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done Endpoint Disabled Transfer Completed **Note:** For control endpoints in DMA mode, this bit must be set for the controller to transfer SETUP data packets to the memory. This bit will not be cleared on Transfer Completed interrupt of the SETUP packet. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Enable Endpoint
[30]	EPDis	**Endpoint Disable** Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Disable Endpoint
[29]	SetD1PID	**Set DATA1 PID** Applies to interrupt and bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA1. **Values:** 0x0 (DISABLED): Disables Set DATA1 PID or Do not force Odd Frame 0x1 (ENABLED): Set Endpoint Data PID to DATA1 or Sets EO_FrNum field to odd (micro)Frame

[28]	**SetD0PID**	**Set DATA0 PID** Applies to interrupt/bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA0. **Values:** 0x0 (DISABLED): Disables Set DATA0 PID or Do not force Even Frame 0x1 (ENABLED): Set Endpoint Data PID to DATA0 or Sets EO_FrNum field to odd (micro)Frame
[27]	**SNAK**	**Set NAK** A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for an endpoint after a SETUP packet is received on that endpoint. **Values:** 0x0 (INACTIVE): No Set NAK 0x1 (ACTIVE): Set NAK
[26]	**CNAK**	**Clear NAK** A write to this bit clears the NAK bit for the endpoint **Values:** 0x0 (INACTIVE): No Clear NAK 0x1 (ACTIVE): Clear NAK
[25:22]	**Reserved**	**Reserved.**
[21]	**Stall**	**STALL Handshake** Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core. Applies to control endpoints only. The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (INACTIVE): STALL All non-active tokens 0x1 (ACTIVE): STALL All Active Tokens
[20]	**Snp**	Reserved.
[19:18]	**EPType**	**Endpoint Type** This is the transfer type supported by this logical endpoint. 2'b00: Control 2'b01: Isochronous 2'b10: Bulk 2'b11: Interrupt **Values:** 0x0 (CONTROL): Control 0x1 (ISOCHRONOUS): Isochronous 0x2 (BULK): Bulk 0x3 (INTERRUPT): Interrupt
[17]	**NAKSts**	**NAK Status (Read Only)** Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status. 1'b1: The core is transmitting NAK handshakes on this endpoint.

		When either the application or the core sets this bit: The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet. For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the TxFIFO. For isochronous IN endpoints: The core sends out a zero-length data packet, even if there data is available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (NONNAK): The core is transmitting non-NAK handshakes based on the FIFO status 0x1 (NAK): The core is transmitting NAK handshakes on this endpoint
[16]	**DPID**	**Endpoint Data PID (Read Only)** Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. The applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID. 1'b0: DATA0 1'b1: DATA1 **Values:** 0x0 (INACTIVE): Endpoint Data PID not active 0x1 (ACTIVE): Endpoint Data PID active
[15]	**USBActEP**	**USB Active Endpoint** Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints (other than EP 0) after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit. **Values:** 0x0 (DISABLED): Not Active 0x1 (ENABLED): USB Active Endpoint
[14:11]	**Reserved**	**Reserved.**
[10:0]	**MPS**	**Maximum Packet Size** The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.

DOEPINT6 Register (DOEPINT6)

Register	Offset	R/W	Description	Reset Value
DOEPINT6	HSUSBD_BA+0xbc8	R/W	Device OUT Endpoint 6 Interrupt Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	NYETIntrpt	NAKIntrpt	BbleErr	PktDrpSts	Reserved	BNAIntr	OutPktErr
7	6	5	4	3	2	1	0
Reserved	Back2BackSETup	StsPhseRcvd	OUTTknEPdis	SetUp	AHBErr	EPDisbld	XferCompl

Bits	Description	
[31:15]	**Reserved**	**Reserved.**
[14]	**NYETIntrpt**	**NYET Interrupt** The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint. **Values:** 0x0 (INACTIVE): No NYET interrupt 0x1 (ACTIVE): NYET Interrupt
[13]	**NAKIntrpt**	**NAK Interrupt** The core generates this interrupt when a NAK is transmitted or received by the device. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo. **Values:** 0x0 (INACTIVE): No NAK interrupt 0x1 (ACTIVE): NAK Interrupt
[12]	**BbleErr**	**NAK Interrupt** The core generates this interrupt when babble is received for the endpoint. **Values:** 0x0 (INACTIVE): No BbleErr interrupt 0x1 (ACTIVE): BbleErr interrupt
[11:10]	**Reserved**	**Reserved.**
[9]	**BNAIntr**	**Buffer Not Available Interrupt** The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as Host busy or DMA done **Values:** 0x0 (INACTIVE): No BNA interrupt 0x1 (ACTIVE): BNA interrupt
[8]	**OutPktErr**	**OUT Packet Error** Applies to OUT endpoints Only

		This interrupt is valid only when thresholding is enabled. This interrupt is asserted when the core detects an overflow or a CRC error for non-Isochronous OUT packet. **Values:** 0x0 (INACTIVE): No OUT Packet Error 0x1 (ACTIVE): OUT Packet Error
[7]	**Reserved**	**Reserved.**
[6]	**Back2BackSETup**	**Back-to-Back SETUP Packets Received** Applies to Control OUT endpoints only. This bit indicates that the core has received more than three back-to-back SETUP packets for this particular endpoint. For information about handling this interrupt, **Values:** 0x0 (INACTIVE): No Back-to-Back SETUP Packets Received 0x1 (ACTIVE): Back-to-Back SETUP Packets Received
[5]	**StsPhseRcvd**	**Status Phase Received for Control Write** This interrupt is valid only for Control OUT endpoints. This interrupt is generated only after the core has transferred all the data that the host has sent during the data phase of a control write transfer, to the system memory buffer. The interrupt indicates to the application that the host has switched from data phase to the status phase of a Control Write transfer. The application can use this interrupt to ACK or STALL the Status phase, after it has decoded the data phase. This is applicable only in Case of Scatter Gather DMA mode. **Values:** 0x0 (INACTIVE): No Status Phase Received for Control Write 0x1 (ACTIVE): Status Phase Received for Control Write
[4]	**OUTTknEPdis**	**OUT Token Received When Endpoint Disabled** Applies only to control OUT endpoints. Indicates that an OUT token was received when the endpoint was not yet enabled. This interrupt is asserted on the endpoint for which the OUT token was received. **Values:** 0x0 (INACTIVE): No OUT Token Received When Endpoint Disabled 0x1 (ACTIVE): OUT Token Received When Endpoint Disabled
[3]	**SetUp**	**SETUP Phase Done** Applies to control OUT endpoints only. Indicates that the SETUP phase for the control endpoint is complete and no more back-to-back SETUP packets were received for the current control transfer. On this interrupt, the application can decode the received SETUP data packet. **Values:** 0x0 (INACTIVE): No SETUP Phase Done 0x1 (ACTIVE): SETUP Phase Done
[2]	**AHBErr**	**AHB Error** Applies to IN and OUT endpoints. When there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address. **Values:** 0x0 (INACTIVE): No AHB Error Interrupt 0x1 (ACTIVE): AHB Error interrupt
[1]	**EPDisbld**	**Endpoint Disabled Interrupt** Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request.

		Values: 0x0 (INACTIVE): No Endpoint Disabled Interrupt 0x1 (ACTIVE): Endpoint Disabled Interrupt
[0]	**XferCompl**	**Transfer Completed Interrupt** Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is Set. **Values:** 0x0 (INACTIVE): No Transfer Complete Interrupt 0x1 (ACTIVE): Transfer Complete Interrupt

DOEPDMA6 Register (DOEPDMA6)

Register	Offset	R/W	Description	Reset Value
DOEPDMA6	HSUSBD_BA+0xbd4	R/W	Device OUT Endpoint 6 DMA Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMAAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMAAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMAAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMAAddr							

Bits	Description	
[31:0]	**DMAAddr**	**Holds the start address of the external memory for storing or fetching endpoint data** **Note:** For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address.

DOEPDMAB6 Register (DOEPDMAB6)

Register	Offset	R/W	Description	Reset Value
DOEPDMAB6	HSUSBD_BA+0xbdc	R	Device OUT Endpoint 6 Buffer Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMABufferAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMABufferAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMABufferAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMABufferAddr							

Bits	Description	
[31:0]	**DMABufferAddr**	**Holds the current buffer address This register is updated as and when the data transfer for the corresponding end point is in progress.**

DOEPCTL7 Register (DOEPCTL7)

Register	Offset	R/W	Description	Reset Value
DOEPCTL7	HSUSBD_BA+0xbe0	R/W	Device Control OUT Endpoint 7 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
EPEna	EPDis	SetD1PID	SetD0PID	SNAK	CNAK	Reserved	
23	22	21	20	19	18	17	16
Reserved		Stall	Snp	EPType		NAKSts	DPID
15	14	13	12	11	10	9	8
USBActEP	Reserved				MPS		
7	6	5	4	3	2	1	0
MPS							

Bits	Description	
[31]	EPEna	**Endpoint Enable** Applies to IN and OUT endpoints. For IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. For OUT endpoint it indicates that the descriptor structure and data buffer to receive data is setup. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done Endpoint Disabled Transfer Completed **Note:** For control endpoints in DMA mode, this bit must be set for the controller to transfer SETUP data packets to the memory. This bit will not be cleared on Transfer Completed interrupt of the SETUP packet. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Enable Endpoint
[30]	EPDis	**Endpoint Disable** Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Disable Endpoint
[29]	SetD1PID	**Set DATA1 PID** Applies to interrupt and bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA1. **Values:** 0x0 (DISABLED): Disables Set DATA1 PID or Do not force Odd Frame 0x1 (ENABLED): Set Endpoint Data PID to DATA1 or Sets EO_FrNum field to odd (micro)Frame

[28]	**SetD0PID**	**Set DATA0 PID** Applies to interrupt/bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA0. **Values:** 0x0 (DISABLED): Disables Set DATA0 PID or Do not force Even Frame 0x1 (ENABLED): Set Endpoint Data PID to DATA0 or Sets EO_FrNum field to odd (micro)Frame
[27]	**SNAK**	**Set NAK** A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for an endpoint after a SETUP packet is received on that endpoint. **Values:** 0x0 (INACTIVE): No Set NAK 0x1 (ACTIVE): Set NAK
[26]	**CNAK**	**Clear NAK** A write to this bit clears the NAK bit for the endpoint **Values:** 0x0 (INACTIVE): No Clear NAK 0x1 (ACTIVE): Clear NAK
[25:22]	**Reserved**	**Reserved.**
[21]	**Stall**	**STALL Handshake** Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core. Applies to control endpoints only. The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (INACTIVE): STALL All non-active tokens 0x1 (ACTIVE): STALL All Active Tokens
[20]	**Snp**	Reserved.
[19:18]	**EPType**	**Endpoint Type** This is the transfer type supported by this logical endpoint. 2'b00: Control 2'b01: Isochronous 2'b10: Bulk 2'b11: Interrupt **Values:** 0x0 (CONTROL): Control 0x1 (ISOCHRONOUS): Isochronous 0x2 (BULK): Bulk 0x3 (INTERRUPT): Interrupt
[17]	**NAKSts**	**NAK Status (Read Only)** Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status. 1'b1: The core is transmitting NAK handshakes on this endpoint.

		When either the application or the core sets this bit: The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet. For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the TxFIFO. For isochronous IN endpoints: The core sends out a zero-length data packet, even if there data is available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (NONNAK): The core is transmitting non-NAK handshakes based on the FIFO status 0x1 (NAK): The core is transmitting NAK handshakes on this endpoint
[16]	**DPID**	**Endpoint Data PID (Read Only)** Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. The applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID. 1'b0: DATA0 1'b1: DATA1 **Values:** 0x0 (INACTIVE): Endpoint Data PID not active 0x1 (ACTIVE): Endpoint Data PID active
[15]	**USBActEP**	**USB Active Endpoint** Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints (other than EP 0) after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit. **Values:** 0x0 (DISABLED): Not Active 0x1 (ENABLED): USB Active Endpoint
[14:11]	**Reserved**	**Reserved.**
[10:0]	**MPS**	**Maximum Packet Size** The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.

DOEPINT7 Register (DOEPINT7)

Register	Offset	R/W	Description	Reset Value
DOEPINT7	HSUSBD_BA+0xbe8	R/W	Device OUT Endpoint 7 Interrupt Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	NYETIntrpt	NAKIntrpt	BbleErr	PktDrpSts	Reserved	BNAIntr	OutPktErr
7	6	5	4	3	2	1	0
Reserved	Back2BackSETup	StsPhseRcvd	OUTTknEPdis	SetUp	AHBErr	EPDisbld	XferCompl

Bits	Description	
[31:15]	**Reserved**	**Reserved.**
[14]	**NYETIntrpt**	**NYET Interrupt** The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint. **Values:** 0x0 (INACTIVE): No NYET interrupt 0x1 (ACTIVE): NYET Interrupt
[13]	**NAKIntrpt**	**NAK Interrupt** The core generates this interrupt when a NAK is transmitted or received by the device. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo. **Values:** 0x0 (INACTIVE): No NAK interrupt 0x1 (ACTIVE): NAK Interrupt
[12]	**BbleErr**	**NAK Interrupt** The core generates this interrupt when babble is received for the endpoint. **Values:** 0x0 (INACTIVE): No BbleErr interrupt 0x1 (ACTIVE): BbleErr interrupt
[11:10]	**Reserved**	**Reserved.**
[9]	**BNAIntr**	**Buffer Not Available Interrupt** The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as Host busy or DMA done **Values:** 0x0 (INACTIVE): No BNA interrupt 0x1 (ACTIVE): BNA interrupt
[8]	**OutPktErr**	**OUT Packet Error** Applies to OUT endpoints Only

		This interrupt is valid only when thresholding is enabled. This interrupt is asserted when the core detects an overflow or a CRC error for non-Isochronous OUT packet. **Values:** 0x0 (INACTIVE): No OUT Packet Error 0x1 (ACTIVE): OUT Packet Error
[7]	**Reserved**	**Reserved.**
[6]	**Back2BackSETup**	**Back-to-Back SETUP Packets Received** Applies to Control OUT endpoints only. This bit indicates that the core has received more than three back-to-back SETUP packets for this particular endpoint. For information about handling this interrupt, **Values:** 0x0 (INACTIVE): No Back-to-Back SETUP Packets Received 0x1 (ACTIVE): Back-to-Back SETUP Packets Received
[5]	**StsPhseRcvd**	**Status Phase Received for Control Write** This interrupt is valid only for Control OUT endpoints. This interrupt is generated only after the core has transferred all the data that the host has sent during the data phase of a control write transfer, to the system memory buffer. The interrupt indicates to the application that the host has switched from data phase to the status phase of a Control Write transfer. The application can use this interrupt to ACK or STALL the Status phase, after it has decoded the data phase. This is applicable only in Case of Scatter Gather DMA mode. **Values:** 0x0 (INACTIVE): No Status Phase Received for Control Write 0x1 (ACTIVE): Status Phase Received for Control Write
[4]	**OUTTknEPdis**	**OUT Token Received When Endpoint Disabled** Applies only to control OUT endpoints. Indicates that an OUT token was received when the endpoint was not yet enabled. This interrupt is asserted on the endpoint for which the OUT token was received. **Values:** 0x0 (INACTIVE): No OUT Token Received When Endpoint Disabled 0x1 (ACTIVE): OUT Token Received When Endpoint Disabled
[3]	**SetUp**	**SETUP Phase Done** Applies to control OUT endpoints only. Indicates that the SETUP phase for the control endpoint is complete and no more back-to-back SETUP packets were received for the current control transfer. On this interrupt, the application can decode the received SETUP data packet. **Values:** 0x0 (INACTIVE): No SETUP Phase Done 0x1 (ACTIVE): SETUP Phase Done
[2]	**AHBErr**	**AHB Error** Applies to IN and OUT endpoints. When there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address. **Values:** 0x0 (INACTIVE): No AHB Error Interrupt 0x1 (ACTIVE): AHB Error interrupt
[1]	**EPDisbld**	**Endpoint Disabled Interrupt** Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request.

		Values: 0x0 (INACTIVE): No Endpoint Disabled Interrupt 0x1 (ACTIVE): Endpoint Disabled Interrupt
[0]	**XferCompl**	**Transfer Completed Interrupt** Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is Set. **Values:** 0x0 (INACTIVE): No Transfer Complete Interrupt 0x1 (ACTIVE): Transfer Complete Interrupt

DOEPDMA7 Register (DOEPDMA7)

Register	Offset	R/W	Description	Reset Value
DOEPDMA7	HSUSBD_BA+0xbf4	R/W	Device OUT Endpoint 7 DMA Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMAAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMAAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMAAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMAAddr							

Bits	Description	
[31:0]	**DMAAddr**	**Holds the start address of the external memory for storing or fetching endpoint data** **Note:** For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address.

DOEPDMAB7 Register (DOEPDMAB7)

Register	Offset	R/W	Description	Reset Value
DOEPDMAB7	HSUSBD_BA+0xbfc	R	Device OUT Endpoint 7 Buffer Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMABufferAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMABufferAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMABufferAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMABufferAddr							

Bits	Description	
[31:0]	**DMABufferAddr**	**Holds the current buffer address This register is updated as and when the data transfer for the corresponding end point is in progress.**

DOEPCTL8 Register (DOEPCTL8)

Register	Offset	R/W	Description	Reset Value
DOEPCTL8	HSUSBD_BA+0xc00	R/W	Device Control OUT Endpoint 8 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
EPEna	EPDis	SetD1PID	SetD0PID	SNAK	CNAK	Reserved	
23	22	21	20	19	18	17	16
Reserved		Stall	Snp	EPType		NAKSts	DPID
15	14	13	12	11	10	9	8
USBActEP	Reserved				MPS		
7	6	5	4	3	2	1	0
MPS							

Bits	Description	
[31]	EPEna	**Endpoint Enable** Applies to IN and OUT endpoints. For IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. For OUT endpoint it indicates that the descriptor structure and data buffer to receive data is setup. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done Endpoint Disabled Transfer Completed **Note:** For control endpoints in DMA mode, this bit must be set for the controller to transfer SETUP data packets to the memory. This bit will not be cleared on Transfer Completed interrupt of the SETUP packet. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Enable Endpoint
[30]	EPDis	**Endpoint Disable** Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint. **Values:** 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Disable Endpoint
[29]	SetD1PID	**Set DATA1 PID** Applies to interrupt and bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA1. **Values:** 0x0 (DISABLED): Disables Set DATA1 PID or Do not force Odd Frame 0x1 (ENABLED): Set Endpoint Data PID to DATA1 or Sets EO_FrNum field to odd (micro)Frame

[28]	**SetD0PID**	**Set DATA0 PID** Applies to interrupt/bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA0. **Values:** 0x0 (DISABLED): Disables Set DATA0 PID or Do not force Even Frame 0x1 (ENABLED): Set Endpoint Data PID to DATA0 or Sets EO_FrNum field to odd (micro)Frame
[27]	**SNAK**	**Set NAK** A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for an endpoint after a SETUP packet is received on that endpoint. **Values:** 0x0 (INACTIVE): No Set NAK 0x1 (ACTIVE): Set NAK
[26]	**CNAK**	**Clear NAK** A write to this bit clears the NAK bit for the endpoint **Values:** 0x0 (INACTIVE): No Clear NAK 0x1 (ACTIVE): Clear NAK
[25:22]		**Reserved.**
[21]	**Stall**	**STALL Handshake** Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core. Applies to control endpoints only. The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (INACTIVE): STALL All non-active tokens 0x1 (ACTIVE): STALL All Active Tokens
[20]	**Snp**	Reserved.
[19:18]	**EPType**	**Endpoint Type** This is the transfer type supported by this logical endpoint. 2'b00: Control 2'b01: Isochronous 2'b10: Bulk 2'b11: Interrupt **Values:** 0x0 (CONTROL): Control 0x1 (ISOCHRONOUS): Isochronous 0x2 (BULK): Bulk 0x3 (INTERRUPT): Interrupt
[17]	**NAKSts**	**NAK Status (Read Only)** Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status. 1'b1: The core is transmitting NAK handshakes on this endpoint.

		When either the application or the core sets this bit: The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet. For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the TxFIFO. For isochronous IN endpoints: The core sends out a zero-length data packet, even if there data is available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. **Values:** 0x0 (NONNAK): The core is transmitting non-NAK handshakes based on the FIFO status 0x1 (NAK): The core is transmitting NAK handshakes on this endpoint
[16]	**DPID**	**Endpoint Data PID (Read Only)** Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. The applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID. 1'b0: DATA0 1'b1: DATA1 **Values:** 0x0 (INACTIVE): Endpoint Data PID not active 0x1 (ACTIVE): Endpoint Data PID active
[15]	**USBActEP**	**USB Active Endpoint** Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints (other than EP 0) after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit. **Values:** 0x0 (DISABLED): Not Active 0x1 (ENABLED): USB Active Endpoint
[14:11]	**Reserved**	**Reserved.**
[10:0]	**MPS**	**Maximum Packet Size** The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.

DOEPINT8 Register (DOEPINT8)

Register	Offset	R/W	Description	Reset Value
DOEPINT8	HSUSBD_BA+0xc08	R/W	Device OUT Endpoint 8 Interrupt Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	NYETIntrpt	NAKIntrpt	BbleErr	PktDrpSts	Reserved	BNAIntr	OutPktErr
7	6	5	4	3	2	1	0
Reserved	Back2BackSETup	StsPhseRcvd	OUTTknEPdis	SetUp	AHBErr	EPDisbld	XferCompl

Bits	Description	
[31:15]	**Reserved**	**Reserved.**
[14]	**NYETIntrpt**	**NYET Interrupt** The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint. **Values:** 0x0 (INACTIVE): No NYET interrupt 0x1 (ACTIVE): NYET Interrupt
[13]	**NAKIntrpt**	**NAK Interrupt** The core generates this interrupt when a NAK is transmitted or received by the device. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo. **Values:** 0x0 (INACTIVE): No NAK interrupt 0x1 (ACTIVE): NAK Interrupt
[12]	**BbleErr**	**NAK Interrupt** The core generates this interrupt when babble is received for the endpoint. **Values:** 0x0 (INACTIVE): No BbleErr interrupt 0x1 (ACTIVE): BbleErr interrupt
[11:10]	**Reserved**	**Reserved.**
[9]	**BNAIntr**	**Buffer Not Available Interrupt** The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as Host busy or DMA done **Values:** 0x0 (INACTIVE): No BNA interrupt 0x1 (ACTIVE): BNA interrupt
[8]	**OutPktErr**	**OUT Packet Error** Applies to OUT endpoints Only

		This interrupt is valid only when thresholding is enabled. This interrupt is asserted when the core detects an overflow or a CRC error for non-Isochronous OUT packet. **Values:** 0x0 (INACTIVE): No OUT Packet Error 0x1 (ACTIVE): OUT Packet Error
[7]	**Reserved**	**Reserved.**
[6]	**Back2BackSETup**	**Back-to-Back SETUP Packets Received** Applies to Control OUT endpoints only. This bit indicates that the core has received more than three back-to-back SETUP packets for this particular endpoint. For information about handling this interrupt, **Values:** 0x0 (INACTIVE): No Back-to-Back SETUP Packets Received 0x1 (ACTIVE): Back-to-Back SETUP Packets Received
[5]	**StsPhseRcvd**	**Status Phase Received for Control Write** This interrupt is valid only for Control OUT endpoints. This interrupt is generated only after the core has transferred all the data that the host has sent during the data phase of a control write transfer, to the system memory buffer. The interrupt indicates to the application that the host has switched from data phase to the status phase of a Control Write transfer. The application can use this interrupt to ACK or STALL the Status phase, after it has decoded the data phase. This is applicable only in Case of Scatter Gather DMA mode. **Values:** 0x0 (INACTIVE): No Status Phase Received for Control Write 0x1 (ACTIVE): Status Phase Received for Control Write
[4]	**OUTTknEPdis**	**OUT Token Received When Endpoint Disabled** Applies only to control OUT endpoints. Indicates that an OUT token was received when the endpoint was not yet enabled. This interrupt is asserted on the endpoint for which the OUT token was received. **Values:** 0x0 (INACTIVE): No OUT Token Received When Endpoint Disabled 0x1 (ACTIVE): OUT Token Received When Endpoint Disabled
[3]	**SetUp**	**SETUP Phase Done** Applies to control OUT endpoints only. Indicates that the SETUP phase for the control endpoint is complete and no more back-to-back SETUP packets were received for the current control transfer. On this interrupt, the application can decode the received SETUP data packet. **Values:** 0x0 (INACTIVE): No SETUP Phase Done 0x1 (ACTIVE): SETUP Phase Done
[2]	**AHBErr**	**AHB Error** Applies to IN and OUT endpoints. When there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address. **Values:** 0x0 (INACTIVE): No AHB Error Interrupt 0x1 (ACTIVE): AHB Error interrupt
[1]	**EPDisbld**	**Endpoint Disabled Interrupt** Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request.

		Values: 0x0 (INACTIVE): No Endpoint Disabled Interrupt 0x1 (ACTIVE): Endpoint Disabled Interrupt
[0]	**XferCompl**	**Transfer Completed Interrupt** Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is Set. **Values:** 0x0 (INACTIVE): No Transfer Complete Interrupt 0x1 (ACTIVE): Transfer Complete Interrupt

DOEPDMA8 Register (DOEPDMA8)

Register	Offset	R/W	Description	Reset Value
DOEPDMA8	HSUSBD_BA+0xc14	R/W	Device OUT Endpoint 8 DMA Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMAAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMAAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMAAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMAAddr							

Bits	Description	
[31:0]	**DMAAddr**	**Holds the start address of the external memory for storing or fetching endpoint data** **Note:** For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address.

DOEPDMAB8 Register (DOEPDMAB8)

Register	Offset	R/W	Description	Reset Value
DOEPDMAB8	HSUSBD_BA+0xc1c	R	Device OUT Endpoint 8 Buffer Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMABufferAddr							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DMABufferAddr							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMABufferAddr							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMABufferAddr							

Bits	Description	
[31:0]	**DMABufferAddr**	**Holds the current buffer address** This register is updated as and when the data transfer for the corresponding end point is in progress.

PCGCCTL Register (PCGCCTL)

Register	Offset	R/W	Description	Reset Value
PCGCCTL	HSUSBD_BA+0xe00	R/W	Power and Clock Gating Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
L1Suspended	PhySleep	Enbl_L1Gating	Reserved				StopPclk

Bits	Description	
[31:8]	**Reserved**	**Reserved.**
[7]	**L1Suspended**	**L1 Deep Sleep (Read Only)** Indicates that the PHY is in deep sleep when in L1 state. **Values:** 0x0 (INACTIVE): Non Deep Sleep 0x1 (ACTIVE): Deep Sleep
[6]	**PhySleep**	**PHY In Sleep (Read Only)** Indicates that the PHY is in Sleep State. **Values:** 0x0 (INACTIVE): Phy not in Sleep state 0x1 (ACTIVE): Phy in Sleep state
[5]	**Enbl_L1Gating**	**Enable Sleep Clock Gating** If this bit is set, core internal clock gating is enabled sleep state if utmi_l1_suspend_n cannot be asserted by the core. The PHY clock will not be gated in sleep state if Enbl_L1Gating is not set. **Values:** 0x0 (DISABLED): The PHY clock is not gated in Sleep state 0x1 (ENABLED): The Core internal clock gating is enabled in Sleep state
[4:1]	**Reserved**	**Reserved.**
[0]	**StopPclk**	**Stop Pclk** The application sets this bit to stop the PHY clock (phy_clk) when the USB is suspended, the session is not valid, or the device is disconnected. The application clears this bit when the USB is resumed or a new session starts. **Values:** 0x0 (DISABLED): Disable Stop Pclk 0x1 (ENABLED): Enable Stop Pclk

PCGCCTL1 Register (PCGCCTL1)

Register	Offset	R/W	Description	Reset Value
PCGCCTL1	HSUSBD_BA+0xe04	R/W	Power and Clock Gating Control Register1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				RAMGateEN	counter		GateEN

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3]	**RAMGateEN**	**RAM Clock Gating Enable** The application programs RAMGateEn to enable RAM Clock Gating. 1b0: Disable RAM Clock Gating (Default) 1b1: Enable RAM Clock Gating **Values:** 0x0 (DISABLED): RAM Clock Gating disable 0x1 (ENABLED): RAM Clock Gating enable
[2:1]	**counter**	**Count to Gate Clock** CntGateClk indicates to the Controller how many PHY Clock cycles and AHB Clock cycles of 'IDLE' (no activity) the Controller will wait for before Gating the respective PHY and AHB clocks internal to the Controller. 2b00: 64 clocks (Default). 2b01: 128 clocks **Values:** 0x0 (CLOCKS64): CLOCKS64 0x1 (CLOCKS128): CLOCKS128
[0]	**GateEN**	**Enable Active Clock Gating** The application programs GateEn to enable Active Clock Gating feature for the PHY and AHB clocks. 1b0: Disable Active Clock Gating (Default) 1b1: Enable Active Clock Gating **Values:** 0x0 (DISABLED): ACG disable 0x1 (ENABLED): ACG enable

6.31 USB 2.0 Host Controller (HSUSBH)

6.31.1 Overview

This chip is equipped with two individual USB 2.0 HS Host Controllers (HSUSBH0, HSUSBH1) that support Enhanced Host Controller Interface (EHCI) Specification, a register-level description of a host controller, to manage the devices and data transfer of Universal Serial Bus (USB).

Each HSUSBH supports an integrated Root Hub with a USB port, a DMA for real-time data transfer between system memory and USB bus, port power control and port overcurrent detection.

Each HSUSBH is responsible for detecting the connect and disconnect of USB devices, managing data transfer, collecting status and activity of USB bus, providing power control and detecting overcurrent of attached USB devices.

6.31.2 Features

- Compliant with Universal Serial Bus (USB) Specification Revision 2.0.
- Supports Enhanced Host Controller Interface (EHCI) Specification Revision 1.0.
- Supports high-speed (480Mbps) USB devices.
- Supports Control, Bulk, Interrupt, Isochronous and Split transfers.
- Supports an integrated Root Hub.
- Supports a port routing logic to route full/low speed device to OHCI controller.
- Supports port power control and port overcurrent detection.
- Supports DMA for real-time data transfer.
- Supports dual-role switching with the USB 2.0 Device Controller (for HSUSBH0 only).

6.31.3 Block Diagram

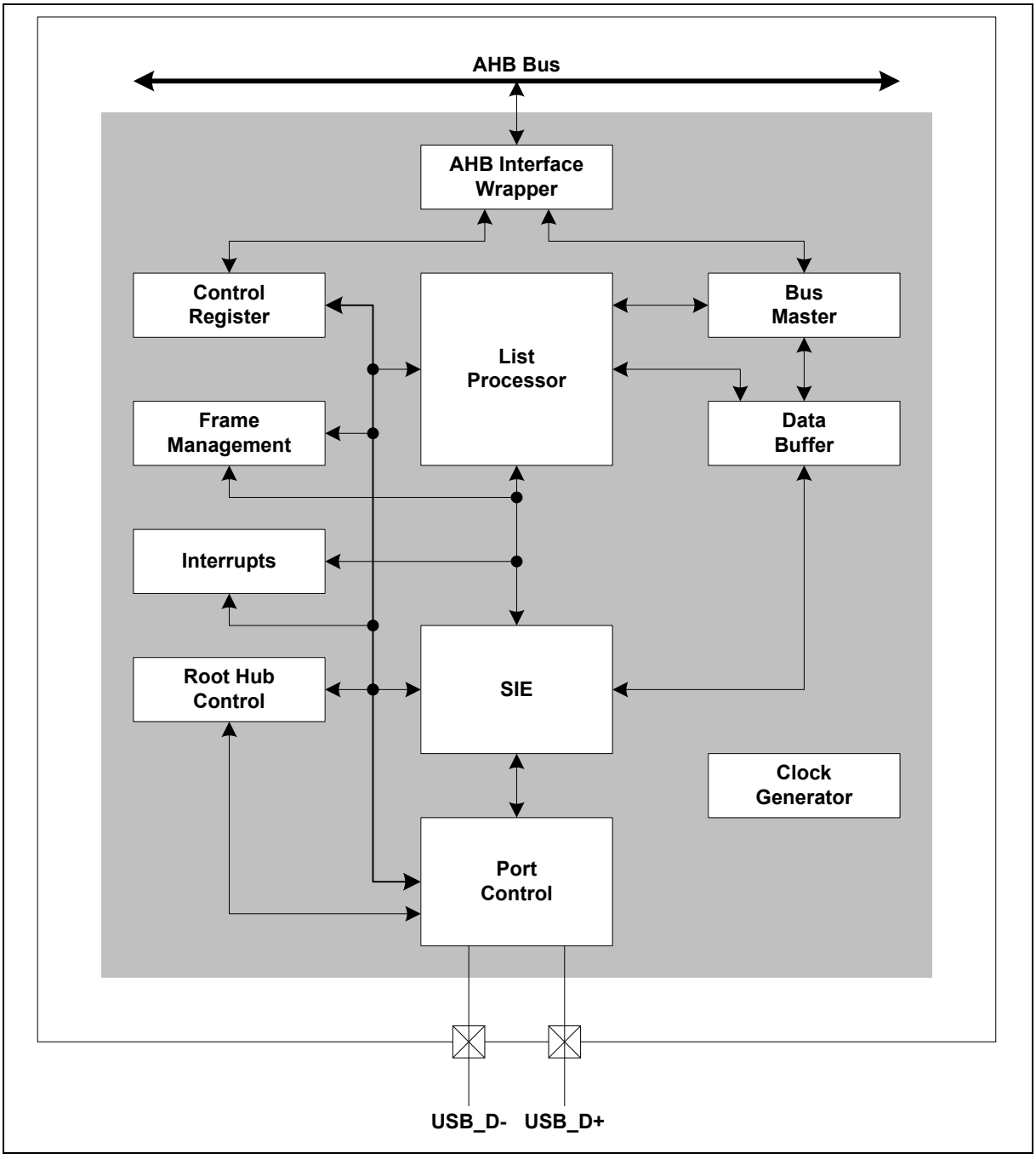


Figure 6.31-1 USB 2.0 HS Host Controller Block Diagram

6.31.4 Basic Configuration

6.31.4.1 HSUSBH EHCI Controller Basic Configuration

- Clock Source Configuration
  - Enable HSUSBH0 AHB clock in HUSBH0EN (CLK_SYSCLK0[21]).
  - Enable HSUSBH1 AHB clock in HUSBH1EN (CLK_SYSCLK0[22]).
  - Enable HSUBSH0 engine clock by the following setting:

        - Set PHY0SUSPEND (SYS_USBPMISCR[1]) to 1.
        - Set PHY0POR (SYS_USBPMISCR[0]) to 0.
        - Set PHY0COMN (SYS_USBPMISCR[2]) to 0.
    - Enable HSUBSH1 engine clock by the following setting:
        - Set PHY1SUSPEND (SYS_USBPMISCR[17]) to 1.
        - Set PHY1POR (SYS_USBPMISCR[16]) to 0.
        - Set PHY1COMN (SYS_USBPMISCR[18]) to 0.
- Overcurrent Detection Level Configuration
    - Set HSUSBH0 and HSUSBH1 overcurrent detection signal to high-active/low-active by setting UHOVRCURH (SYS_MISCFCR0[12]) to 1/0 according to application conditions.
- Dual-role Switching Configuration
    - HSUSBH0 can be configured to act as a USB 2.0 Host Controller/USB 2.0 Device Controller by inputting 1/0 signal to HSUSB0_ID pin.
    - HSUSBH1 doesn't support dual-role switching and can only act as a USB 2.0 Host Controller.
- Reset Configuration
    - Reset HSUSBH0 controller in HSUSBH0RST (SYS_IPRST0[20]).
    - Reset HSUSBH1 controller in HSUSBH1RST (SYS_IPRST0[21]).

6.31.5 Functional Description

6.31.5.1 EHCI Controller

The EHCI is interfaced with the system through AHB interface. Whenever the CPU wants to initiate a register read or register write, it uses the AHB slave I/F signals and performs the necessary operation (register read or register write). The CPU acts as a bus master, having initiated this transfer. At that time, EHCI acts as a target and responds to the transfer initiated by the system software. For example, if the CPU wants to write into one of the memory mapped registers of EHCI, it says the address and value to be written into that addressed register. EHCI targets the register by using that address and fills the register with the value specified by the software. If it is a register read, EHCI gets the value from the addressed register and puts it on the system bus. Likewise, when the EHCI wants to perform a data transfer, it acts as a master and initiates a data transfer. At that time, the system memory acts as a bus target. EHCI, as a master can perform two types of data transfers, from EHCI to the system memory and from system memory to the EHCI. When the EHCI wants the data to be moved from the downstream USB2.0 device to the system memory, it initiates a memory write transfer by accessing the memory interfacing signals. EHCI writes the control word (write), data and data count to be moved to the system memory. The memory controller accepts the data and moves it to the memory. If the data has to be moved from memory to the downstream device, the EHCI issue a read transfer to system bus. The memory controller gives data through the memory interfacing signals. EHCI accepts the data and moves them to the downstream device.

For more information and the detailed standard register interface of EHCI, please refer to the *Enhanced Host Controller Interface Specification for Universal Serial Bus*, Revision 1.0.

6.31.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
HSUSBH Base Address: **HSUSBHn_BA = 0x4014_0000 + (0x8_0000 * n)** **n= 0, 1**				
EHCVNR	HSUSBHn_BA+0x00	R	EHCI Version Number Register	0x0100_0010
EHCSPR	HSUSBHn_BA+0x04	R	EHCI Structural Parameters Register	0x0000_1111
EHCCPR	HSUSBHn_BA+0x08	R	EHCI Capability Parameters Register	0x0000_A010
UCMDR	HSUSBHn_BA+0x10	R/W	USB Command Register	0x0008_0000
USTSR	HSUSBHn_BA+0x14	R/W	USB Status Register	0x0000_1000
UIENR	HSUSBHn_BA+0x18	R/W	USB Interrupt Enable Register	0x0000_0000
UFINDR	HSUSBHn_BA+0x1c	R/W	USB Frame Index Register	0x0000_0000
UPFLBAR	HSUSBHn_BA+0x24	R/W	USB Periodic Frame List Base Address Register	0x0000_0000
UCALAR	HSUSBHn_BA+0x28	R/W	USB Current Asynchronous List Address Register	0x0000_0000
UCFGR	HSUSBHn_BA+0x50	R/W	USB Configure Flag Register	0x0000_0000
UPSCR	HSUSBHn_BA+0x54	R/W	USB Port Status and Control Register	0x0000_2000

6.31.7 Register Description

EHCI Version Number Register (EHCVNR)

Register	Offset	R/W	Description	Reset Value
EHCVNR	HSUSBHn_BA+0x00	R	EHCI Version Number Register	0x0100_0010

31	30	29	28	27	26	25	24
VERSION							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
VERSION							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CRLEN							

Bits	Description	
[31:16]	**VERSION**	**Host Controller Interface Version Number** This is a two-byte register containing a BCD encoding of the EHCI revision number supported by this host controller. The most significant byte of this register represents a major revision and the least significant byte is the minor revision.
[15:8]	**Reserved**	Reserved.
[7:0]	**CRLEN**	**Capability Registers Length** This register is used as an offset to add to register base to find the beginning of the Operational Register Space defined in EHCI specification.

EHCI Structural Parameters Register (EHCSPR)

Register	Offset	R/W	Description	Reset Value
EHCSPR	HSUSBHn_BA+0x04	R	EHCI Structural Parameters Register	0x0000_1111

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
N_CC				N_PCC			
7	6	5	4	3	2	1	0
Reserved			PPC	N_PORTS			

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:12]	**N_CC**	**Number of Companion Controller** This field indicates the number of companion controllers associated with this USB 2.0 host controller. A zero in this field indicates there are no companion host controllers. Port-ownership hand-off is not supported. Only high-speed devices are supported on the host controller root ports. A value larger than zero in this field indicates there are companion USB 1.1 host controller(s). Port-ownership hand-offs are supported. High, Full- and Low-speed devices are supported on the host controller root ports.
[11:8]	**N_PCC**	**Number of Ports Per Companion Controller** This field indicates the number of ports supported per companion host controller. It is used to indicate the port routing configuration to system software. For example, if N_PORTS has a value of 6 and N_CC has a value of 2 then N_PCC could have a value of 3. The convention is that the first N_PCC ports are assumed to be routed to companion controller 1, the next N_PCC ports to companion controller 2, etc. In the previous example, the N_PCC could have been 4, where the first 4 are routed to companion controller 1 and the last two are routed to companion controller 2. The number in this field must be consistent with N_PORTS and N_CC.
[7:5]	**Reserved**	Reserved.
[4]	**PPC**	**Port Power Control** This field indicates whether the host controller implementation includes port power control. A one in this bit indicates the ports have port power switches. A zero in this bit indicates the port do not have port power stitches. The value of this field affects the functionality of the Port Power field in each port status and control register.
[3:0]	**N_PORTS**	**Number of Physical Downstream Ports** This field specifies the number of physical downstream ports implemented on this host controller. The value of this field determines how many port registers are addressable in the Operational Register Space (see Table 2-8 of EHCI spec.). Valid values are in the range of 1H to FH. A zero in this field is undefined.

EHCI Capability Parameters Register (EHCCPR)

Register	Offset	R/W	Description	Reset Value
EHCCPR	HSUSBHn_BA+0x08	R	EHCI Capability Parameters Register	0x0000_A010

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
EECP							
7	6	5	4	3	2	1	0
IST				Reserved	ASPC	PFLF	AC64

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:8]	**EECP**	**EHCI Extended Capabilities Pointer** 0 = No extended capabilities are implemented. 1 = Extended capabilities are implemented.
[7:4]	**IST**	**Isochronous Scheduling Threshold** This field indicates, relative to the current position of the executing host controller, where software can reliably update the isochronous schedule. When bit [7] is zero, the value of the least significant 3 bits indicates the number of micro-frames a host controller can hold a set of isochronous data structures (one or more) before flushing the state. When bit [7] is a one, then host software assumes the host controller may cache an isochronous data structure for an entire frame.
[3]	**Reserved**	Reserved.
[2]	**ASPC**	**Asynchronous Schedule Park Capability** 0 = This EHCI host controller doesn't support park feature of high-speed queue heads in the Asynchronous Schedule. 1 = This EHCI host controller support park feature of high-speed queue heads in the Asynchronous Schedule.
[1]	**PFLF**	**Programmable Frame List Flag** 0 = System software must use a frame list length of 1024 elements with this EHCI host controller. 1 = System software can specify and use a smaller frame list and configure the host controller via the USB Command Register.
[0]	**AC64**	**64-bit Addressing Capability** 0 = Data structure using 32-bit address memory pointers. 1 = Data structure using 64-bit address memory pointers.

USB Command Register (UCMDR)

Register	Offset	R/W	Description	Reset Value
UCMDR	HSUSBHn_BA+0x10	R/W	USB Command Register	0x0008_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
ITC							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	IAAD	ASEN	PSEN	FLSZ		HCRST	RUN

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**ITC**	**Interrupt Threshold Control (Read/Write)** This field is used by system software to select the maximum rate at which the host controller will issue interrupts. The only valid values are defined below. If software writes an invalid value to this register, the results are undefined. Value Maximum Interrupt Interval 0x00 = Reserved. 0x01 = 1 micro-frame. 0x02 = 2 micro-frames. 0x04 = 4 micro-frames. 0x08 = 8 micro-frames (default, equates to 1 ms). 0x10 = 16 micro-frames (2 ms). 0x20 = 32 micro-frames (4 ms). 0x40 = 64 micro-frames (8 ms). Any other value in this register yields undefined results. Software modifications to this bit while HCHalted bit is equal to zero results in undefined behavior.
[15:7]	**Reserved**	Reserved.
[6]	**IAAD**	**Interrupt on Asynchronous Advance Doorbell (Read/Write)** This bit is used as a doorbell by software to tell the host controller to issue an interrupt the next time it advances asynchronous schedule. Software must write a 1 to this bit to ring the doorbell. When the host controller has evicted all appropriate cached schedule state, it sets the Interrupt on Asynchronous Advance status bit in the USBSTS register. If the Interrupt on Asynchronous Advance Enable bit in the USBINTR register is a one then the host controller will assert an interrupt at the next interrupt threshold. The host controller sets this bit to a zero after it has set the Interrupt on Asynchronous Advance status bit in the USBSTS register to a one. Software should not write a one to this bit when the asynchronous schedule is disabled. Doing so will yield undefined results.

[5]	**ASEN**	**Asynchronous Schedule Enable (Read/Write)** This bit controls whether the host controller skips processing the Asynchronous Schedule. Values mean: 0 = Do not process the Asynchronous Schedule. 1 = Use the ASYNCLISTADDR register to access the Asynchronous Schedule.
[4]	**PSEN**	**Periodic Schedule Enable (Read/Write)** This bit controls whether the host controller skips processing the Periodic Schedule. Values mean: 0 = Do not process the Periodic Schedule. 1 = Use the PERIODICLISTBASE register to access the Periodic Schedule.
[3:2]	**FLSZ**	**Frame List Size (Read Only)** This field is R/W only if Programmable Frame List Flag in the HCCPARAMS registers is set to a one. This field specifies the size of the frame list. The size the frame list controls which bits in the Frame Index Register should be used for the Frame List Current index. Values mean: 00 = 1024 elements (4096 bytes) Default value. 01 = 512 elements (2048 bytes). 10 = 256 elements (1024 bytes) – for resource-constrained environment. 11 = Reserved.
[1]	**HCRST**	**Host Controller Reset (Read/Write)** This control bit is used by software to reset the host controller. The effects of this on Root Hub registers are similar to a Chip Hardware Reset. When software writes a one to this bit, the Host Controller resets its internal pipelines, timers, counters, state machines, etc. to their initial value. Any transaction currently in progress on USB is immediately terminated. A USB reset is not driven on downstream ports. All operational registers, including port registers and port state machines are set to their initial values. Port ownership reverts to the companion host controller(s), with the side effects. Software must reinitialize the host controller in order to return the host controller to an operational state. This bit is set to zero by the Host Controller when the reset process is complete. Software cannot terminate the reset process early by writing a zero to this register. Software should not set this bit to a one when the HCHalted bit in the USBSTS register is a zero. Attempting to reset an actively running host controller will result in undefined behavior.
[0]	**RUN**	**Run/Stop (Read/Write)** When set to a 1, the Host Controller proceeds with execution of the schedule. The Host Controller continues execution as long as this bit is set to a 1. When this bit is set to 0, the Host Controller completes the current and any actively pipelined transactions on the USB and then halts. The Host Controller must halt within 16 micro-frames after software clears the Run bit. The HC Halted bit in the status register indicates when the Host Controller has finished its pending pipelined transactions and has entered the stopped state. Software must not write a one to this field unless the host controller is in the Halted state (i.e. HCHalted in the USBSTS register is a one). Doing so will yield undefined results. 0 = Stop. 1 = Run.

USB Status Register (USTSR)

Register	Offset	R/W	Description	Reset Value
USTSR	HSUSBHn_BA+0x14	R/W	USB Status Register	0x0000_1000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
ASS	PSS	RECLA	HCHalted	Reserved			
7	6	5	4	3	2	1	0
Reserved		IAA	HSERR	FLR	PCD	UERRINT	USBINT

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15]	**ASS**	**Asynchronous Schedule Status (Read Only)** The bit reports the current real status of the Asynchronous Schedule. If this bit is a zero then the status of them Asynchronous Schedule is disabled. If this bit is a one then the status of the Asynchronous Schedule is enabled. The Host Controller is not required to immediately disable or enable the Asynchronous Schedule when software transitions the Asynchronous Schedule Enable bit in the USBCMD register. When this bit and the Asynchronous Schedule Enable bit are the same value, the Asynchronous Schedule is either enabled (1) or disabled (0).
[14]	**PSS**	**Periodic Schedule Status (Read Only)** The bit reports the current real status of the Periodic Schedule. If this bit is a zero then the status of the Periodic Schedule is disabled. If this bit is a one then the status of the Periodic Schedule is enabled. The Host Controller is not required to immediately disable or enable the Periodic Schedule when software transitions the Periodic Schedule Enable bit in the USBCMD register. When this bit and the Periodic Schedule Enable bit are the same value, the Periodic Schedule is either enabled (1) or disabled (0).
[13]	**RECLA**	**Reclamation (Read Only)** This is a read-only status bit, which is used to detect an empty asynchronous schedule.
[12]	**HCHalted**	**HCHalted (Read Only)** This bit is a zero whenever the Run/Stop bit is a one. The Host Controller sets this bit to one after it has stopped executing as a result of the Run/Stop bit being set to 0, either by software or by the Host Controller hardware (e.g. internal error).
[11:6]	**Reserved**	Reserved.
[5]	**IAA**	**Interrupt on Asynchronous Advance** System software can force the host controller to issue an interrupt the next time the host controller advances the asynchronous schedule by writing a one to the Interrupt on Asynchronous Advance Doorbell bit in the USBCMD register. This status bit indicates the assertion of that interrupt source. **Note:** This bit is cleared by writing 1 to it.
[4]	**HSERR**	**Host System Error** The Host Controller sets this bit to 1 when a serious error occurs during a host system access involving the Host Controller module. **Note:** This bit is cleared by writing 1 to it.

[3]	**FLR**	**Frame List Rollover** The Host Controller sets this bit to a one when the Frame List Index rolls over from its maximum value to zero. The exact value at which the rollover occurs depends on the frame list size. For example, if the frame list size (as programmed in the Frame List Size field of the USBCMD register) is 1024, the Frame Index Register rolls over every time FRINDEX[13] toggles. Similarly, if the size is 512, the Host Controller sets this bit to a one every time FRINDEX[12] toggles. **Note:** This bit is cleared by writing 1 to it.
[2]	**PCD**	**Port Change Detect** The Host Controller sets this bit to a one when any port for which the Port Owner bit is set to zero has a change bit transition from a zero to a one or a Force Port Resume bit transition from a zero to a one as a result of a J-K transition detected on a suspended port. This bit will also be set as a result of the Connect Status Change being set to a one after system software has relinquished ownership of a connected port by writing a one to a port's Port Owner bit. This bit is allowed to be maintained in the Auxiliary power well. Alternatively, it is also acceptable that on a D3 to D0 transition of the EHCI HC device, this bit is loaded with the OR of all of the PORTSC change bits (including: Force port resume, overcurrent change, enable/disable change and connect status change). **Note:** This bit is cleared by writing 1 to it.
[1]	**UERRINT**	**USB Error Interrupt** The Host Controller sets this bit to 1 when completion of a USB transaction results in an error condition (e.g., error counter underflow). If the TD on which the error interrupt occurred also had its IOC bit set, both this bit and USBINT bit are set. **Note:** This bit is cleared by writing 1 to it.
[0]	**USBINT**	**USB Interrupt** The Host Controller sets this bit to 1 on the completion of a USB transaction, which results in the retirement of a Transfer Descriptor that had its IOC bit set. The Host Controller also sets this bit to 1 when a short packet is detected (actual number of bytes received was less than the expected number of bytes). **Note:** This bit is cleared by writing 1 to it.

USB Interrupt Enable Register (UIENR)

Register	Offset	R/W	Description	Reset Value
UIENR	HSUSBHn_BA+0x18	R/W	USB Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		IAAEN	HSERREN	FLREN	PCIEN	UERRIEN	USBIEN

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**IAAEN**	**Interrupt on Asynchronous Advance Enable or Disable Bit** When this bit is a one, and the Interrupt on Asynchronous Advance bit in the USBSTS register is a one, the host controller will issue an interrupt at the next interrupt threshold. The interrupt is acknowledged by software clearing the Interrupt on Asynchronous Advance bit. 0 = Interrupt on Asynchronous Advance Disabled. 1 = Interrupt on Asynchronous Advance Enabled.
[4]	**HSERREN**	**Host System Error Enable or Disable Bit** When this bit is a one, and the Host System Error Status bit in the USBSTS register is a one, the host controller will issue an interrupt. The interrupt is acknowledged by software clearing the Host System Error bit. 0 = Host System Error interrupt Disabled. 1 = Host System Error interrupt Enabled.
[3]	**FLREN**	**Frame List Rollover Enable or Disable Bit** When this bit is a one, and the Frame List Rollover bit in the USBSTS register is a one, the host controller will issue an interrupt. The interrupt is acknowledged by software clearing the Frame List Rollover bit. 0 = Frame List Rollover interrupt Disabled. 1 = Frame List Rollover interrupt Enabled.
[2]	**PCIEN**	**Port Change Interrupt Enable or Disable Bit** When this bit is a one, and the Port Change Detect bit in the USBSTS register is a one, the host controller will issue an interrupt. The interrupt is acknowledged by software clearing the Port Change Detect bit. 0 = Port Change interrupt Disabled. 1 = Port Change interrupt Enabled.
[1]	**UERRIEN**	**USB Error Interrupt Enable or Disable Bit** When this bit is a one, and the USBERRINT bit in the USBSTS register is a one, the host controller will issue an interrupt at the next interrupt threshold. The interrupt is acknowledged by software clearing the USBERRINT bit. 0 = USB Error interrupt Disabled. 1 = USB Error interrupt Enabled.

[0]	**USBIEN**	**USB Interrupt Enable or Disable Bit** When this bit is a one, and the USBINT bit in the USBSTS register is a one, the host controller will issue an interrupt at the next interrupt threshold. The interrupt is acknowledged by software clearing the USBINT bit. 0 = USB interrupt Disabled. 1 = USB interrupt Enabled.

USB Frame Index Register (UFINDR)

Register	Offset	R/W	Description	Reset Value
UFINDR	HSUSBHn_BA+0x1c	R/W	USB Frame Index Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		FI					
7	6	5	4	3	2	1	0
FI							

Bits	Description	
[31:14]	**Reserved**	Reserved.
[13:0]	**FI**	**Frame Index** The value in this register increment at the end of each time frame (e.g. micro-frame). Bits [N:3] are used for the Frame List current index. This means that each location of the frame list is accessed 8 times (frames or micro-frames) before moving to the next index. The following illustrates values of N based on the value of the Frame List Size field in the USBCMD register. **FLSZ (UCMDR[3:2] Number Elements N** 0x0 1024 12 0x1 512 11 0x2 256 10 0x3 Reserved

USB Periodic Frame List Base Address Register (UPFLBAR)

Register	Offset	R/W	Description	Reset Value
UPFLBAR	HSUSBHn_BA+0x24	R/W	USB Periodic Frame List Base Address Register	0x0000_0000

31	30	29	28	27	26	25	24
BADDR							
23	22	21	20	19	18	17	16
BADDR							
15	14	13	12	11	10	9	8
BADDR				Reserved			
7	6	5	4	3	2	1	0
Reserved							

Bits	Description	
[31:12]	**BADDR**	**Base Address** These bits correspond to memory address signals [31:12], respectively.
[11:0]	**Reserved**	Reserved.

USB Current Asynchronous List Address Register (UCALAR)

Register	Offset	R/W	Description	Reset Value
UCALAR	HSUSBHn_BA+0x28	R/W	USB Current Asynchronous List Address Register	0x0000_0000

31	30	29	28	27	26	25	24
LPL							
23	22	21	20	19	18	17	16
LPL							
15	14	13	12	11	10	9	8
LPL							
7	6	5	4	3	2	1	0
LPL			Reserved				

Bits	Description	
[31:5]	**LPL**	**Link Pointer Low** These bits correspond to memory address signals [31:5], respectively. This field may only reference a Queue Head (QH).
[4:0]	**Reserved**	Reserved.

USB Configure Flag Register (UCFGR)

Register	Offset	R/W	Description	Reset Value
UCFGR	HSUSBHn_BA+0x50	R/W	USB Configure Flag Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							CF

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**CF**	**Configure Flag** Host software sets this bit as the last action in its process of configuring the Host Controller. This bit controls the default port-routing control logic. Bit values and side-effects are listed below. 0 = Port routing control logic default-routes each port to an implementation dependent classic host controller. 1 = Port routing control logic default-routes all ports to this host controller.

USB Port Status and Control Register (UPSCR)

Register	Offset	R/W	Description	Reset Value
UPSCR	HSUSBHn_BA+0x54	R/W	USB Port Status and Control Register	0x0000_2000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				PTC			
15	14	13	12	11	10	9	8
Reserved		PO	PP	LSTS		Reserved	PRST
7	6	5	4	3	2	1	0
SUSPEND	FPR	OCC	OCA	PEC	PE	CSC	CCS

Bits	Description	
[31:20]	**Reserved**	Reserved.
[19:16]	**PTC**	**Port Test Control (Read/Write)** When this field is zero, the port is NOT operating in a test mode. A non-zero value indicates that it is operating in test mode and the specific test mode is indicated by the specific value. The encoding of the test mode bits are (0x6 ~ 0xF are reserved): Bits Test Mode 0x0 = Test mode not enabled. 0x1 = Test J_STATE. 0x2 = Test K_STATE. 0x3 = Test SE0_NAK. 0x4 = Test Packet. 0x5 = Test FORCE_ENABLE.
[15:14]	**Reserved**	Reserved.
[13]	**PO**	**Port Owner (Read/Write)** This bit unconditionally goes to a 0b when the Configured bit in the CONFIGFLAG register makes a 0 to 1 transition. This bit unconditionally goes to 1 whenever the Configured bit is zero. System software uses this field to release ownership of the port to a selected host controller (in the event that the attached device is not a high-speed device). Software writes a one to this bit when the attached device is not a high-speed device. A one in this bit means that a companion host controller owns and controls the port.
[12]	**PP**	**Port Power** Host controller has port power control switches. This bit represents the Current setting of the switch (0 = off, 1 = on). When power is not available on a port (i.e. PP equals a 0), the port is nonfunctional and will not report attaches, detaches, etc. When an overcurrent condition is detected on a powered port and PPC is a one, the PP bit in each affected port may be transitioned by the host controller from a 1 to 0 (removing power from the port).

[11:10]	**LSTS**	**Line Status (Read Only)** These bits reflect the current logical levels of the D+ (bit 11) and D- (bit 10) signal lines. These bits are used for detection of low-speed USB devices prior to the port reset and enable sequence. This field is valid only when the port enable bit is zero and the current connect status bit is set to a one. The encoding of the bits are: Bits[11:10] USB State Interpretation 00 = SE0 Not Low-speed device, perform EHCI reset. 01 = K-state Low-speed device, release ownership of port. 10 = J-state Not Low-speed device, perform EHCI reset. 11 = Undefined Not Low-speed device, perform EHCI reset. This value of this field is undefined if Port Power is zero.
[9]	**Reserved**	Reserved.
[8]	**PRST**	**Port Reset (Read/Write)** When software writes a one to this bit (from a zero), the bus reset sequence as defined in the USB Specification Revision 2.0 is started. Software writes a zero to this bit to terminate the bus reset sequence. Software must keep this bit at a one long enough to ensure the reset sequence, as specified in the USB Specification Revision 2.0, completes. **Note:** when software writes this bit to a one, it must also write a zero to the Port Enable bit. Note that when software writes a zero to this bit there may be a delay before the bit status changes to a zero. The bit status will not read as a zero until after the reset has completed. If the port is in high-speed mode after reset is complete, the host controller will automatically enable this port (e.g. set the Port Enable bit to a one). A host controller must terminate the reset and stabilize the state of the port within 2 milliseconds of software transitioning this bit from a one to a zero. For example: if the port detects that the attached device is high-speed during reset, then the host controller must have the port in the enabled state within 2ms of software writing this bit to a zero. The HCHalted bit in the USBSTS register should be a zero before software attempts to use this bit. The host controller may hold Port Reset asserted to a one when the HCHalted bit is a one. This field is zero if Port Power is zero. 0 = Port is not in Reset. 1 = Port is in Reset.
[7]	**SUSPEND**	**Suspend (Read/Write)** Port Enabled Bit and Suspend bit of this register define the port states as follows: 00 = Port Disable. 01 = Port Disable. 10 = Port Enable. 11 = Port Suspend. When in suspend state, downstream propagation of data is blocked on this port, except for port reset. The blocking occurs at the end of the current transaction, if a transaction was in progress when this bit was written to 1. In the suspend state, the port is sensitive to resume detection. Note that the bit status does not change until the port is suspended and that there may be a delay in suspending a port if there is a transaction currently in progress on the USB. A write of zero to this bit is ignored by the host controller. The host controller will unconditionally set this bit to a zero when: Software sets the Force Port Resume bit to a zero (from a one). Software sets the Port Reset bit to a one (from a zero). If host software sets this bit to a one when the port is not enabled (i.e. Port enabled bit is a zero) the results are undefined. This field is zero if Port Power is zero. 0 = Port not in suspend state. 1 = Port in suspend state.

[6]	**FPR**	**Force Port Resume (Read/Write)** This functionality defined for manipulating this bit depends on the value of the Suspend bit. For example, if the port is not suspended (Suspend and Enabled bits are a one) and software transitions this bit to a one, then the effects on the bus are undefined. Software sets this bit to a 1 to drive resume signaling. The Host Controller sets this bit to a 1 if a J-to-K transition is detected while the port is in the Suspend state. When this bit transitions 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. If software sets this bit to a one, the host controller must not set the Port Change Detect bit. Note that when the EHCI controller owns the port, the resume sequence follows the defined sequence documented in the USB Specification Revision 2.0. The resume signaling (Full-speed 'K') is driven on the port as long as this bit remains a one. Software must appropriately time the Resume and set this bit to a zero when the appropriate amount of time has elapsed. Writing a zero (from one) causes the port to return to high-speed mode (forcing the bus below the port into a high-speed idle). This bit will remain a one until the port has switched to the high-speed idle. The host controller must complete this transition within 2 milliseconds of software setting this bit to a zero. This field is zero if Port Power is zero. 0 = No resume (K-state) detected/driven on port. 1 = Resume detected/driven on port.
[5]	**OCC**	**Overcurrent Change** 0 = Default. 1 = This bit gets set to a one when there is a change to Overcurrent Active. Software clears this bit by writing a one to this bit position. **Note:** This bit is cleared by writing 1 to it.
[4]	**OCA**	**Overcurrent Active (Read Only)** This bit will automatically transition from a one to a zero when the overcurrent condition is removed. 0 = This port does not have an overcurrent condition. 1 = This port currently has an overcurrent condition.
[3]	**PEC**	**Port Enable/Disable Change** For the root hub, this bit gets set to a one only when a port is disabled due to the appropriate conditions existing at the EOF2 point (See Chapter 11 of the USB Specification for the definition of a Port Error). Software clears this bit by writing a 1 to it. This field is zero if Port Power is zero. 0 = No change. 1 = Port enabled/disabled status has changed. **Note:** This bit is cleared by writing 1 to it.
[2]	**PE**	**Port Enabled/Disabled (Read/Write)** 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. The host controller will only set this bit to a one when the reset sequence determines that the attached device is a high-speed device. Ports can be disabled by either a fault condition (disconnect event or other fault condition) or by host software. Note that the bit status does not change until the port state actually changes. There may be a delay in disabling or enabling a port due to other host controller and bus events. When the port is disabled, downstream propagation of data is blocked on this port, except for reset. This field is zero if Port Power is zero. 0 = Port Disabled. 1 = Port Enabled.

[1]	**CSC**	**Connect Status Change (Read/Write)** Indicates a change has occurred in the port's Current Connect Status. The host controller sets this bit for all changes to the port device connect status, even if system software has not cleared an existing connect status change. For example, the insertion status changes twice before system software has cleared the changed condition, hub hardware will be "setting" an already-set bit (i.e., the bit will remain set).Software sets this bit to 0 by writing a 1 to it. This field is zero if Port Power is zero. 0 = No change. 1 = Change in Current Connect Status.
[0]	**CCS**	**Current Connect Status (Read Only)** This value reflects the current state of the port, and may not correspond directly to the event that caused the Connect Status Change bit (Bit 1) to be set. This field is zero if Port Power is zero. 0 = No device is present. 1 = Device is present on port.

6.32 USB 1.1 Host Controller (USBH)

6.32.1 Overview

This chip is equipped with two USB 1.1 FS Host Controller (USBH) that supports Open Host Controller Interface (OpenHCI, OHCI) Specification, a register-level description of a host controller, to manage the devices and data transfer of Universal Serial Bus (USB). USBH0 and USBH1 are the companion host controller of HSUSBH0 and HSUSBH1 respectively.

Each USBH supports an integrated Root Hub with a USB port, a DMA for real-time data transfer between system memory and USB bus, port power control and port overcurrent detection.

The USBH is responsible for detecting the connect and disconnect of USB devices, managing data transfer, collecting status and activity of USB bus, providing power control and detecting overcurrent of attached USB devices.

6.32.2 Features

- Compliant with Universal Serial Bus (USB) Specification Revision 2.0.
- Supports Open Host Controller Interface (OpenHCI) Specification Revision 1.0.
- Supports full-speed (12Mbps) and low-speed (1.5Mbps) USB devices.
- Supports Control, Bulk, Interrupt, Isochronous and Split transfers.
- Supports an integrated Root Hub.
- Supports port power control and port overcurrent detection.
- Supports DMA for real-time data transfer.

6.32.3 Block Diagram

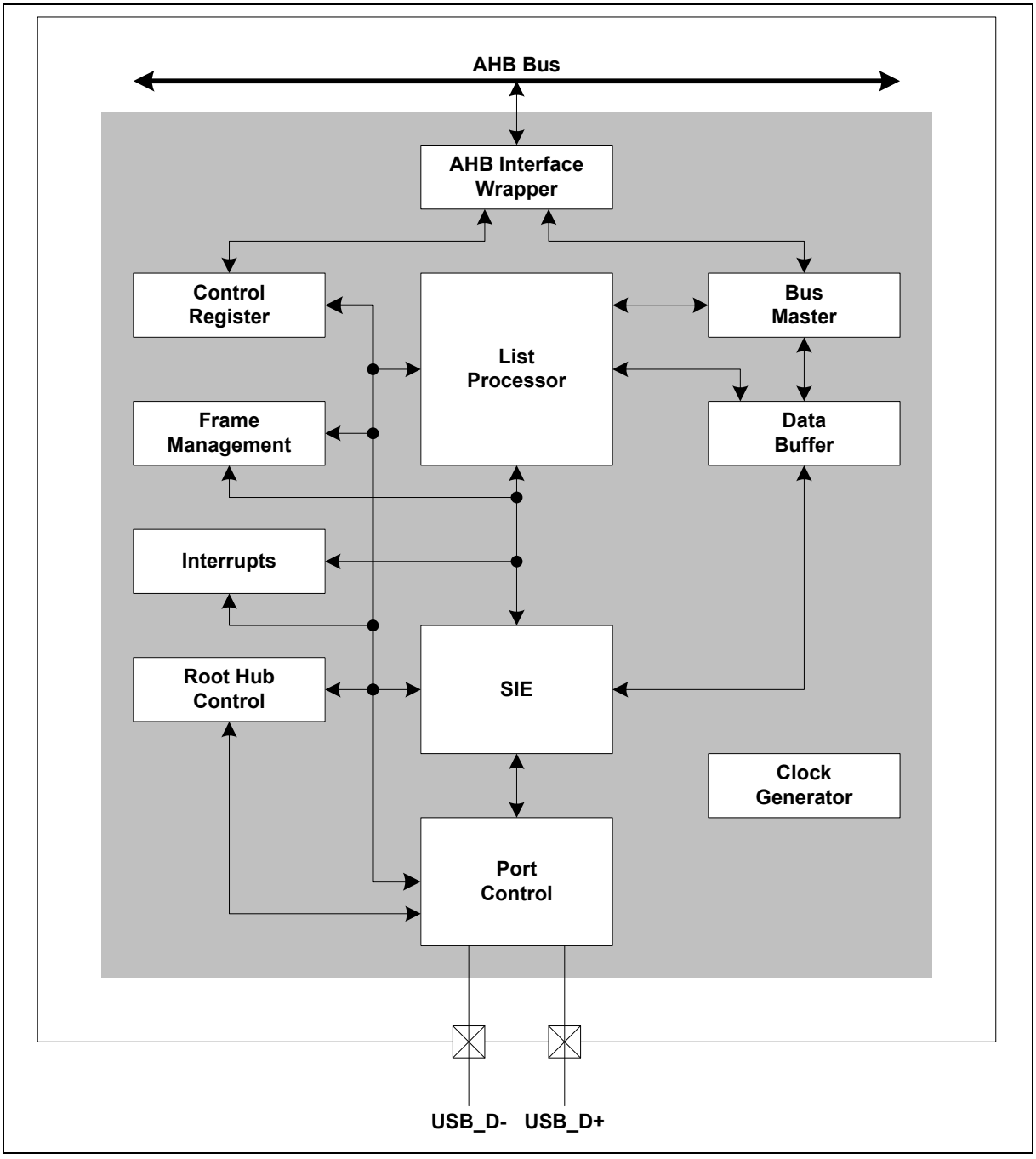


Figure 6.32-1 USB 1.1 Host Controller Block Diagram

6.32.4 Basic Configuration

6.32.4.1 USBH OHCI Controller Basic Configuration

- Clock Source Configuration
  - Enable USBH0 AHB clock in HUSBH0EN (CLK_SYSCLK0[21]).
  - Enable USBH1 AHB clock in HUSBH1EN (CLK_SYSCLK0[22]).
  - Enable USBH0 engine clock by the following setting:

        - Set PHY0SUSPEND (SYS_USBPMISCR[1]) to 1'b1.
        - Set PHY0POR (SYS_USBPMISCR[0]) to 1'b0.
        - Set PHY0COMN (SYS_USBPMISCR[2]) to 1'b0.
    - Enable USBH1 engine clock by the following setting:
        - Set PHY1SUSPEND (SYS_USBPMISCR[17]) to 1'b1.
        - Set PHY1POR (SYS_USBPMISCR[16]) to 1'b0.
        - Set PHY1COMN (SYS_USBPMISCR[18]) to 1'b0.
- Overcurrent Detection Level Configuration
    - Set USBH0/USBH1 overcurrent detection signal to high active in UHOVRCURH (SYS_MISCFCR0[12]).
- Reset Configuration
    - Reset USBH0 controller in HSUSBH0RST (SYS_IPRST0[20]).
    - Reset USBH1 controller in HSUSBH1RST (SYS_IPRST0[21]).

6.32.5 Functional Description

6.32.5.1 OHCI Controller

AHB Interface

The OpenHCI Host Controller is connected to the system by the AHB bus. The design requires both master and slave bus operations. As a master, the Host Controller is responsible for running cycles on the AHB bus to access EDs and TDs as well as transferring data between memory and the local data buffer. As a slave, the Host Controller monitors the cycles on the AHB bus and determines when to respond to these cycles. Configuration and non-real-time control access to the Host Controller operational registers are through the AHB bus slave interface.

Host Controller

The host controller includes 5 functional blocks, including List Processing, Frame Management, Interrupt Processing, Host Controller Bus Master and Data Buffer.

The List Processor manages the data structures from the Host Controller Driver and coordinates all activity within the Host Controller.

The Frame Management is responsible for managing the frame specific tasks required by the USB specification and the OpenHCI specification. These tasks are:

- Management of the OpenHCI frame specific Operational Registers
- Operation of the Largest Data Packet Counter.
- Performing frame qualifications on USB Transaction requests to the SIE.
- Generate SOF token requests to the SIE.

Interrupts are the communication method for HC-initiated communication with the Host Controller Driver. There are several events that may trigger an interrupt from the Host Controller. Each specific event sets a specific bit in the HcInterruptStatus register.

The Host Controller Bus Master is the central block in the data path. The Host Controller Bus Master coordinates all access to the AHB Interface. There are two sources of bus mastering within Host Controller: the List Processor and the Data Buffer Engine.

The Data Buffer serves as the data interface between the Bus Master and the SIE. It is a combination of a 64-byte latched based bi-directional asynchronous FIFO and a single DWORD AHB Holding

Register.

USB Interface

The USB interface includes the integrated Root Hub with an USB port, Port 1 as well as the Serial Interface Engine (SIE) and USB clock generator. The interface combines responsibility for executing bus transactions requested by the HC as well as the hub and port management specified by USB.

The SIE is responsible for managing all transactions to the USB. It controls the bus protocol, packet generation/extraction, data parallel-to-serial conversion, CRC coding, bit stuffing, and NRZI encoding. All transactions on the USB are requested from the List Processor and Frame Manager.

The Root Hub is a collection of ports that are individually controlled and a hub that maintains control/status over functions common to all ports.

6.32.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
USBH Base Address: **USBHn_BA = 0x4015_0000 + (0x8_0000 * n)** **n= 0, 1**				
HcRevision	USBHn_BA+0x000	R	Host Controller Revision Register	0x0000_0110
HcControl	USBHn_BA+0x004	R/W	Host Controller Control Register	0x0000_0000
HcCommandStatus	USBHn_BA+0x008	R/W	Host Controller Command Status Register	0x0000_0000
HcInterruptStatus	USBHn_BA+0x00C	R/W	Host Controller Interrupt Status Register	0x0000_0000
HcInterruptEnable	USBHn_BA+0x010	R/W	Host Controller Interrupt Enable Register	0x0000_0000
HcInterruptDisable	USBHn_BA+0x014	R/W	Host Controller Interrupt Disable Register	0x0000_0000
HcHCCA	USBHn_BA+0x018	R/W	Host Controller Communication Area Register	0x0000_0000
HcPeriodCurrentED	USBHn_BA+0x01C	R/W	Host Controller Period Current ED Register	0x0000_0000
HcControlHeadED	USBHn_BA+0x020	R/W	Host Controller Control Head ED Register	0x0000_0000
HcControlCurrentED	USBHn_BA+0x024	R/W	Host Controller Control Current ED Register	0x0000_0000
HcBulkHeadED	USBHn_BA+0x028	R/W	Host Controller Bulk Head ED Register	0x0000_0000
HcBulkCurrentED	USBHn_BA+0x02C	R/W	Host Controller Bulk Current ED Register	0x0000_0000
HcDoneHead	USBHn_BA+0x030	R/W	Host Controller Done Head Register	0x0000_0000
HcFmInterval	USBHn_BA+0x034	R/W	Host Controller Frame Interval Register	0x0000_2EDF
HcFmRemaining	USBHn_BA+0x038	R	Host Controller Frame Remaining Register	0x0000_0000
HcFmNumber	USBHn_BA+0x03C	R	Host Controller Frame Number Register	0x0000_0000
HcPeriodicStart	USBHn_BA+0x040	R/W	Host Controller Periodic Start Register	0x0000_0000
HcLSThreshold	USBHn_BA+0x044	R/W	Host Controller Low-speed Threshold Register	0x0000_0628
HcRhDescriptorA	USBHn_BA+0x048	R/W	Host Controller Root Hub Descriptor A Register	0x0100_0001
HcRhDescriptorB	USBHn_BA+0x04C	R/W	Host Controller Root Hub Descriptor B Register	0x0000_0000
HcRhStatus	USBHn_BA+0x050	R/W	Host Controller Root Hub Status Register	0x0000_0000
HcRhPortStatus	USBHn_BA+0x054	R/W	Host Controller Root Hub Port Status	0x0000_0000
HcPhyControl	USBHn_BA+0x200	R/W	Host Controller PHY Control Register	0x080F_0000

6.32.7 Register Description

Host Controller Revision Register (HcRevision)

Register	Offset	R/W	Description	Reset Value
HcRevision	USBHn_BA+0x000	R	Host Controller Revision Register	0x0000_0110

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
REV							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**REV**	**Revision Number** Indicates the Open HCI Specification revision number implemented by the Hardware. Host Controller supports 1.1 specification. (X.Y = XYh).

Host Controller Control Register (HcControl)

Register	Offset	R/W	Description	Reset Value
HcControl	USBHn_BA+0x004	R/W	Host Controller Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					RWE	RWC	IR
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
HCFS		BLE	CLE	IE	PLE	CBSR	

Bits	Description	
[31:11]	**Reserved**	Reserved.
[10]	**RWE**	**Remote Wakeup Enable** This bit is used by HCD to enable or disable the remote wakeup feature upon the detection of upstream resume signaling. When this bit is set and the ResumeDetected bit in HcInterruptStatus is set, a remote wakeup is signaled to the host system. Setting this bit has no impact on the generation of hardware interrupt.
[9]	**RWC**	**Remote Wakeup Connected** This bit indicates whether HC supports remote wakeup signaling. If remote wakeup is supported and used by the system it is the responsibility of system firmware to set this bit during POST. HC clears the bit upon a hardware reset but does not alter it upon a software reset. Remote wakeup signaling of the host system is host-bus-specific and is not described in this specification.
[8]	**IR**	**Interrupt Routing** This bit determines the routing of interrupts generated by events registered in HcInterruptStatus. If clear, all interrupts are routed to the normal host bus interrupt mechanism. If set, interrupts are routed to the System Management Interrupt. HCD clears this bit upon a hardware reset, but it does not alter this bit upon a software reset. HCD uses this bit as a tag to indicate the ownership of HC.
[7:6]	**HCFS**	**Host Controller Functional State** This field sets the Host Controller state. The Controller may force a state change from USBSUSPEND to USBRESUME after detecting resume signaling from a downstream port. States are: 00 = USBRESET. 01 = USBRESUME. 10 = USBOPERATIONAL. 11 = USBSUSPEND.
[5]	**BLE**	**Bulk List Enable Bit** 0 = Processing of the Bulk list after next SOF (Start-Of-Frame) Disabled. 1 = Processing of the Bulk list in the next frame Enabled.
[4]	**CLE**	**Control List Enable Bit** 0 = Processing of the Control list after next SOF (Start-Of-Frame) Disabled. 1 = Processing of the Control list in the next frame Enabled.

[3]	**IE**	**Isochronous List Enable Bit** Both ISOEn and PLE (HcControl[2]) high enables Host Controller to process the Isochronous list. Either ISOEn or PLE (HcControl[2]) is low disables Host Controller to process the Isochronous list. 0 = Processing of the Isochronous list after next SOF (Start-Of-Frame) Disabled. 1 = Processing of the Isochronous list in the next frame Enabled, if the PLE (HcControl[2]) is high, too.
[2]	**PLE**	**Periodic List Enable Bit** When set, this bit enables processing of the Periodic (interrupt and isochronous) list. The Host Controller checks this bit prior to attempting any periodic transfers in a frame. 0 = Processing of the Periodic (Interrupt and Isochronous) list after next SOF (Start-Of-Frame) Disabled. 1 = Processing of the Periodic (Interrupt and Isochronous) list in the next frame Enabled. **Note:** To enable the processing of the Isochronous list, user has to set both PLE and IE (HcControl[3]) high.
[1:0]	**CBSR**	**Control Bulk Service Ratio** This specifies the service ratio between Control and Bulk EDs. Before processing any of the non-periodic lists, HC must compare the ratio specified with its internal count on how many nonempty Control EDs have been processed, in determining whether to continue serving another Control ED or switching to Bulk EDs. The internal count will be retained when crossing the frame boundary. In case of reset, HCD is responsible for restoring this value. 00 = Number of Control EDs over Bulk EDs served is 1:1. 01 = Number of Control EDs over Bulk EDs served is 2:1. 10 = Number of Control EDs over Bulk EDs served is 3:1. 11 = Number of Control EDs over Bulk EDs served is 4:1.

Host Controller Command Status Register (HcCommandStatus)

Register	Offset	R/W	Description	Reset Value
HcCommandStatus	USBHn_BA+0x008	R/W	Host Controller Command Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						SOC	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					BLF	CLF	HCR

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17:16]	**SOC**	**Schedule Overrun Count (Read Only)** These bits are incremented on each scheduling overrun error. It is initialized to 00b and wraps around at 11b. This will be incremented when a scheduling overrun is detected even if SO (HcInterruptStatus[0]) has already been set.
[15:3]	**Reserved**	Reserved.
[2]	**BLF**	**Bulk List Filled** Set high to indicate there is an active TD on the Bulk list. This bit may be set by either software or the Host Controller and cleared by the Host Controller each time it begins processing the head of the Bulk list. 0 = No active TD found or Host Controller begins to process the head of the Bulk list. 1 = An active TD added or found on the Bulk list.
[1]	**CLF**	**Control List Filled** Set high to indicate there is an active TD on the Control List. It may be set by either software or the Host Controller and cleared by the Host Controller each time it begins processing the head of the Control List. 0 = No active TD found or Host Controller begins to process the head of the Control list. 1 = An active TD added or found on the Control list.
[0]	**HCR**	**Host Controller Reset** This bit is set to initiate the software reset of Host Controller. This bit is cleared by the Host Controller, upon completed of the reset operation. This bit, when set, didn't reset the Root Hub and no subsequent reset signaling be asserted to its downstream ports. 0 = Host Controller is not in software reset state. 1 = Host Controller is in software reset state.

Host Controller Interrupt Status Register (HcInterruptStatus)

Register	Offset	R/W	Description	Reset Value
HcInterruptStatus	USBHn_BA+0x00C	R/W	Host Controller Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	RHSC	FNO	Reserved	RD	SF	WDH	SO

Bits	Description	
[31:7]	**Reserved**	Reserved.
[6]	**RHSC**	**Root Hub Status Change** This bit is set when the content of HcRhStatus or the content of HcRhPortStatus register has changed. 0 = The content of HcRhStatus and the content of HcRhPortStatus register didn't change. 1 = The content of HcRhStatus or the content of HcRhPortStatus register has changed. **Note:** This bit is cleared by writing h'1f to HcRhPortStatus6[20:16].
[5]	**FNO**	**Frame Number Overflow** This bit is set when bit 15 of Frame Number changes from 1 to 0 or from 0 to 1. 0 = The bit 15 of Frame Number didn't change. 1 = The bit 15 of Frame Number changes from 1 to 0 or from 0 to 1. **Note:** This bit is cleared by writing 1 to it.
[4]	**Reserved**	Reserved.
[3]	**RD**	**Resume Detected** Set when Host Controller detects resume signaling on a downstream port. 0 = No resume signaling detected on a downstream port. 1 = Resume signaling detected on a downstream port. **Note:** This bit is cleared by writing 1 to it.
[2]	**SF**	**Start of Frame** Set when the Frame Management functional block signals a 'Start of Frame' event. Host Control generates a SOF token at the same time. 0 = Not the start of a frame. 1 = Indicate the start of a frame and Host Controller generates a SOF token. **Note:** This bit is cleared by writing 1 to it.

[1]	**WDH**	**Write Back Done Head** Set after the Host Controller has written HcDoneHead to HccaDoneHead. Further updates of the HccaDoneHead will not occur until this bit has been cleared. 0 = Host Controller didn't update HccaDoneHead. 1 = Host Controller has written HcDoneHead to HccaDoneHead. **Note:** This bit is cleared by writing 1 to it.
[0]	**SO**	**Scheduling Overrun** Set when the List Processor determines a Schedule Overrun has occurred. 0 = Schedule Overrun didn't occur. 1 = Schedule Overrun has occurred. **Note:** This bit is cleared by writing 1 to it.

Host Controller Interrupt Enable Register (HcInterruptEnable)

Register	Offset	R/W	Description	Reset Value
HcInterruptEnable	USBHn_BA+0x010	R/W	Host Controller Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
MIE	Reserved						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	RHSC	FNO	Reserved	RD	SF	WDH	SO

Bits	Description	
[31]	**MIE**	**Master Interrupt Enable Bit** This bit is a global interrupt enable. A write of '1' allows interrupts to be enabled via the specific enable bits listed above. Interrupt generation due to RHSC (HcInterruptStatus[6]), FNO (HcInterruptStatus[5]), RD (HcInterruptStatus[3]), SF (HcInterruptStatus[2]), WDH (HcInterruptStatus[1]) or SO (HcInterruptStatus[0]) are enabled only if the corresponding bit in HcInterruptEnable is 1 and MIE is 1. Write Operation: 0 = No effect. 1 = Enable global interrupt. Read Operation: 0 = Global interrupt disabled. 1 = Global interrupt enabled.
[30:7]	**Reserved**	Reserved.
[6]	**RHSC**	**Root Hub Status Change Enable Bit** Write Operation: 0 = No effect. 1 = Enable Interrupt generation due to RHSC (HcInterruptStatus[6]). Read Operation: 0 = Interrupt generation due to RHSC (HcInterruptStatus[6]) Disabled. 1 = Interrupt generation due to RHSC (HcInterruptStatus[6]) Enabled.
[5]	**FNO**	**Frame Number Overflow Enable Bit** Write Operation: 0 = No effect. 1 = EnableInterrupt generation due to FNO (HcInterruptStatus[5]). Read Operation: 0 = Interrupt generation due to FNO (HcInterruptStatus[5]) Disabled. 1 = Interrupt generation due to FNO (HcInterruptStatus[5]) Enabled.
[4]	**Reserved**	Reserved.

[3]	**RD**	**Resume Detected Enable Bit** Write Operation: 0 = No effect. 1 = Enable Interrupt generation due to RD (HcInterruptStatus[3]). Read Operation: 0 = Interrupt generation due to RD (HcInterruptStatus[3]) Disabled. 1 = Interrupt generation due to RD (HcInterruptStatus[3]) Enabled.
[2]	**SF**	**Start of Frame Enable Bit** Write Operation: 0 = No effect. 1 = Enable Interrupt generation due to SF (HcInterruptStatus[2]). Read Operation: 0 = Interrupt generation due to SF (HcInterruptStatus[2]) Disabled. 1 = Interrupt generation due to SF (HcInterruptStatus[2]) Enabled.
[1]	**WDH**	**Write Back Done Head Enable Bit** Write Operation: 0 = No effect. 1 = Enable Interrupt generation due to WDH (HcInterruptStatus[1]). Read Operation: 0 = Interrupt generation due to WDH (HcInterruptStatus[1]) Disabled. 1 = Interrupt generation due to WDH (HcInterruptStatus[1]) Enabled.
[0]	**SO**	**Scheduling Overrun Enable Bit** Write Operation: 0 = No effect. 1 = Enable Interrupt generation due to SO (HcInterruptStatus[0]). Read Operation: 0 = Interrupt generation due to SO (HcInterruptStatus[0]) Disabled. 1 = Interrupt generation due to SO (HcInterruptStatus[0]) Enabled.

Host Controller Interrupt Disable Register (HcInterruptDisable)

Register	Offset	R/W	Description	Reset Value
HcInterruptDisable	USBHn_BA+0x014	R/W	Host Controller Interrupt Disable Register	0x0000_0000

31	30	29	28	27	26	25	24
MIE	Reserved						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	RHSC	FNO	Reserved	RD	SF	WDH	SO

Bits	Description	
[31]	**MIE**	**Master Interrupt Disable Bit** Global interrupt disable. Writing ‘1’ to disable all interrupts. Write Operation: 0 = No effect. 1 = Disable global interrupt. Read Operation: 0 = Global interrupt disabled. 1 = Global interrupt enabled.
[30:7]	**Reserved**	Reserved.
[6]	**RHSC**	**Root Hub Status Change Disable Bit** Write Operation: 0 = No effect. 1 = Disable Interrupt generation due to RHSC (HcInterruptStatus[6]). Read Operation: 0 = Interrupt generation due to RHSC (HcInterruptStatus[6]) Disabled. 1 = Interrupt generation due to RHSC (HcInterruptStatus[6]) Enabled.
[5]	**FNO**	**Frame Number Overflow Disable Bit** Write Operation: 0 = No effect. 1 = Disable Interrupt generation due to FNO (HcInterruptStatus[5]). Read Operation: 0 = Interrupt generation due to FNO (HcInterruptStatus[5]) Disabled. 1 = Interrupt generation due to FNO (HcInterruptStatus[5]) Enabled.
[4]	**Reserved**	Reserved.

[3]	**RD**	**Resume Detected Disable Bit** Write Operation: 0 = No effect. 1 = Disable Interrupt generation due to RD (HcInterruptStatus[3]). Read Operation: 0 = Interrupt generation due to RD (HcInterruptStatus[3]) Disabled. 1 = Interrupt generation due to RD (HcInterruptStatus[3]) Enabled.
[2]	**SF**	**Start of Frame Disable Bit** Write Operation: 0 = No effect. 1 = Disable Interrupt generation due to SF (HcInterruptStatus[2]). Read Operation: 0 = Interrupt generation due to SF (HcInterruptStatus[2]) Disabled. 1 = Interrupt generation due to SF (HcInterruptStatus[2]) Enabled.
[1]	**WDH**	**Write Back Done Head Disable Bit** Write Operation: 0 = No effect. 1 = Disable Interrupt generation due to WDH (HcInterruptStatus[1]). Read Operation: 0 = Interrupt generation due to WDH (HcInterruptStatus[1]) Disabled. 1 = Interrupt generation due to WDH (HcInterruptStatus[1]) Enabled.
[0]	**SO**	**Scheduling Overrun Disable Bit** Write Operation: 0 = No effect. 1 = Disable Interrupt generation due to SO (HcInterruptStatus[0]). Read Operation: 0 = Interrupt generation due to SO (HcInterruptStatus[0]) Disabled. 1 = Interrupt generation due to SO (HcInterruptStatus[0]) Enabled.

Host Controller Communication Area Register (HcHCCA)

Register	Offset	R/W	Description	Reset Value
HcHCCA	USBHn_BA+0x018	R/W	Host Controller Communication Area Register	0x0000_0000

31	30	29	28	27	26	25	24
HCCA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
HCCA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
HCCA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:8]	**HCCA**	**Host Controller Communication Area** Pointer to indicate base address of the Host Controller Communication Area (HCCA).
[7:0]	**Reserved**	Reserved.

Host Controller Period Current ED Register (HcPeriodCurrentED)

Register	Offset	R/W	Description	Reset Value
HcPeriodCurrentED	USBHn_BA+0x01C	R/W	Host Controller Period Current ED Register	0x0000_0000

31	30	29	28	27	26	25	24
PCED							
23	22	21	20	19	18	17	16
PCED							
15	14	13	12	11	10	9	8
PCED							
7	6	5	4	3	2	1	0
PCED				Reserved			

Bits	Description	
[31:4]	**PCED**	**Periodic Current ED** Pointer to indicate physical address of the current Isochronous or Interrupt Endpoint Descriptor.
[3:0]	**Reserved**	Reserved.

Host Controller Control Head ED Register (HcControlHeadED)

Register	Offset	R/W	Description	Reset Value
HcControlHeadED	USBHn_BA+0x020	R/W	Host Controller Control Head ED Register	0x0000_0000

31	30	29	28	27	26	25	24
CHED							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CHED							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CHED							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CHED				Reserved			

Bits	Description	
[31:4]	**CHED**	**Control Head ED** Pointer to indicate physical address of the first Endpoint Descriptor of the Control list.
[3:0]	**Reserved**	Reserved.

Host Controller Control Current ED Register (HcControlCurrentED)

Register	Offset	R/W	Description	Reset Value
HcControlCurrentED	USBHn_BA+0x024	R/W	Host Controller Control Current ED Register	0x0000_0000

31	30	29	28	27	26	25	24
CCED							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CCED							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CCED							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CCED				Reserved			

Bits	Description	
[31:4]	**CCED**	**Control Current Head ED** Pointer to indicate the physical address of the current Endpoint Descriptor of the Control list.
[3:0]	**Reserved**	Reserved.

Host Controller Bulk Head ED Register (HcBulkHeadED)

Register	Offset	R/W	Description	Reset Value
HcBulkHeadED	USBHn_BA+0x028	R/W	Host Controller Bulk Head ED Register	0x0000_0000

31	30	29	28	27	26	25	24
BHED							
23	22	21	20	19	18	17	16
BHED							
15	14	13	12	11	10	9	8
BHED							
7	6	5	4	3	2	1	0
BHED				Reserved			

Bits	Description	
[31:4]	**BHED**	**Bulk Head ED** Pointer to indicate the physical address of the first Endpoint Descriptor of the Bulk list.
[3:0]	**Reserved**	Reserved.

Host Controller Bulk Current ED Register (HcBulkCurrentED)

Register	Offset	R/W	Description	Reset Value
HcBulkCurrentED	USBHn_BA+0x02C	R/W	Host Controller Bulk Current ED Register	0x0000_0000

31	30	29	28	27	26	25	24
BCED							
23	22	21	20	19	18	17	16
BCED							
15	14	13	12	11	10	9	8
BCED							
7	6	5	4	3	2	1	0
BCED				Reserved			

Bits	Description	
[31:4]	**BCED**	**Bulk Current Head ED** Pointer to indicate the physical address of the current endpoint of the Bulk list.
[3:0]	**Reserved**	Reserved.

Host Controller Done Head Register (HcDoneHead)

Register	Offset	R/W	Description	Reset Value
HcDoneHead	USBHn_BA+0x030	R/W	Host Controller Done Head Register	0x0000_0000

31	30	29	28	27	26	25	24
DH							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DH							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DH							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DH				Reserved			

Bits	Description	
[31:4]	**DH**	**Done Head** Pointer to indicate the physical address of the last completed Transfer Descriptor that was added to the Done queue.
[3:0]	**Reserved**	Reserved.

Host Controller Frame Interval Register (HcFmInterval)

Register	Offset	R/W	Description	Reset Value
HcFmInterval	USBHn_BA+0x034	R/W	Host Controller Frame Interval Register	0x0000_2EDF

31	30	29	28	27	26	25	24
FIT	Reserved	FSMPS					
23	22	21	20	19	18	17	16
FSMPS							
15	14	13	12	11	10	9	8
Reserved		FI					
7	6	5	4	3	2	1	0
FI							

Bits	Description	
[31]	**FIT**	**Frame Interval Toggle** This bit is toggled by Host Controller Driver when it loads a new value into FI (HcFmInterval[13:0]). 0 = Host Controller Driver didn't load new value into FI (HcFmInterval[13:0]). 1 = Host Controller Driver loads a new value into FI (HcFmInterval[13:0]).
[30]	**Reserved**	Reserved.
[29:16]	**FSMPS**	**FS Largest Data Packet** This field specifies a value that is loaded into the Largest Data Packet Counter at the beginning of each frame.
[15:14]	**Reserved**	Reserved.
[13:0]	**FI**	**Frame Interval** This field specifies the length of a frame as (bit times - 1). For 12,000 bit times in a frame, a value of 11,999 is stored here.

Host Controller Frame Remaining Register (HcFmRemaining)

Register	Offset	R/W	Description	Reset Value
HcFmRemaining	USBHn_BA+0x038	R	Host Controller Frame Remaining Register	0x0000_0000

31	30	29	28	27	26	25	24
FRT	Reserved						
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		FR					
7	6	5	4	3	2	1	0
FR							

Bits	Description	
[31]	**FRT**	**Frame Remaining Toggle** This bit is loaded from the FIT (HcFmInterval[31]) whenever FR (HcFmRemaining[13:0]) reaches 0.
[30:14]	**Reserved**	Reserved.
[13:0]	**FR**	**Frame Remaining** When the Host Controller is in the USBOPERATIONAL state, this 14-bit field decrements each 12 MHz clock period. When the count reaches 0, (end of frame) the counter reloads with Frame Interval. In addition, the counter loads when the Host Controller transitions into USBOPERATIONAL.

Host Controller Frame Number Register (HcFmNumber)

Register	Offset	R/W	Description	Reset Value
HcFmNumber	USBHn_BA+0x03C	R	Host Controller Frame Number Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
FN							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FN							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**FN**	**Frame Number** This 16-bit incrementing counter field is incremented coincident with the re-load of FR (HcFmRemaining[13:0]). The count rolls over from 'FFFFh' to '0h.'

Host Controller Periodic Start Register (HcPeriodicStart)

Register	Offset	R/W	Description	Reset Value
HcPeriodicStart	USBHn_BA+0x040	R/W	Host Controller Periodic Start Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		PS					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PS							

Bits	Description	
[31:14]	**Reserved**	Reserved.
[13:0]	**PS**	**Periodic Start** This field contains a value used by the List Processor to determine where in a frame the Periodic List processing must begin.

Host Controller Low-speed Threshold Register (HcLSThreshold)

Register	Offset	R/W	Description	Reset Value
HcLSThreshold	USBHn_BA+0x044	R/W	Host Controller Low-speed Threshold Register	0x0000_0628

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				LST			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
LST							

Bits	Description	
[31:12]	**Reserved**	Reserved.
[11:0]	**LST**	**Low-speed Threshold** This field contains a value which is compared to the FR (HcFmRemaining[13:0]) field prior to initiating a Low-speed transaction. The transaction is started only if FR (HcFmRemaining[13:0]) >= this field. The value is calculated by Host Controller Driver with the consideration of transmission and setup overhead.

Host Controller Root Hub Descriptor A Register (HcRhDescriptorA)

Register	Offset	R/W	Description	Reset Value
HcRhDescriptorA	USBHn_BA+0x048	R/W	Host Controller Root Hub Descriptor A Register	0x0100_0001

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			NOCP	OCPM	Reserved		PSM
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NDP							

Bits	Description	
[31:13]	**Reserved**	Reserved.
[12]	**NOCP**	**No overcurrent Protection** This bit describes how the overcurrent status for the Root Hub ports reported. 0 = Overcurrent status is reported. 1 = Overcurrent status is not reported.
[11]	**OCPM**	**Overcurrent Protection Mode** This bit describes how the overcurrent status for the Root Hub ports reported. This bit is only valid when NOCP (HcRhDescriptorA[12]) is cleared. 0 = Global overcurrent. 1 = Individual overcurrent.
[10:9]	**Reserved**	Reserved.
[8]	**PSM**	**Power Switching Mode** This bit is used to specify how the power switching of the Root Hub ports is controlled. 0 = Global switching. 1 = Individual switching.
[7:0]	**NDP**	**Number Downstream Ports** USB host control supports two downstream ports and only one port is available in this series of chip. **Note:** NDP = 6 in this series of chip.

Host Controller Root Hub Descriptor B Register (HcRhDescriptorB)

Register	Offset	R/W	Description	Reset Value
HcRhDescriptorB	USBHn_BA+0x04C	R/W	Host Controller Root Hub Descriptor B Register	0x0000_0000

31	30	29	28	27	26	25	24
PPCM							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PPCM							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:16]	**PPCM**	**Port Power Control Mask** Global power switching. This field is only valid if PowerSwitchingMode is set (individual port switching). When set, the port only responds to individual port power switching commands (Set/Clear Port Power). When cleared, the port only responds to global power switching commands (Set/Clear Global Power). 0 = Port power controlled by global power switching. 1 = Port power controlled by port power switching. **Note:** PPCM[15:2] and PPCM[0] are reserved.
[15:0]	**Reserved**	Reserved.

Host Controller Root Hub Status Register (HcRhStatus)

Register	Offset	R/W	Description	Reset Value
HcRhStatus	USBHn_BA+0x050	R/W	Host Controller Root Hub Status Register	0x0000_0000

31	30	29	28	27	26	25	24
CRWE	Reserved						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						OCIC	LPSC
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DRWE	Reserved						
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						OCI	LPS

Bits	Description	
[31]	**CRWE**	**Clear Remote Wake-up Enable Bit** This bit is use to clear DRWE (HcRhStatus[15]). This bit always read as zero. Write Operation: 0 = No effect. 1 = Clear DRWE (HcRhStatus[15]).
[30:18]	**Reserved**	Reserved.
[17]	**OCIC**	**Overcurrent Indicator Change** This bit is set by hardware when a change has occurred in OCI (HcRhStatus[1]). Write 1 to clear this bit to zero. 0 = OCI (HcRhStatus[1]) didn't change. 1 = OCI (HcRhStatus[1]) change.
[16]	**LPSC**	**Set Global Power** In global power mode (PSM (HcRhDescriptorA[8]) = 0), this bit is written to one to enable power to all ports. This bit always read as zero. Write Operation: 0 = No effect. 1 = Set global power.
[15]	**DRWE**	**Device Remote Wakeup Enable Bit** This bit controls if port's Connect Status Change as a remote wake-up event. Write Operation: 0 = No effect. 1 = Connect Status Change as a remote wake-up event Enabled. Read Operation: 0 = Connect Status Change as a remote wake-up event Disabled. 1 = Connect Status Change as a remote wake-up event Enabled.
[14:2]	**Reserved**	Reserved.

[1]	**OCI**	**Overcurrent Indicator (Read Only)** This bit reflects the state of the overcurrent status pin. This field is only valid if NOCP (HcRhDescriptorA[12]) and OCPM (HcRhDescriptorA[11]) are cleared. 0 = No overcurrent condition. 1 = Overcurrent condition.
[0]	**LPS**	**Clear Global Power** In global power mode (PSM (HcRhDescriptorA[8]) = 0), this bit is written to one to clear all ports' power. This bit always read as zero. Write Operation: 0 = No effect. 1 = Clear global power.

Host Controller Root Hub Port Status (HcRhPortStatus)

Register	Offset	R/W	Description	Reset Value
HcRhPortStatus	USBHn_BA+0x054	R/W	Host Controller Root Hub Port Status	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			PRSC	OCIC	PSSC	PESC	CSC
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						LSDA	PPS
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			PRS	POCI	PSS	PES	CCS

Bits	Description	
[31:21]	**Reserved**	Reserved.
[20]	**PRSC**	**Port Reset Status Change** This bit indicates that the port reset signal has completed. Write 1 to clear this bit to zero. 0 = Port reset is not complete. 1 = Port reset is complete.
[19]	**OCIC**	**Port overcurrent Indicator Change** This bit is set when POCI (HcRhPortStatus[3]) changes. Write 1 to clear this bit to zero. 0 = POCI (HcRhPortStatus[3]) didn't change. 1 = POCI (HcRhPortStatus[3]) changes.
[18]	**PSSC**	**Port Suspend Status Change** This bit indicates the completion of the selective resume sequence for the port. Write 1 to clear this bit to zero. 0 = Port resume is not completed. 1 = Port resume completed.
[17]	**PESC**	**Port Enable Status Change** This bit indicates that the port has been disabled (PES (HcRhPortStatus6[1]) cleared) due to a hardware event. Write 1 to clear this bit to zero. 0 = PES (HcRhPortStatus[1]) didn't change. 1 = PES (HcRhPortStatus[1]) changed.
[16]	**CSC**	**Connect Status Change** This bit indicates connect or disconnect event has been detected (CCS (HcRhPortStatus[0]) changed). Write 1 to clear this bit to zero. 0 = No connect/disconnect event (CCS (HcRhPortStatus[0]) didn't change). 1 = Hardware detection of connect/disconnect event (CCS (HcRhPortStatus[0]) changed).

[15:10]	**Reserved**	Reserved.
[9]	**LSDA**	**Low Speed Device Attached or Clear Port Power** This bit defines the speed (and bus idle) of the attached device. It is only valid when CCS (HcRhPortStatus[0]) is set. This bit is also used to clear port power. Write Operation: 0 = No effect. 1 = Clear PPS (HcRhPortStatus[8]). Read Operation: 0 = Full Speed device. 1 = Low-speed device.
[8]	**PPS**	**Port Power Status or Set Port Power** This bit reflects the power state of the port regardless of the power switching mode. Write Operation: 0 = No effect. 1 = Port Power Enabled. Read Operation: 0 = Port power is Disabled. 1 = Port power is Enabled.
[7:5]	**Reserved**	Reserved.
[4]	**PRS**	**Port Reset Status or Set Port Reset** This bit reflects the reset state of the port. Write Operation: 0 = No effect. 1 = Set port reset. Read Operation 0 = Port reset signal is not active. 1 = Port reset signal is active.
[3]	**POCI**	**Port overcurrent Indicator or Clear Port Suspend** This bit reflects the state of the overcurrent status pin dedicated to this port. This field is only valid if NOCP (HcRhDescriptorA[12]) is cleared and OCPM (HcRhDescriptorA[11]) is set. This bit is also used to initiate the selective result sequence for the port. Write Operation: 0 = No effect. 1 = Clear port suspend. Read Operation: 0 = No overcurrent condition. 1 = Overcurrent condition.
[2]	**PSS**	**Port Suspend Status or Set Port Suspend** This bit indicates the port is suspended Write Operation: 0 = No effect. 1 = Set port suspend. Read Operation: 0 = Port is not suspended. 1 = Port is selectively suspended.

[1]	**PES**	**Port Enable Status or Set Port Enable** Write Operation: 0 = No effect. 1 = Set port enable. Read Operation: 0 = Port Disabled. 1 = Port Enabled.
[0]	**CCS**	**Current Connect Status or Clear Port Enable** Write Operation: 0 = No effect. 1 = Clear port enable. Read Operation: 0 = No device connected. 1 = Device connected.

Host Controller PHY Control Register (HcPhyControl)

Register	Offset	R/W	Description	Reset Value
HcPhyControl	USBHn_BA+0x200	R/W	Host Controller PHY Control Register	0x080F_0000

31	30	29	28	27	26	25	24
Reserved				STBYEN	Reserved		
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27]	**STBYEN**	**USB Transceiver Standby Enable Bit** This bit controls if USB transceiver could enter the standby mode to reduce power consumption. 0 = The USB transceiver would never enter the standby mode. 1 = The USB transceiver will enter standby mode while port is in power off state (port power is inactive).
[26:0]	**Reserved**	Reserved.

6.33 Controller Area Network with Feasibility Data Rate (CAN FD)

6.33.1 Overview

The CAN FD controller performs communication according to ISO 11898-1:2015 and need be connected to additional transceiver hardware for the physical layer.

The CAN FD controller consists of one CAN Core, Memory access control and arbiter, Tx Handler, Rx Handler, a shared Message RAM memory and a 32-bit AHB interface for control and configuration registers.

The message storage is intended to be a single-ported Message RAM outside of the CAN Core module. It is connected to the CAN Core via the memory control interface. The Message RAM implements filters, receive FIFOs, transmit event FIFOs and transmit FIFOs.

All functions concerning the handling of messages are implemented by the Rx Handler and the Tx Handler. The Rx Handler manages message acceptance filtering, the transfer of received messages from the CAN Core to the Message RAM as well as providing received message status information. The Tx Handler is responsible for the transfer of transmit messages from the Message RAM to the CAN Core as well as providing transmitted status information.

The controller's clock domain concept allows the separation among CAN Core clock and the AHB clock.

6.33.2 Features

- Conform with CAN protocol version 2.0 part A, B and ISO 11898-1: 2015
- CAN FD with up to 64 data bytes supported
- CAN Error Logging
- AUTOSAR support
- SAE J1939 support
- Improved acceptance filtering
- Two configurable Receive FIFOs
- Separate signaling on reception of High Priority Messages
- Configurable Transmit FIFO, Transmit Queue, Transmit Event FIFO
- Direct Message RAM access for CPU
- Programmable loop-back test mode
- Maskable module interrupts
- Two clock domains (CAN Core clock and AHB clock)
- Power-down support

6.33.3 Block Diagram

The controller interfaces with the AMBA AHB bus. Figure 6.33-1 shows the block diagram of the controller.

CAN Core

CAN Protocol Controller and Rx/Tx shift register. Handle all 2.0 part A, B and ISO 11898-1:2015 protocol functions. Supports 11-bit and 29-bit identifiers

Sync

Synchronizes signals from the AHB clock domain to the CAN Core clock domain and vice versa

Control and Configuration Registers

All registers and related configuration used to control and to configure the CAN Core.

Interrupt and power down control

The CAN FD controller provides two interrupt lines CANFDx_INT0 and CANFDx_INT1. By programming EINT0 and EINT1 bits in CANFD_ILE register, the interrupt lines can be enabled or disabled separately.

The CAN FD controller can be set into power down controlled by both CANnSTREQ (SYS_MISCRFCR1[3:0]), and CANnSTACK (SYS_MISCRFCR1[3:0]) clear signal.

Memory access control and Arbiter

The arbiter is responsible for the Message RAM access authority between the CPU and the interface of Tx/Rx Handler to an external 32-bit Message RAM.

Message RAM (SRAM)

Stores message objects and Identifier masks. The maximum Message RAM size is 2K words.

Tx/Rx Message Handler

State Machine that controls the data transfer between the Rx/Tx Shift Register of the CAN Core and the Message RAM as well as the generation of interrupts as programmed in the Control and Configuration registers.

AHB Interface

Connect the CAN FD controller to the AHB bus.

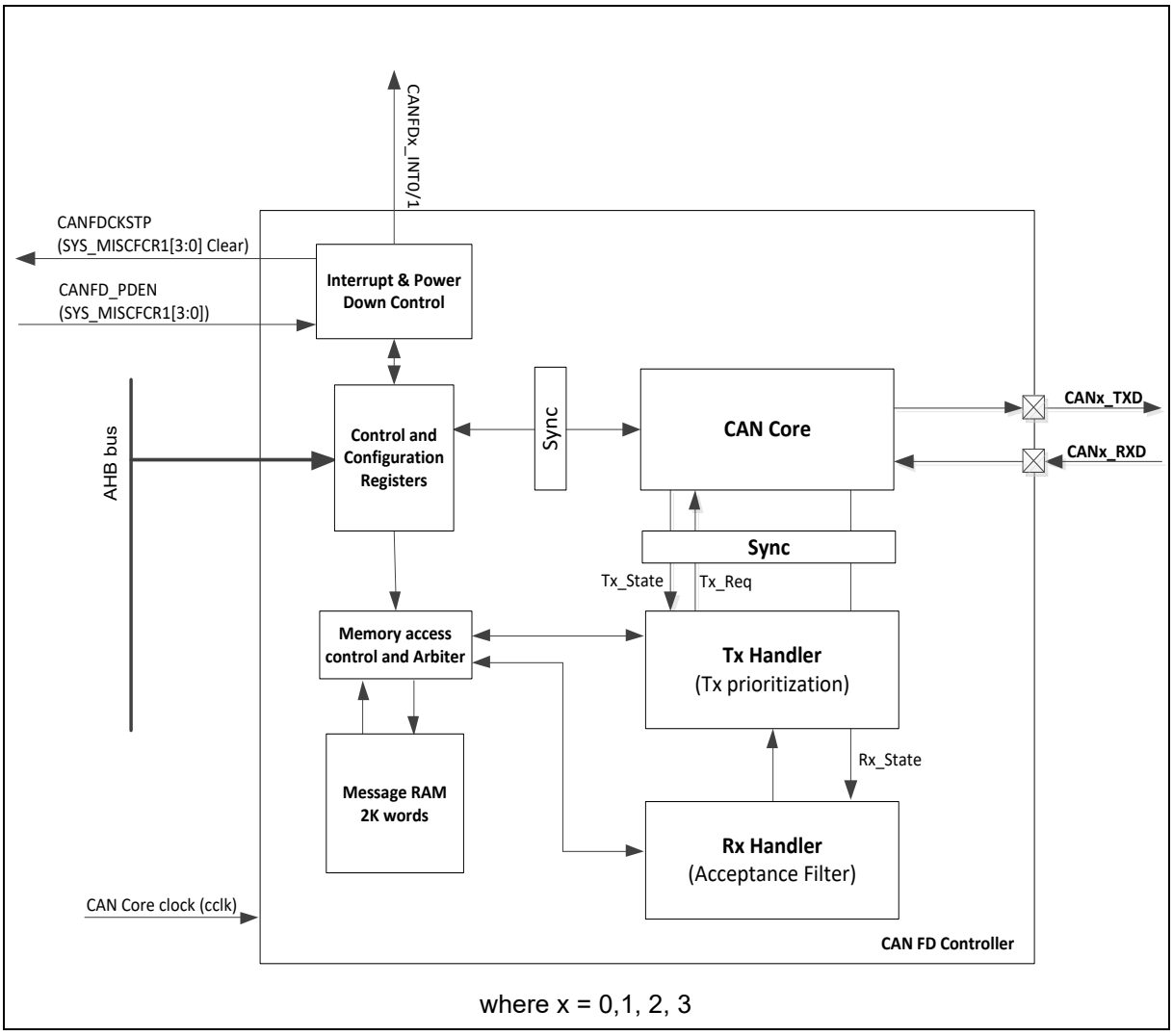


Figure 6.33-1 CAN FD Controller Block Diagram

6.33.4 Basic Configuration

6.33.4.1 CAN FD 0 Basic Configuration

- Clock Source Configuration
  - Select the clock source of CAN FD 0 controller in CANFD0SEL (CLK_CLKSEL4[16]).
  - Select the clock divider number of CAN FD 0 controller in CANFD0DIV (CLK_CLKDIV0[2:0]).
  - Enable the clock of CAN FD 0 controller in CAN0CKEN (CLK_SYSCLK0[8]).
- Reset Configuration
  - Reset CAN FD 0 controller in CANFD0RST (SYS_IPRST1[24]).
- Pin Configuration

Group	Pin Name	GPIO	MFP

CAN FD 0	CAN0_RXD	PB.10	MFP3
		PC.2	MFP3
		PG.2	MFP3
		PI.2	MFP3
		PJ.10	MFP3
		PK.6	MFP3
		PL.6	MFP3
		PM.6	MFP3
		PN.2	MFP3
	CAN0_TXD	PB.11	MFP3
		PC.3	MFP3
		PG.3	MFP3
		PI.3	MFP3
		PJ.11	MFP3
		PK.7	MFP3
		PL.7	MFP3
		PM.7	MFP3
		PN.3	MFP3

6.33.4.2 CAN FD 1 Basic Configuration

- Clock Source Configuration
  - Select the clock source of CAN FD 1 controller in CANFD1SEL (CLK_CLKSEL4[17]).
  - Select the clock divider number of CAN FD 1 controller in CANFD1DIV (CLK_CLKDIV0[6:4]).
  - Enable the clock of CAN FD 1 controller in CAN1CKEN (CLK_SYSCLK0[9]).
- Reset Configuration
  - Reset CAN FD 1 controller in CANFD1RST (SYS_IPRST1[25]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
CAN FD 1	CAN1_RXD	PB.14	MFP3
		PC.6	MFP3
		PG.6	MFP3
		PI.6	MFP3
		PJ.2	MFP3
		PK.10	MFP3
		PL.14	MFP4

		PN.6	MFP3
	CAN1_TXD	PB.15	MFP3
		PC.7	MFP3
		PG.7	MFP3
		PI.7	MFP3
		PJ.3	MFP3
		PK.11	MFP3
		PL.15	MFP4
		PN.7	MFP3

6.33.4.3 CAN FD 2 Basic Configuration

- Clock Source Configuration
  - Select the clock source of CAN FD 2 controller in CANFD2SEL (CLK_CLKSEL4[18]).
  - Select the clock divider number of CAN FD 2 controller in CANFD2DIV (CLK_CLKDIV0[10:8]).
  - Enable the clock of CAN FD 2 controller in CAN2CKEN (CLK_SYSCLK0[10]).
- Reset Configuration
  - Reset CAN FD 2 controller in CANFD2RST (SYS_IPRST2[28]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
CAN FD 2	CAN2_RXD	PA.15	MFP5
		PB.12	MFP5
		PF.4	MFP13
		PK.2	MFP3
		PM.10	MFP3
		PM.14	MFP4
		PN.10	MFP3
	CAN2_TXD	PB.13	MFP5
		PF.5	MFP13
		PG.1	MFP5
		PK.3	MFP3
		PM.11	MFP3
		PM.15	MFP4
		PN.11	MFP3

6.33.4.4 CAN FD 3 Basic Configuration

- Clock Source Configuration
  - Select the clock source of CAN FD 3 controller in CANFD3SEL (CLK_CLKSEL4[19]).
  - Select the clock divider number of CAN FD 3 controller in CANFD3DIV (CLK_CLKDIV0[14:12]).
  - Enable the clock of CAN FD 3 controller in CAN3CKEN (CLK_SYSCLK0[11]).
- Reset Configuration
  - Reset CAN FD 3 controller in CANFD3RST (SYS_IPRST3[23]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
CAN FD 3	CAN3_RXD	PA.14	MFP3
		PD.14	MFP3
		PG.8	MFP3
		PJ.6	MFP3
		PJ.14	MFP3
		PK.14	MFP3
		PL.2	MFP3
		PL.10	MFP3
		PM.2	MFP3
		PN.14	MFP3
	CAN3_TXD	PG.0	MFP3
		PD.15	MFP3
		PG.9	MFP3
		PJ.7	MFP3
		PJ.15	MFP3
		PK.15	MFP3
		PL.3	MFP3
		PL.11	MFP3
		PM.3	MFP3
		PN.15	MFP3

6.33.5 Functional Description

6.33.5.1 Operation Mode

Software Initialization

Software initialization is started by setting bit INIT (CANFD_CCCR[0]) bit, either by software or by a hardware reset, when an uncorrected bit error was detected in the Message RAM, or by going Bus_Off.

While INIT (CANFD_CCCR[0]) bit is set, message transfer from and to the CAN bus is stopped, the status of the CAN bus output CANx_TXD pin is recessive (HIGH). The counters of the Error Management Logic EML are unchanged. Setting INIT (CANFD_CCCR[0]) bit does not change any configuration register. Resetting INIT (CANFD_CCCR[0]) bit finishes the software initialization. Afterwards the Bit Stream Processor (BSP) synchronizes itself to the data transfer on the CAN bus by waiting for the occurrence of a sequence of 11 consecutive recessive bits (=Bus_Idle) before it can take part in bus activities and start the message transfer.

Access to the CAN FD configuration registers is only enabled when both bits INIT (CANFD_CCCR[0]) and CCE (CANFD_CCCR[1]) are set (protected write). CCE (CANFD_CCCR[1]) can only be set/reset while INIT (CANFD_CCCR[0]) = 1. CCE (CANFD_CCCR[1]) is automatically reset when INIT (CANFD_CCCR[0]) is reset.

The following registers are reset when CCE (CANFD_CCCR[1]) is set:

- HPMS - High Priority Message Status
- RXF0S - Rx FIFO 0 Status
- RXF1S - Rx FIFO 1 Status
- TXFQS - Tx FIFO/Queue Status
- TXBRP - Tx Buffer Request Pending
- TXBTO - Tx Buffer Transmission Occurred
- TXBCF - Tx Buffer Cancellation Finished
- TXEFS - Tx Event FIFO Status

The Timeout Counter value TOC (CANFD_TOCV[15:0]) is preset to the value configured by TOP (CANFD_TOCC[31:16]) when CCE (CANFD_CCCR[1]) is set.

In addition the state machines of the Tx Handler and Rx Handler are held in idle state while CCE (CANFD_CCCR[1]) bit is set.

The following registers are only writeable while CCE (CANFD_CCCR[1]) bit is reset.

- TXBAR - Tx Buffer Add Request
- TXBCR - Tx Buffer Cancellation Request

TEST (CANFD_CCCR[7]) bit and MON (CANFD_CCCR[5]) bit can only be set by the software while both of INIT (CANFD_CCCR[0]) bit and CCE (CANFD_CCCR[1]) bit are set. Both bits may be reset at any time. DAR (CANFD_CCCR[6]) can only be set/reset while both INIT (CANFD_CCCR[0]) and CCE (CANFD_CCCR[1]) are set.

Normal Operation

Once the CAN FD controller is initialized and INIT (CANFD_CCCR[0]) is reset to zero, the CAN FD controller synchronizes itself to the CAN bus and is ready for communication.

After passing the acceptance filtering, received messages including Message ID and DLC are stored into a dedicated Rx Buffer or into Rx FIFO 0 or Rx FIFO 1.

For messages to be transmitted dedicated Tx Buffers and/or a Tx FIFO or a Tx queue can be initialized or updated. Automated transmission on reception of remote frames is not implemented.

CAN FD Operation

There are two variants in the CAN FD frame transmission:

(1). The CAN FD frame without bit rate switching.

(2). The CAN FD frame where control field, data field, and CRC field are transmitted with a higher bit

rate than the beginning and the end of the frame.

The previously reserved bit in CAN frames with 11-bit identifiers and the first previously reserved bit in CAN frames with 29-bit identifiers will now be decoded as FDF bit.

If FDF is recessive, it signifies a CAN FD frame.

If FDF is dominant, it signifies a Classic CAN frame.

In a CAN FD frame, the two bits following FDF, res and BRS, decide whether the bit rate inside of this CAN FD frame is switched. A CAN FD bit rate switch is signified by res = dominant and BRS = recessive. The coding of res = recessive is reserved for future expansion of the protocol.

In case the CAN FD controller receives a frame with FDF = recessive and res = recessive, it will signal a Protocol Exception Event by setting bit PXE (CANFD_PSR[14]) bit. When Protocol Exception Handling is enabled (PXHD (CANFD_CCCR[12]) is reset), this causes the operation state to change from Receiver (ACT (CANFD_PSR[4:3]) = 2'b10) to Integrating (ACT (CANFD_PSR[4:3]) = 00) at the next sample point.

In case Protocol Exception Handling is disabled (PXHD (CANFD_CCCR[12]) is set), the CAN FD controller will treat a recessive res bit as a form error and will respond with an error frame.

CAN FD operation is enabled by programming FDOE (CANFD_CCCR[8]). In case FDOE (CANFD_CCCR[8]) = 1, transmission and reception of CAN FD frames is enabled. Transmission and reception of Classic CAN frames is always possible. Whether a CAN FD frame or a Classic CAN frame is transmitted can be configured via bit FDF in the respective Tx Buffer element. With FDOE (CANFD_CCCR[8]) = 0, received frames are interpreted as Classic CAN frames, which leads to the transmission of an error frame when receiving a CAN FD frame. When CAN FD operation is disabled, no CAN FD frames are transmitted even if bit FDF of a Tx Buffer element is set. FDOE (CANFD_CCCR[8]) and BRSE (CANFD_CCCR[9]) can only be changed while INIT (CANFD_CCCR[0]) and CCE (CANFD_CCCR[1]) are both set.

With FDOE (CANFD_CCCR[8]) = 0, the setting of bits FDF and BRS is ignored and frames are transmitted in Classic CAN format. If FDOE (CANFD_CCCR[8]) is set and BRSE (CANFD_CCCR[9]) is clear, only bit FDF of a Tx Buffer element is evaluated. If both FDOE (CANFD_CCCR[8]) and BRSE (CANFD_CCCR[9]) are set, transmission of CAN FD frames with bit rate switching is enabled. All Tx Buffer elements with bits FDF and BRS set are transmitted in CAN FD format with bit rate switching.

A mode change during CAN operation is only recommended under the following conditions:

- The failure rate in the CAN FD data phase is significant higher than in the CAN FD arbitration phase. In this case, disable the CAN FD bit rate switching option for transmissions.
- During system startup all nodes are transmitting Classic CAN messages until it is verified that they are able to communicate in CAN FD format. If this is true, all nodes switch to CAN FD operation.
- Wake-up messages in CAN Partial Networking have to be transmitted in Classic CAN format.
- End-of-line programming in case not all nodes are CAN FD capable. Non CAN FD nodes are held in Silent mode until programming has completed. Then all nodes switch back to Classic CAN communication.

In the CAN FD format, the coding of the DLC differs from the standard CAN format. The DLC codes 0 to 8 have the same coding as in standard CAN, the codes 9 to 15, in standard CAN all code a data field of 8 bytes, which are coded according to Table 6.33-1 below.

DLC	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Number of Data Bytes	12	16	20	24	32	48	64

Table 6.33-1 DLC Coding in CAN FD Format

In CAN FD frames, the bit timing will be switched inside the frame, after the BRS (Bit Rate Switch) bit, if this bit is recessive. Before the BRS bit, in the CAN FD arbitration phase, the nominal CAN bit timing is used as defined by the Nominal Bit Timing and Prescaler Register NBTP. In the following CAN FD data phase, the data phase bit timing is used as defined by the Data Bit Timing and Prescaler Register DBTP. The bit timing is switched back from the data phase timing at the CRC delimiter or when an error is detected, whichever occurs first.

The maximum configurable bit rate in the CAN FD data phase depends on the CAN Core clock frequency (cclk). Example: with a CAN FD controller clock frequency of 20 MHz and the shortest configurable bit time of 4 time quanta (tq), the bit rate in the data phase is 5 Mbit/s.

In both data frame formats, CAN FD and CAN FD with bit rate switching, the value of the bit ESI (Error Status Indicator) is determined by the transmitter's error state at the start of the transmission. If the transmitter is error passive, ESI is transmitted recessive, else it is transmitted dominant.

Transmitter Delay Compensation

During the data phase of a CAN FD transmission only one node is transmitting, all others are receivers. The length of the bus line has no impact. When transmitting via pin CANx_TXD the CAN FD Controller receives the transmitted data from its local CAN transceiver via pin CANx_RXD. The received data is delayed by the transceiver loop delay. In case this delay is greater than TSEG1 (time segment before sample point), a bit error is detected. In order to enable a data phase bit time that is even shorter than the transceiver loop delay, the delay compensation is introduced. Without transceiver delay compensation, the bit rate in the data phase of a CAN FD frame is limited by the transceiver loop delay.

The CAN FD Controller implements a delay compensation mechanism to compensate the CAN transceiver loop delay, thereby enabling transmission with higher bit rates during the CAN FD data phase independent of the delay of a specific CAN transceiver.

To check for bit errors during the data phase of transmitting nodes, the delayed transmit data is compared against the received data at the Secondary Sample Point (SSP). If a bit error is detected, the transmitter will react on this bit error at the next following regular sample point. During arbitration phase the delay compensation is always disabled.

The transmitter delay compensation enables configurations where the data bit time is shorter than the transmitter delay, it is described in detail in ISO 11898-1:2015. It is enabled by setting bit TDC (CANFD_DBTP[23]).

The received bit is compared against the transmitted bit at the SSP. The SSP position is defined as the sum of the measured delay from the CAN FD controllers transmit output CANx_TXD pin through the transceiver to the receive input CANx_RXD pin plus the transmitter delay compensation offset as configured by TDCO (CANFD_TDCR[[14:8]). The transmitter delay compensation offset is used to adjust the position of the SSP inside the received bit (e.g. half of the bit time in the data phase). The position of the secondary sample point is rounded down to the next integer number of mtq (minimum time quantum).

TDCV (CANFD_PSR[22:16]) shows the actual transmitter delay compensation value. TDCV (CANFD_PSR[22:16]) is cleared when INIT (CANFD_CCCR[0]) is set and is updated at each transmission of an FD frame while TDC (CANFD_DBTP[23]) is set.

The following boundary conditions have to be considered for the transmitter delay compensation implemented in the CAN FD controller:

- The sum of the measured delay from CANx_TXD to CANx_RXD and the configured transmitter delay compensation offset TDCO (CANFD_TDCR[[14:8]) has to be less than 6 bit times in the data phase.
- The sum of the measured delay from CANx_TXD to CANx_RXD and the configured transmitter delay compensation offset TDCO (CANFD_TDCR[[14:8]) has to be less than or equal to 127 mtq (minimum time quantum). In case this sum exceeds 127 mtq, the maximum value of 127 mtq is used f or transmitter delay compensation.
- The data phase ends at the sample point of the CRC delimiter, that stops checking of

receive bits at the SSPs.

If transmitter delay compensation is enabled by programming TDC (CANFD_DBTP[23]) is set, the measurement is started within each transmitted CAN FD frame at the falling edge of bit FDF to bit res. The measurement is stopped when this edge is seen at the receive input CANx_RXD pin of the transmitter. The resolution of this measurement is one mtq (minimum time quantum).

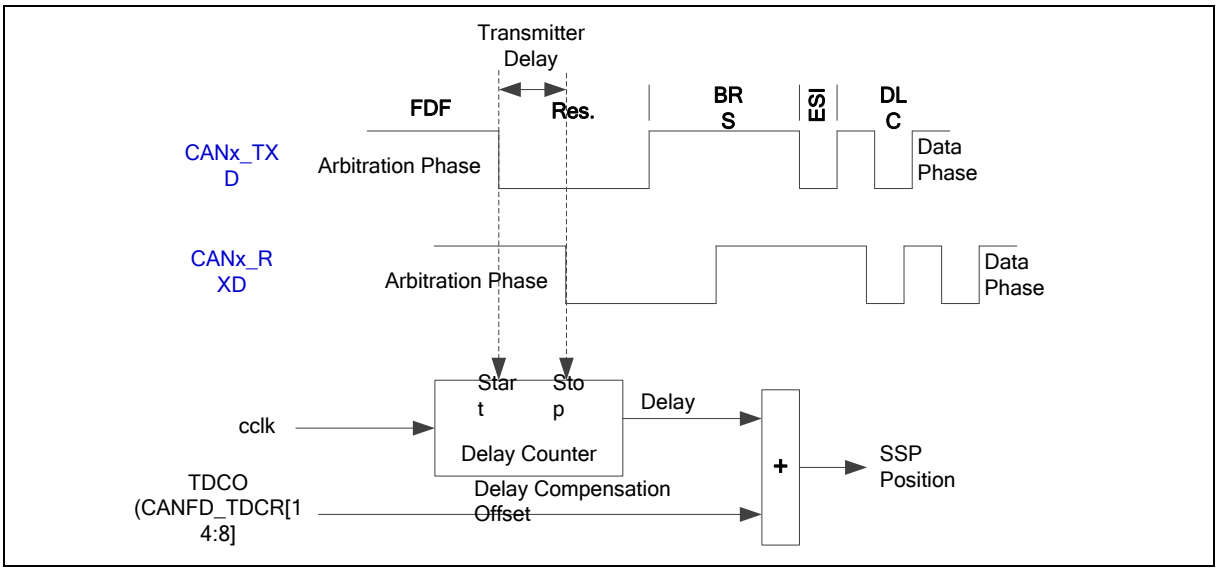


Figure 6.33-2 Transceiver Delay Measurement

To avoid that a dominant glitch inside the received FDF bit ends the delay compensation measurement before the falling edge of the received res bit, resulting in a too early SSP position, the use of a transmitter delay compensation filter window can be enabled by programming TDCF (CANFD_TDCR[6:0]). This defines a minimum value for the SSP position. Dominant edges on CANx_RXD, that would result in an earlier SSP position are ignored for transmitter delay measurement. The measurement is stopped when the SSP position is at least TDCF (CANFD_TDCR[6:0]) AND CANx_RXD is low.

Restricted Operation Mode

In Restricted Operation Mode the node is able to receive data and remote frames and to give acknowledge to valid frames, but it does not send data frames, remote frames, active error frames, or overload frames. In case of an error condition or overload condition, it does not send dominant bits, instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication. The error counters (REC (CANFD_ECR[14:8]), TEC (CANFD_ECR[7:0])) are frozen while Error Logging (CEL (CANFD_ECR[23:16])) is active. The software can set the CAN FD controller into Restricted Operation mode by setting bit ASM (CANFD_CCCR[2]). The bit can only be set by the software when both CCE (CANFD_CCCR[1]) and INIT (CANFD_CCCR[0]) are set to 1. The bit can be reset by the software at any time.

Restricted Operation Mode is automatically entered when the Tx Handler was not able to read data from the Message RAM in time. To leave Restricted Operation Mode, the software has to reset ASM (CANFD_CCCR[2]).

The Restricted Operation Mode can be used in applications that adapt themselves to different CAN bit rates. In this case the application tests different bit rates and leaves the Restricted Operation Mode after it has received a valid frame.

Note: The Restricted Operation Mode must not be combined with the Loop Back Mode (internal or external).

Bus Monitor Mode

The CAN FD controller is set in Bus Monitoring Mode by programming MON (CANFD_CCCR[5]) to one. In Bus Monitoring Mode (refer to ISO 11898-1:2015, 10.14 Bus monitoring), the CAN FD controller is able to receive valid data frames and valid remote frames, but cannot start a transmission. In this mode, it sends only recessive bits on the CAN bus. If the controller is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is rerouted internally so that the controller monitors this dominant bit, although the CAN bus may remain in recessive state. In Bus Monitoring Mode register CANFD_TXBRP is held in reset state.

The Bus Monitoring Mode can be used to analyze the traffic on a CAN bus without affecting it by the transmission of dominant bits. Figure 6.33-3 shows the connection of signals CANx_TXD pin and CANx_RXD pin to the CAN FD controller in Bus Monitoring Mode.

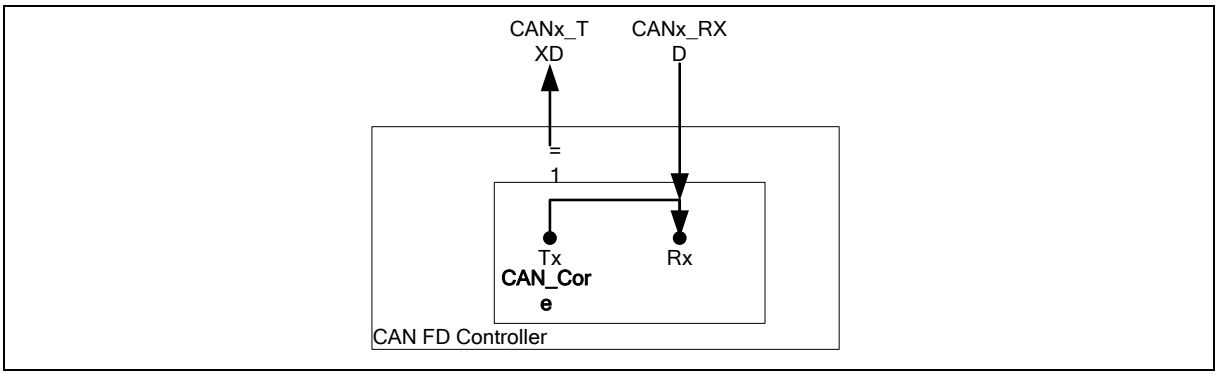


Figure 6.33-3 Pin Controller in Bus Monitor Mode

Disable Automatic Retransmission (DAR) mode

According to the CAN Specification (refer to ISO 11898-1:2015, 8.3.4 Recovery Management), the CAN FD controller provides means for automatic retransmission of frames that have lost arbitration or that have been disturbed by errors during transmission. By default, automatic retransmission is enabled. To support time-triggered communication as described in ISO 11898-1:2015, chapter 9.2, the automatic retransmission may be disabled via DAR (CANFD_CCCR[6]).

Frame Transmission in DAR Mode

In DAR mode all transmissions are automatically cancelled after they started on the CAN bus. A Tx Buffer's Tx Request Pending bit TRPx (CANFD_TXBRP[31:0]) is reset after successful transmission, when a transmission has not yet been started at the point of cancellation, has been aborted due to lost arbitration, or when an error occurred during frame transmission.

- Successful transmission:
  - Corresponding Tx Buffer Transmission Occurred bit TOn (CANFD_TXBTO[31:0]) set
  - Corresponding Tx Buffer Cancellation Finished bit CFn (CANFD_TXBCF[31:0]) not set
- Successful transmission in spite of cancellation:
  - Corresponding Tx Buffer Transmission Occurred bit TOn (CANFD_TXBTO[31:0]) set
  - Corresponding Tx Buffer Cancellation Finished bit CFn (CANFD_TXBCF[31:0]) set
- Arbitration lost or frame transmission disturbed:
  - Corresponding Tx Buffer Transmission Occurred bit TOn (CANFD_TXBTO[31:0]) not set
  - Corresponding Tx Buffer Cancellation Finished bit CFn (CANFD_TXBCF[31:0]) set

In case of a successful frame transmission, and if storage of Tx events is enabled, a Tx Event FIFO element is written with Event Type ET = 10 (transmission in spite of cancellation).

Power Mode

The CAN FD controller can be set into Power-down mode controlled by input signal power_down_req or via CSR (CANFD_CCCR[4]). As long as the clock stop request signal power_down_req is active, bit CSR (CANFD_CCCR[4]) is read as one.

When all pending transmission requests have completed, the CAN FD controller waits until bus idle state is detected. Then the CAN FD controller sets then INIT (CANFD_CCCR[0]) to one to prevent any further CAN transfers. Now the CAN FD controller acknowledges that it is ready for power down by setting output signal power_down_ack to one and CSA(CANFD_CCCR[3]) to one. Before the clocks are switched off, further register accesses can be made except for INIT (CANFD_CCCR[0]) which is held at one.

Note: In case of a heavily disturbed CAN bus, it may happen that idle state is never reached and INIT (CANFD_CCCR[0]) is therefore not set by the controller. This situation can be detected by polling ACT (CANFD_PSR[4:3]). In case the controller does not enter idle state, the software can write INIT (CANFD_CCCR[0]) to 1 which stops CAN communication of the CAN FD controller immediately, regardless whether there is a transmission/reception ongoing or not.

To leave Power-down mode, the application has to turn on the module clocks before resetting control register flag CSR (CANFD_CCCR[4]). The controller will acknowledge this by resetting CSA (CANFD_CCCR[3]). Afterwards, the application can restart CAN communication by resetting bit INIT (CANFD_CCCR[0]).

Test Mode

To enable write access to register TEST, bit TEST (CANFD_CCCR[7]) has to be set to 1. This allows the configuration of the test modes and test functions.

Four output functions are available for the CAN transmit pin CANx_TXD by programming TX (CANFD_TEST[6:5]). Additionally to its default function – the serial data output – it can drive the CAN Sample Point signal to monitor the CAN FD controller bit timing and it can drive constant dominant or recessive values. The actual value at pin CANx_RXD can be read from RX (CANFD_TEST[7]). Both functions can be used to check the CAN bus physical layer.

Due to the synchronization mechanism between CAN Core clock and AHB clock domain, there may be a delay of several AHB clock periods between writing to TX (CANFD_TEST6:5]) until the new configuration is visible at output pin CANx_TXD. This applies also when reading input pin CANx_RXD via RX (CANFD_TEST[7]).

Note: Test modes should be used for production tests or self-test only. The software control for pin CANx_TXD interferes with all CAN protocol functions. It is not recommended to use test modes for application.

- External Loop Back Mode

  The CAN FD controller can be set in External Loop Back Mode by programming LBCK (CANFD_TEST[4]) to one. In Loop Back Mode, the controller treats its own transmitted messages as received messages and stores them (if they pass acceptance filtering) into an Rx Buffer or an Rx FIFO. Figure 6.33-4 shows the connection of signals CANx_TXD pin and CANx_RXD pin to the CAN FD controller in External Loop Back Mode.

  This mode is provided for hardware self-test. To be independent from external stimulation, the controller ignores acknowledge errors (recessive bit sampled in the acknowledge slot of a data/remote frame) in Loop Back Mode. In this mode the controller performs an internal feedback from its Tx output to its Rx input. The actual value of the CANx_RXD input pin is disregarded by the controller. The transmitted messages can be monitored at the CANx_TXD pin.

- Internal Loop Back Mode

  Internal Loop Back Mode is entered by programming bits LBCK (CANFD_TEST[4]) and MON (CANFD_CCCR[5]) to one. This mode can be used for a “Hot Selftest”, meaning

the controller can be tested without affecting a running CAN system connected to the pins CANx_TXD and CANx_RXD. In this mode pin CANx_RXD is disconnected from the controller and pin CANx_TXD is held recessive. Figure 6.33-4 shows the connection of CANx_TXD and CANx_RXD to the controller in case of Internal Loop Back Mode.

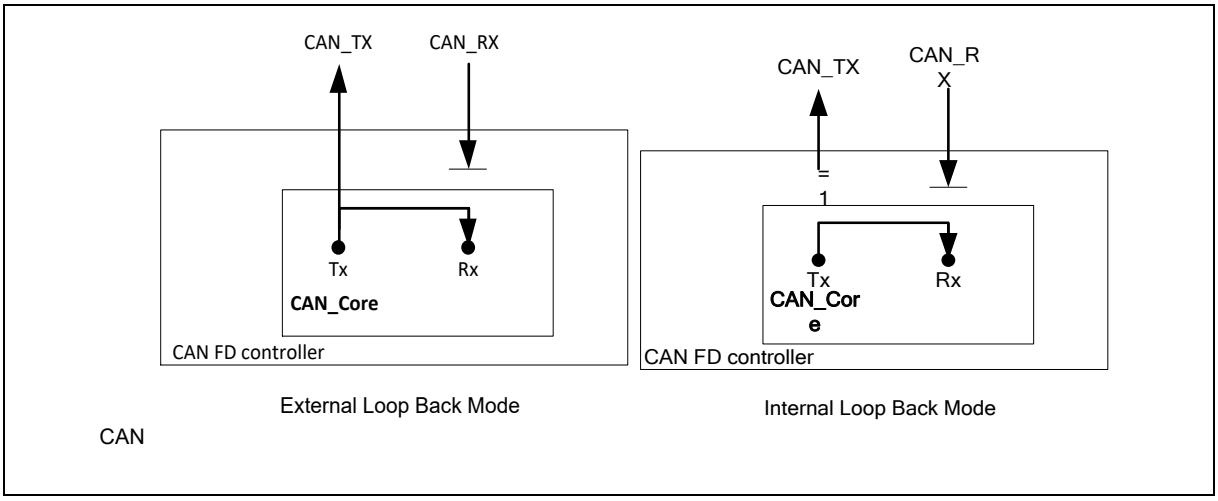


Figure 6.33-4 Pin Control in Loop Back Modes

6.33.5.2 Timestamp Generation

For timestamp generation the CAN FD controller supplies a 16-bit wrap-around counter. A pre-scaler TCP (CANFD_TSCC[19:16]) can be configured to clock the counter in multiples of CAN bit times (1…16). The counter is readable via TSC (CANFD_TSCV[15:0]). A write access to register CANFD_TSCV resets the counter to 0. When the timestamp counter wraps around interrupt flag TSW (CANFD_IR[16]) is set.

On start of frame reception / transmission (SOF) the counter value is captured and stored into the **Timestamp Generation** section of an Rx Buffer / Rx FIFO (RXTS(RX1A[15:0])) or Tx Event FIFO (TXTS(E1A[15:0])) element.

By programming bit TSS (CANFD_TSCC[[1:0]), a 16-bit timestamp can be used.

6.33.5.3 Timeout Counter

To signal timeout conditions for Rx FIFO 0, Rx FIFO 1, and the Tx Event FIFO the CANFD controller supplies a 16-bit timeout counter. It operates as down-counter and uses the same pre-scaler controlled by TCP (CANFD_TSCC[19:16]) as the timestamp counter. The timeout counter is configured via register CANFD_TOCC. The actual counter value can be read from TOC (CANFD_TOCV[15:0]). The timeout counter can only be started while INIT (CANFD_CCCR[0]) = 0. It is stopped when INIT (CANFD_CCCR[0]) = 1, e.g. when the CAN FD Controller enters Bus_Off state.

The operation mode is selected by TOS (CANFD_TOCC[2:1]). When operating in Continuous Mode, the counter starts when INIT (CANFD_CCCR[0]) is reset. A write to TOCV presets the counter to the value configured by TOP (CANFD_TOCC[31:16]) and continues down-counting.

When the timeout counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by TOP (CANFD_TOCC[31:16]). Down-counting is started when the first FIFO element is stored. Writing to TOCV has no effect.

When the counter reaches zero, interrupt flag TOO (CANFD_IR[18]) is set. In Continuous Mode, the counter is immediately restarted at TOP (CANFD_TOCC[31:16]).

Note: The clock signal for the timeout counter is derived from the CAN Core's sample point signal.

Therefore, the point in time where the timeout counter is decremented may vary due to the synchronization / re-synchronization mechanism of the CAN Core. If the bit rate switch feature in CAN FD controller is used, the timeout counter is clocked differently in arbitration and data field.

6.33.5.4 Rx Handling

The Rx Handler controls the acceptance filtering, the transfer of received messages to the Rx Buffers or to one of the two Rx FIFOs, as well as the Rx FIFO's put and get Indices.

Acceptance Filtering

The CAN FD controller offers the possibility to configure two sets of acceptance filters, one for standard identifiers and one for extended identifiers. These filters can be assigned to an Rx Buffer or to Rx FIFO 0,1. For acceptance filtering each list of filters is executed from element #0 until the first matching element. Acceptance filtering stops at the first matching element. The following filter elements are not evaluated for this message.

The main features are:

- Each filter element can be configured as
  - range filter (from 0 to 128 elements for the 11-bit filter and from 0 to 64 for the 29-bit filter)
  - filter for one or two dedicated IDs
  - classic bit mask filter
- Each filter element is configurable for acceptance or rejection filtering
- Each filter element can be enabled / disabled individually
- Filters are checked sequentially, execution stops with the first matching filter element

Related configuration registers are:

- Global Filter Configuration CANFD_GFC
- Standard ID Filter Configuration CANFD_SIDFC
- Extended ID Filter Configuration CANFD_XIDFC
- Extended ID AND Mask CANFD_XIDAM

Depending on the configuration of the filter element (SFEC/EFEC) a match triggers one of the following actions:

- Store received frame in FIFO 0 or FIFO 1
- Store received frame in Rx Buffer
- Store received frame in Rx Buffer and generate pulse at filter event pin
- Reject received frame
- Set High Priority Message interrupt flag HPM (CANFD_IR[8])
- Set High Priority Message interrupt flag HPM (CANFD_IR[8]) and store received frame in FIFO 0 or FIFO 1

Acceptance filtering is started after the complete identifier has been received. After acceptance filtering has completed, and if a matching Rx Buffer or Rx FIFO has been found, the Message Handler starts writing the received message data in portions of 32 bit to the matching Rx Buffer or Rx FIFO. If the CAN protocol controller has detected an error condition (e.g. CRC error), this message is discarded with the following impact on the affected Rx Buffer or Rx FIFO:

Rx Buffer

New Data flag of matching Rx Buffer is not set, but Rx Buffer (partly) overwritten with received data. For error type refer to LEC (CANFD_PSR[2:0]) respectively DLEC (CANFD_PSR[10:8]).

Rx FIFO

Put index of matching Rx FIFO is not updated, but related Rx FIFO element (partly) overwritten with received data. For error type refer to LEC (CANFD_PSR[2:0]) respectively DLEC (CANFD_PSR[10:8]). In case the matching Rx FIFO is operated in overwrite mode, the boundary conditions described in **Rx FIFO Overwrite Mode** section have to be considered.

Note: When an accepted message is written to one of the two Rx FIFOs, or into an Rx Buffer, the unmodified received identifier is stored independent of the filter(s) used. The result of the acceptance filter process is strongly depending on the sequence of configured filter elements.

- Range Filter

  The filter matches for all received frames with Message IDs in the range defined by SF1ID/SF2ID resp. EF1ID/EF2ID.

  There are two possibilities when range filtering is used together with extended frames:

  EFT = 2'b00: The Message ID of received frames is ANDed with the Extended ID AND Mask (CANFD_XIDAM) before the range filter is applied.

  EFT = 2'b11: The Extended ID AND Mask (CANFD_XIDAM) is not used for range filtering.

- Filter for specific IDs

  A filter element can be configured to filter for one or two specific Message IDs. To filter for one specific Message ID, the filter element has to be configured with SF1ID = SF2ID resp. EF1ID = EF2ID.

- Classic Bit Mask Filter

  Classic bit mask filtering is intended to filter groups of Message IDs by masking single bits of a received Message ID. With classic bit mask filtering SF1ID/EF1ID is used as Message ID filter, while SF2ID/EF2ID is used as filter mask.

  A zero bit at the filter mask will mask out the corresponding bit position of the configured ID filter, e.g. the value of the received Message ID at that bit position is not relevant for acceptance filtering.

  Only those bits of the received Message ID where the corresponding mask bits are one are relevant for acceptance filtering.

  In case all mask bits are one, a match occurs only when the received Message ID and the Message

  ID filter are identical. If all mask bits are zero, all Message IDs match.

- Standard Message ID Filtering

  Figure 6.33-6 below shows the flow for standard Message ID (11-bit Identifier) filtering. The Standard Message ID Filter element is described in **Message RAM** section.

  Controlled by the global filter configuration GFC and the Standard ID filter configuration (CANFD_SIDFC) Message ID, Remote Transmission Request bit (RTR), and the Identifier Extension bit (IDE) of received frames are compared against the list of configured filter elements.

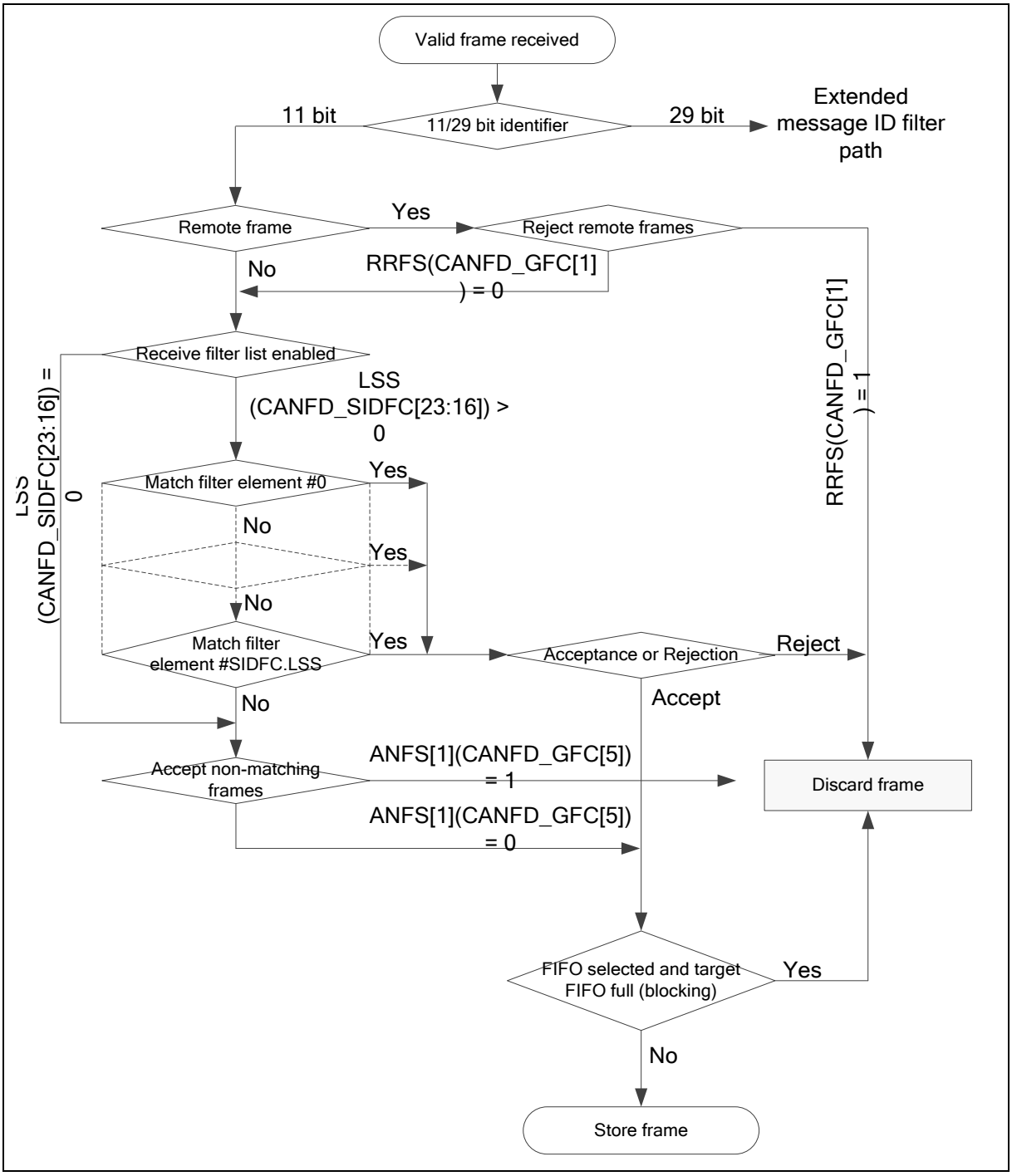


Figure 6.33-5 Standard Message ID Filter Path

- Extended Message ID Filtering

  Figure 6.33-6 shows the flow for extended Message ID (29-bit Identifier) filtering. The Extended Message ID Filter element is described in **Message RAM** Section.

  Controlled by the global filter configuration (CANFD_GFC) and the Extended ID filter configuration (CANFD_XIDFC) Message ID, Remote Transmission Request bit (RTR), and the Identifier Extension bit (IDE) of received frames are compared against the list of configured filter elements.

The Extended ID AND Mask (CANFD_XIDAM) is AND-ed with the received identifier before the filter list is executed.

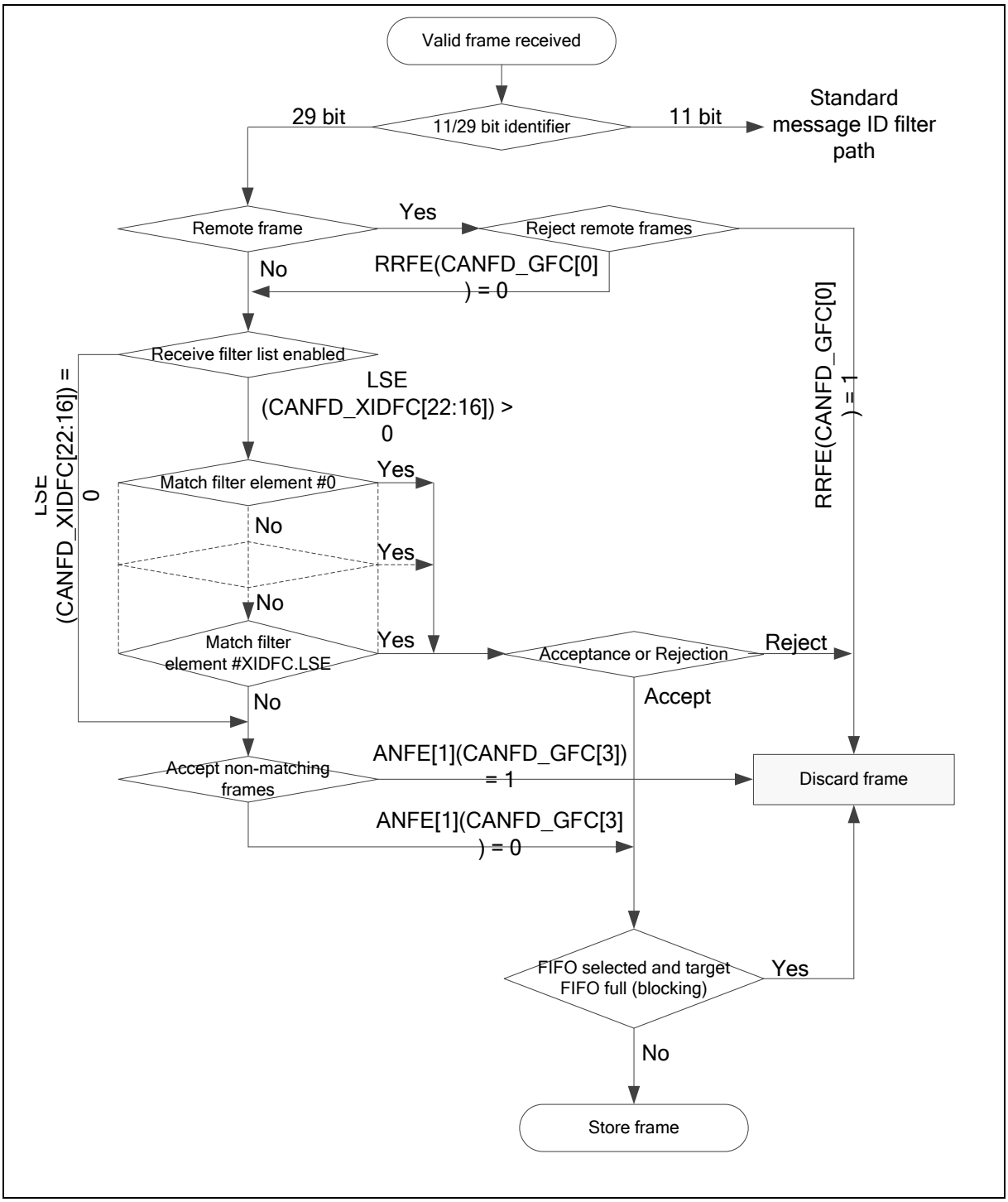


Figure 6.33-6 Extended Message ID Filter Path

Rx FIFOs

Rx FIFO 0 and Rx FIFO 1 can be configured to hold up to 64 elements each. Configuration of the two Rx FIFOs is done via registers CANFD_RXF0C and CANFD_RXF1C.

Received messages that passed acceptance filtering are transferred to the Rx FIFO as configured by

the matching filter element. For a description of the filter mechanisms available for Rx FIFO 0 and Rx FIFO 1 refer to **Acceptance Filtering** section. The Rx FIFO element is described in Rx Buffer and **FIFO Element** section.

To avoid an Rx FIFO overflow, the Rx FIFO watermark can be used. When the Rx FIFO fill level reaches the Rx FIFO watermark configured by FnWM (CANFD_RxFnC[30:24]), interrupt flag RFnW (CANFD_IR) is set. When the Rx FIFO put Index reaches the Rx FIFO get Index an Rx FIFO Full condition is signalled by FnF (CANFD_RXFnS). In addition, interrupt flag RFnF (CANFD_IR) is set.

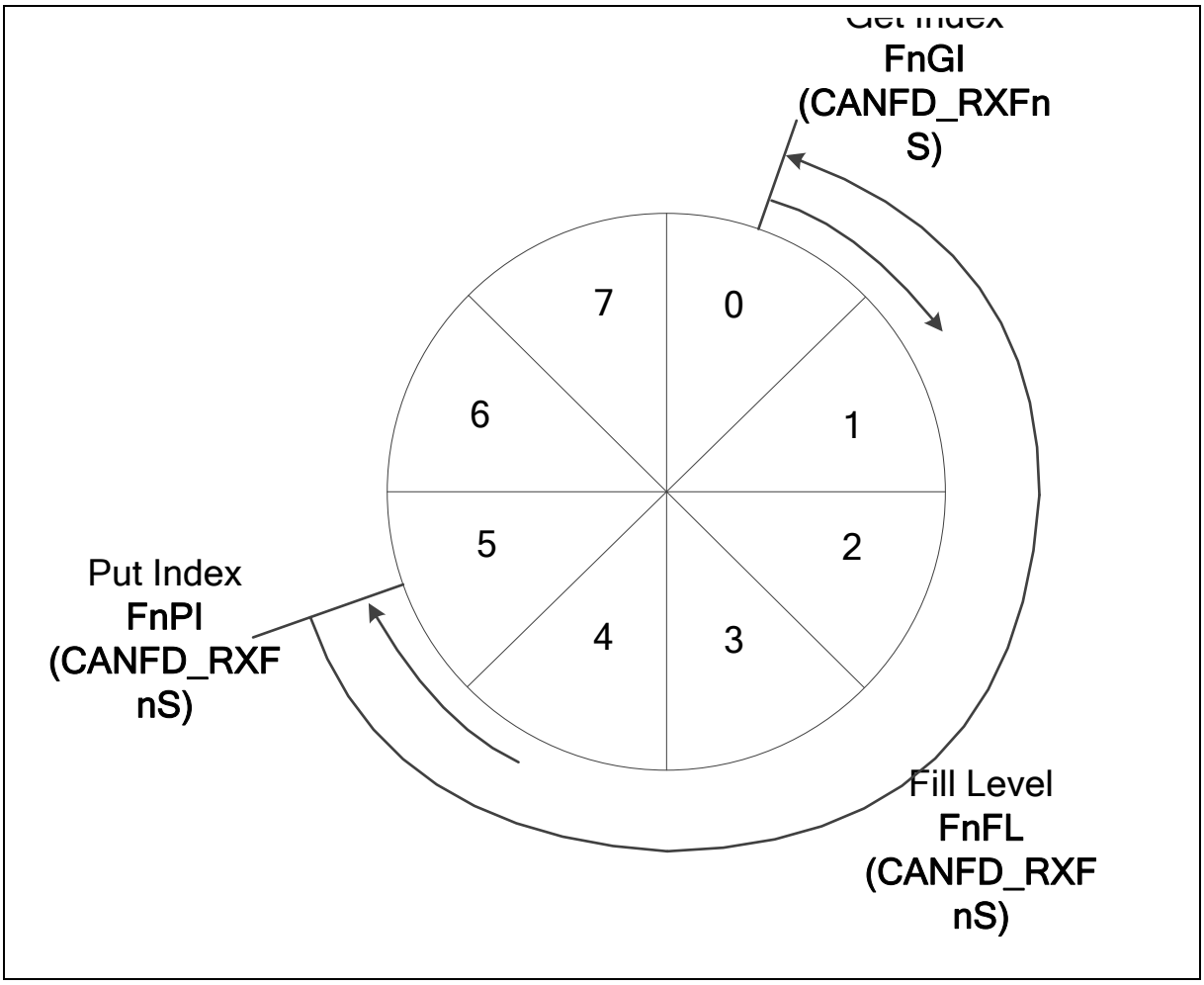


Figure 6.33-7 Rx FIFO Status

When reading from an Rx FIFO, Rx FIFO Get Index FnGI (CANFD_RXFnS) + FIFO element size has to be added to the corresponding Rx FIFO start address FnSA (CANFD_RXFnC).

RBDS(CANFD_RXESC[10:8]) FnDS(CANFD_RXESC[2:0])	Data Field [Bytes]	FIFO Element Size [RAM Words]
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8

101	32	10
110	48	14
111	64	18

Table 6.33-2 Rx Buffer / FIFO Element Size

- Rx FIFO Blocking Mode

  The Rx FIFO blocking mode is configured by FnOM (CANFD_RXFnC[31]) = 0. This is the default operation mode for the Rx FIFOs.

  When an Rx FIFO full condition is reached (FnPI (CANFD_RXFnS[21:16]) = FnGI (CANFD_RXFnS[13:8])), no further messages are written to the corresponding Rx FIFO until at least one message has been read out and the Rx FIFO Get Index has been incremented. An Rx FIFO full condition is signalled by FnF (CANFD_RXFnS[24]) = 1. In addition interrupt flag RFnF (CANFD_IR) is set.

  In case a message is received while the corresponding Rx FIFO is full, this message is discarded and the message lost condition is signalled by RFnL (CANFD_RXFnS[25]) = 1. In addition interrupt flag RFnL (CANFD_IR) is set.

- Rx FIFO Overwrite Mode

  The Rx FIFO overwrite mode is configured by FnOM (CANFD_RXFnC[31]) = 1.

  When an Rx FIFO full condition (FnPI (CANFD_RXFnS[21:16]) = FnGI (CANFD_RXFnS[13:8])) is signalled by FnF (CANFD_RXFnS[24]) = 1, the next message accepted for the FIFO will overwrite the oldest FIFO message. Put and get index are both incremented by one.

  When an Rx FIFO is operated in overwrite mode and an Rx FIFO full condition is signalled, reading of the Rx FIFO elements should start at least at get index + 1. The reason for that is, that it might happen, that a received message is written to the Message RAM (put index) while the CPU is reading from the Message RAM (get index). In this case inconsistent data may be read from the respective Rx FIFO element. Adding an offset to the get index when reading from the Rx FIFO avoids this problem. The offset depends on how fast the CPU accesses the Rx FIFO. Figure 6.33-8 shows an offset of two with respect to the get index when reading the Rx FIFO. In this case the two messages stored in element 1 and 2 are lost.

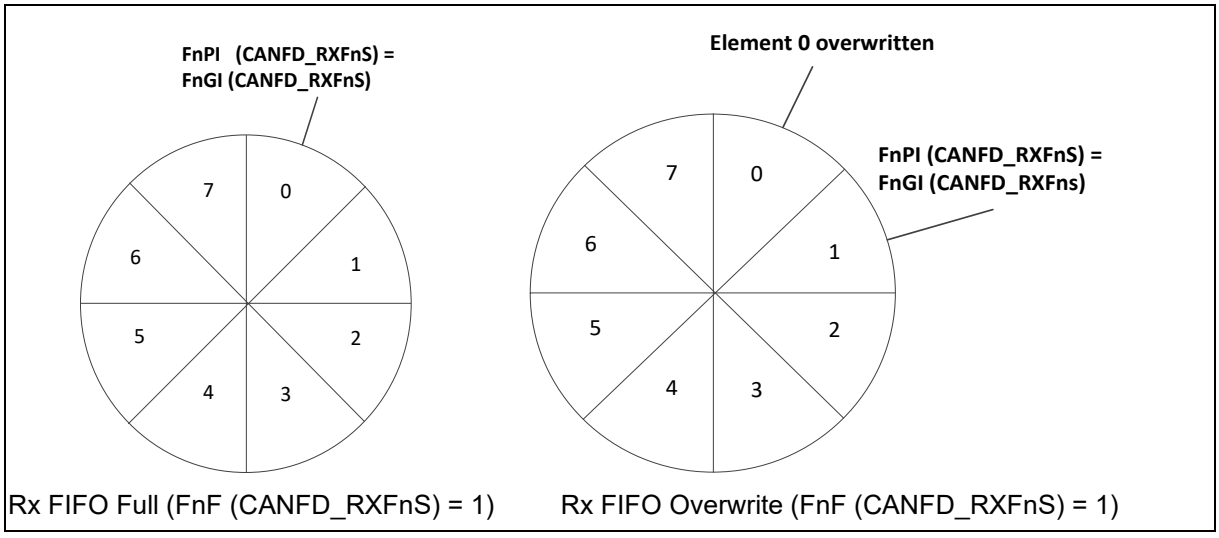


Figure 6.33-8 Rx FIFO Overflow Handling

After reading from the Rx FIFO, the number of the last element read has to be written to the Rx FIFO Acknowledge Index FnA (CANFD_RXFnA[5:0]). This increments the get index to that element number. In case the put index has not been incremented to this Rx FIFO element, the Rx FIFO full condition is reset (FnF (CANFD_RXFnS[24]) = '0').

Dedicated Rx Buffers

The CAN FD controller supports up to 64 dedicated Rx Buffers. The start address of the dedicated Rx Buffer section is configured via RBSA (CANFD_RXBC[15:2]).

For each Rx Buffer a Standard or Extended Message ID Filter Element with SFEC / EFEC = 111 and SFID2 / EFID2[10:9] = 00 has to be configured (refer to **Standard Message ID Filter Element** and **Extended Message ID Filter Element**).

After a received message has been accepted by a filter element, the message is stored into the Rx Buffer in the Message RAM referenced by the filter element. The format is the same as for an Rx FIFO element. In addition the flag DRX(CANFD_IR[19]) (Message stored in Dedicated Rx Buffer) in the interrupt register is set.

Filter Element	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID message 1	00	00 0000
1	ID message 2	00	00 0001
2	ID message 3	00	00 0010

Table 6.33-3 Example Filter Configuration for Rx Buffers

After the last word of a matching received message has been written to the Message RAM, the respective New Data flag in register NDAT1,2 is set. As long as the New Data flag is set, the respective Rx Buffer is locked against updates from received matching frames. The New Data flags have to be reset by the software by writing a '1' to the respective bit position.

While an Rx buffer new data flag is set, a message ID filter element referencing this specific Rx Buffer will not match, causing the acceptance filtering to continue. The following message ID filter elements may cause the received message to be stored into another Rx Buffer, or into an Rx FIFO, or the message may be rejected, depending on filter configuration.

- Rx Buffer Handling
  - Reset interrupt flag DRX(CANFD_IR[19])
  - Read New Data registers
  - Read messages from Message RAM
  - Reset New Data flags of processed messages

6.33.5.5 Tx Handling

The Tx Handler handles transmission requests for the dedicated Tx Buffers, the Tx FIFO, and the Tx Queue. It controls the transfer of transmit messages to the CAN Core, the Put and Get Indices, and the Tx Event FIFO. Up to 32 Tx Buffers can be set up for message transmission. The CAN mode for transmission (Classic CAN or CAN FD) can be configured separately for each Tx Buffer element. The Tx Buffer element is described in **Tx Buffer Element** section. Table 6.33-4 describes the possible configurations for frame transmission.

CANFD_CCCR	Tx Buffer Element	Frame Transmission

BRSE	FDOE	FDF	BRS	
ignored	0	ignored	ignored	Classic CAN
0	1	0	ignored	Classic CAN
0	1	1	ignored	FD without bit rate switching
1	1	0	ignored	Classic CAN
1	1	1	0	FD without bit rate switching
1	1	1	1	FD with bit rate switching

Table 6.33-4 Possible Configuration for Frame Transmission

Note: AUTOSAR requires at least three Tx Queue Buffers and support of transmit cancellation

The Tx Handler starts a Tx scan of the Message RAM to check for the highest priority pending Tx request (Tx buffer with lowest message ID) when the Tx buffer request pending register CANFD_TXBRP is updated, or when a transmission has been started. CANFD_TXBRP is updated when a transmission finishes, or after the software has requested a transmission by writing to CANFD_TXBAR, or when the Host has cancelled a pending transmission by writing to CANFD_TXBCR.

Transmit Pause

The transmit pause feature is intended for use in CAN systems where the CAN message identifiers are (permanently) specified to specific values and cannot easily be changed. These message identifiers may have a higher CAN arbitration priority than other defined messages, while in a specific application their relative arbitration priority should be inverse. This may lead to a case where one electronic control unit (ECU) sends a burst of CAN messages that cause another ECU CAN messages to be delayed because that other messages have a lower CAN arbitration priority.

If e.g. CAN ECU-1 has the transmit pause feature enabled and is requested by its application software to transmit four messages, it will, after the first successful message transmission, wait for two CAN bit times of bus idle before it is allowed to start the next requested message. If there are other ECUs with pending messages, those messages are started in the idle time, they would not need to arbitrate with the next message of ECU-1. After having received a message, ECU-1 is allowed to start its next transmission as soon as the received message releases the CAN bus.

The transmit pause feature is controlled by bit TXP (CANFD_CCCR[14]). If the bit is set, the controller will, each time it has successfully transmitted a message, pause for two CAN bit times before starting the next transmission. This enables other CAN nodes in the network to transmit messages even if their messages have lower prior identifiers. Default is transmit pause disabled (TXP (CANFD_CCCR[14]) = 0).

This feature looses up burst transmissions coming from a single node and it protects against "babbling idiot" scenarios where the application program erroneously requests too many transmissions.

Dedicated Tx Buffers

Dedicated Tx Buffers are intended for message transmission under complete control of the CPU. Each Dedicated Tx Buffer is configured with a specific Message ID. In case that multiple Tx Buffers are configured with the same Message ID, the Tx Buffer with the lowest buffer number is transmitted first.

If the data section has been updated, a transmission is requested by an Add Request via ARn (CANFD_TXBAR[31:0]). The requested messages arbitrate internally with messages from an optional Tx FIFO or Tx queue and externally with messages on the CAN bus, and are sent out according to their Message ID.

A dedicated Tx buffer allocates four 32-bit words in the Message RAM (refer to Table 6.33-5). Therefore the start address of a dedicated Tx Buffer in the Message RAM is calculated by adding transmit buffer

index (0…31) to the Tx buffer start address in TBSA (CANFD_TXBC[15:2]).

TBDS (CANFD_TXESC[2:0])	Data Field [Bytes]	Element Size [RAM Words]
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

Table 6.33-5 Tx Buffer / FIFO / Queue Element Size

Tx FIFO

Tx FIFO operation is configured by programming TFQM (CANFD_TXBC[30]) to 0. Messages stored in the Tx FIFO are transmitted starting with the message referenced by the get index TFGI (CANFD_TXFQS[12:8]). After each transmission the Get index is incremented cyclically until the Tx FIFO is empty. The Tx FIFO enables transmission of messages with the same Message ID from different Tx Buffers in the order these messages have been written to the Tx FIFO. The controller calculates the Tx FIFO Free Level TFFL (CANFD_TXFQS[5:0]) as difference between get and put index. It indicates the number of available (free) Tx FIFO elements.

New transmit messages have to be written to the Tx FIFO starting with the Tx Buffer referenced by the put index TFQPI(CANFD_TXFQS[20:16]). An add request increments the Put Index to the next free Tx FIFO element. When the put index reaches the get index, Tx FIFO Full (TFQF (CANFD_TXFQS[21]) = 1) is signaled. In this case no further messages should be written to the Tx FIFO until the next message has been transmitted and the Get Index has been incremented.

When a single message is added to the Tx FIFO, the transmission is requested by writing a 1 to the TXBAR bit related to the Tx Buffer referenced by the Tx FIFO put index.

When multiple (n) messages are added to the Tx FIFO, they are written to n consecutive Tx Buffers starting with the Put Index. The transmissions are then requested via CANFD_TXBAR. The put index is then cyclically incremented by n. The number of requested Tx buffers should not exceed the number of free Tx Buffers as indicated by the Tx FIFO Free Level.

When a transmission request for the Tx Buffer referenced by the get Index is cancelled, the Get index is incremented to the next Tx Buffer with pending transmission request and the Tx FIFO free level is recalculated. When transmission cancellation is applied to any other Tx Buffer, the get index and the FIFO free level remain unchanged.

A Tx FIFO element allocates element size 32-bit words in the Message RAM (refer to Table 6.33-5). Therefore the start address of the next available (free) Tx FIFO Buffer is calculated by adding the put index TFQPI (CANFD_TXFQS[20:16]) * element size to the Tx buffer start address TBSA (CANFD_TXBC[15:2]).

Tx Queue

Tx Queue operation is configured by programming TFQM (CANFD_TXBC[30]) to 1. Messages stored in the Tx Queue are transmitted starting with the message with the lowest Message ID (highest priority). In case that multiple Queue Buffers are configured with the same Message ID, the Queue Buffer with the lowest buffer number is transmitted first.

New messages have to be written to the Tx Buffer referenced by the Put Index TFQPI

(CANFD_TXFQS[20:16]). An Add Request cyclically increments the Put Index to the next free Tx Buffer. In case that the Tx Queue is full (TFQF (CANFD_TXFQS[21]) = 1), the Put Index is not valid and no further message should be written to the Tx Queue until at least one of the requested messages has been sent out or a pending transmission request has been cancelled.

The application may use register CANFD_TXBRP instead of the put index and may place messages to any Tx Buffer without pending transmission request.

A Tx Queue Buffer allocates element size 32-bit words in the Message RAM (refer to Table 6.33-5). Therefore the start address of the next available (free) Tx Queue Buffer is calculated by adding Tx FIFO/Queue put index TFQPI (CANFD_TXFQS[20:16]) * element size to the Tx buffer start address TBSA (CANFD_TXBC[15:2]).

Mixed Dedicated Tx Buffers / Tx FIFO

In this case the Tx Buffers section in the Message RAM is subdivided into a set of dedicated Tx Buffers and a Tx FIFO. The number of dedicated Tx Buffers is configured by NDTB (CANFD_TXBC[21:16]). The number of Tx Buffers assigned to the Tx FIFO is configured by TFQS (CANFD_TXBC[29:24]). In case TFQS (CANFD_TXBC[29:24]) is programmed to zero, only Dedicated Tx Buffers are used.

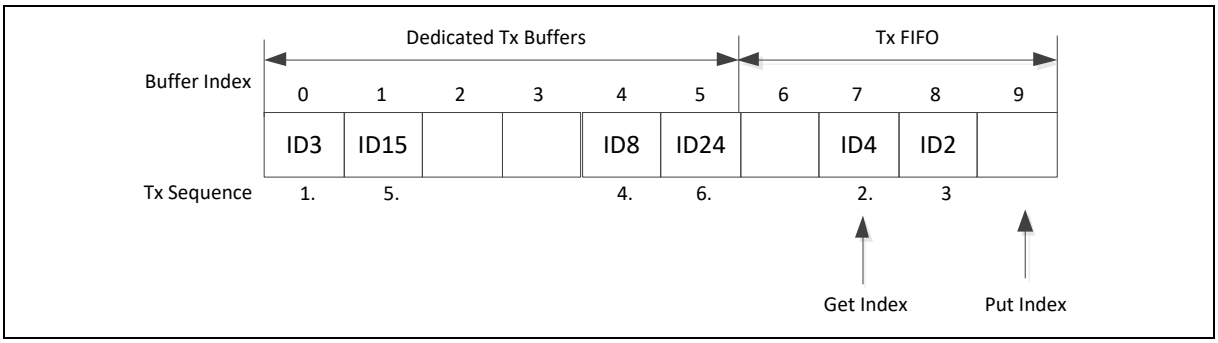


Figure 6.33-9 Example of Mixed Configuration Dedicated Tx Buffers / Tx FIFO

Tx prioritization:

- Scan Dedicated Tx Buffers and oldest pending Tx FIFO Buffer (referenced by TFGI (CANFD_TXFQS[12:8]))
- Buffer with lowest Message ID gets highest priority and is transmitted next.

Mixed Dedicated Tx Buffers / Tx Queue

In this case the Tx Buffers section in the Message RAM is subdivided into a set of dedicated Tx Buffers and a Tx Queue. The number of dedicated Tx Buffers is configured by NDTB (CANFD_TXBC[21:16]). The number of Tx Queue Buffers is configured by TFQS (CANFD_TXBC[29:24]). In case TFQS (CANFD_TXBC[29:24]) is programmed to zero, only dedicated Tx Buffers are used.

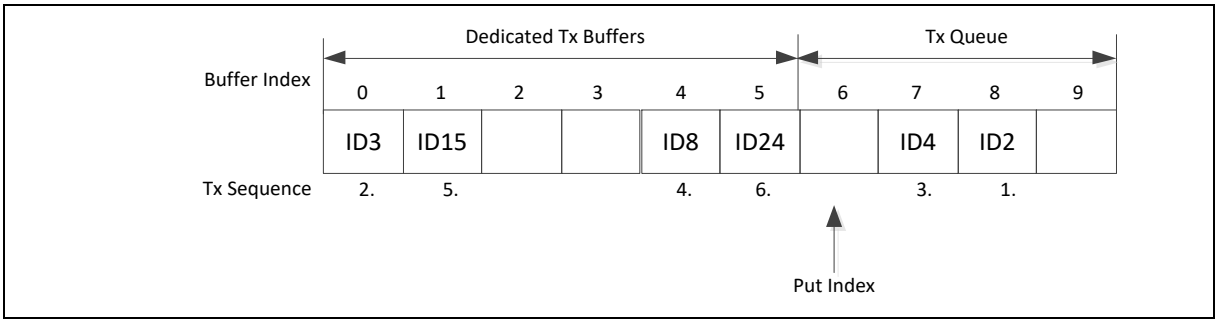


Figure 6.33-10 Example of Mixed Configuration Dedicated Tx Buffers / Tx Queue

Tx prioritization:

- Scan all Tx Buffers with activated transmission request
- Tx Buffer with lowest Message ID gets highest priority and is transmitted next

Transmit Cancellation

The CAN FD controller supports transmit cancellation. This feature is especially intended for gateway applications and AUTOSAR based applications. To cancel a requested transmission from a dedicated Tx Buffer or a Tx Queue Buffer the software has to write a 1 to the corresponding bit position (=number of Tx Buffer) of register CANFD_TXBCR. Transmit cancellation is not intended for Tx FIFO operation.

Successful cancellation is signaled by setting the corresponding bit of register CANFD_TXBCF to 1. In case a transmit cancellation is requested while a transmission from a Tx Buffer is already ongoing, the corresponding CANFD_TXBRP bit remains set as long as the transmission is in progress. If the transmission was successful, the corresponding CANFD_TXBTO and CANFD_TXBCF bits are set. If the transmission was not successful, it is not repeated and only the corresponding CANFD_TXBCF bit is set.

Note: In case a pending transmission is cancelled immediately before this transmission could have been started, there follows a short time window where no transmission is started even if another message is also pending in this node. This may enable another node to transmit a message which may have a lower priority than the second message in this node.

Tx Event Handling

To support Tx event handling the controller has implemented a Tx Event FIFO. After the controller has transmitted a message on the CAN bus, message ID and timestamp are stored in a Tx Event FIFO element. To link a Tx event to a Tx Event FIFO element, the message marker from the transmitted Tx Buffer is copied into the Tx Event FIFO element.

The Tx Event FIFO can be configured to a maximum of 32 elements. The Tx Event FIFO element is described in **Tx Event FIFO Element** section.

The purpose of the Tx Event FIFO is to decouple handling transmit status information from transmit message handling i.e. a Tx Buffer holds only the message to be transmitted, while the transmit status is stored separately in the Tx Event FIFO. This has the advantage, especially when operating a dynamically managed transmit queue, that a Tx Buffer can be used for a new message immediately after successful transmission. There is no need to save transmit status information from a Tx Buffer before overwriting that Tx Buffer.

When a Tx Event FIFO full condition is signaled by TEFF (CANFD_IR[14]), no further elements are written to the Tx Event FIFO until at least one element has been read out and the Tx Event FIFO get Index has been incremented. In case a Tx event occurs while the Tx Event FIFO is full, this event is discarded and interrupt flag TEFL (CANFD_IR[15]) is set.

To avoid a Tx Event FIFO overflow, the Tx Event FIFO watermark can be used. When the Tx Event FIFO fill level reaches the Tx Event FIFO watermark configured by EFWM (CANFD_TXEFC[29:24]), interrupt flag TEFW (CANFD_IR[13]) is set.

When reading from the Tx Event FIFO, two times the Tx Event FIFO get Index EFGI (CANFD_TXEFS[12:8]) has to be added to the Tx Event FIFO start address EFSA (CANFD_TXEFC[15:2]).

6.33.5.6 FIFO Acknowledge Handling

The get indices of Rx FIFO 0, Rx FIFO 1, and the Tx Event FIFO are controlled by writing to the corresponding FIFO Acknowledge Index (refer to **Rx FIFO 0 Acknowledge**, **Rx FIFO 1 Acknowledge**, and **Tx Event FIFO Acknowledge**). Writing to the FIFO Acknowledge Index will set the FIFO Get Index to the FIFO Acknowledge Index plus one and thereby updates the FIFO Fill Level. There are two use cases:

- When only a single element has been read from the FIFO (the one being pointed to by the get index), this get index value is written to the FIFO Acknowledge Index.

- When a sequence of elements has been read from the FIFO, it is sufficient to write the FIFO Acknowledge Index only once at the end of that read sequence (value: Index of the last element read), to update the FIFO's get index.

Due to the fact that the CPU has free access to the controller Message RAM, special care has to be taken when reading FIFO elements in an arbitrary order (get index not considered). This might be useful when reading a High Priority Message from one of the two Rx FIFOs. In this case the FIFO's Acknowledge Index should not be written because this would set the get index to a wrong position and also alters the FIFO's Fill Level. In this case some of the older FIFO elements would be lost.

Note: The application has to ensure that a valid value is written to the FIFO Acknowledge Index. The controller does not check for erroneous values.

6.33.5.7 Message RAM

For storage of Rx/Tx messages and for storage of the filter configuration a single-ported Message RAM has to be connected to the CAN Core module.

Message RAM Configuration

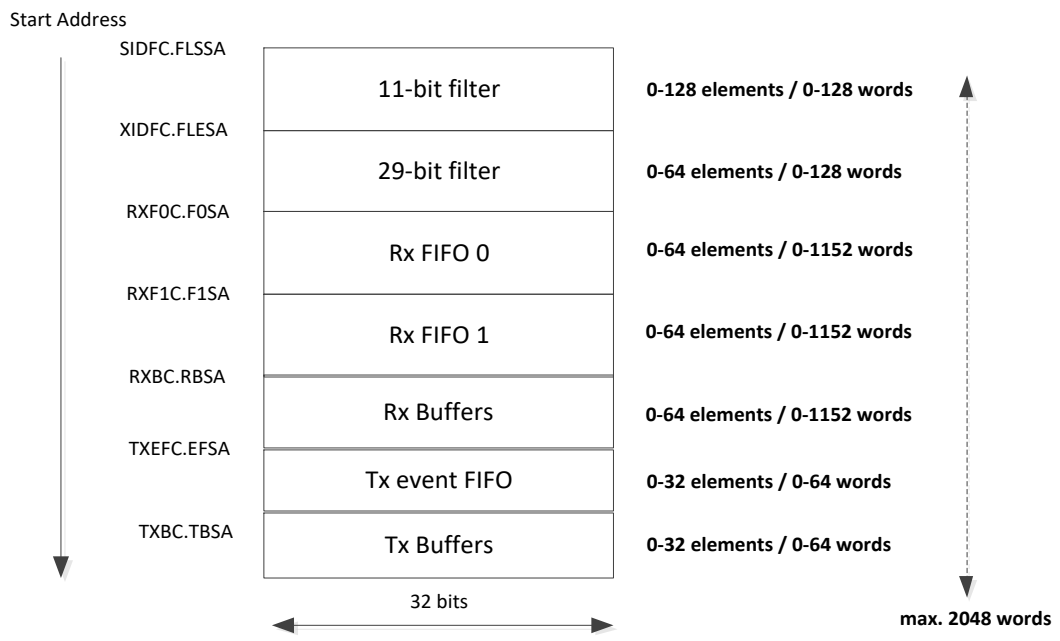
The address of the Message RAM and the address of the register are accessed through the same AHB interface. The maximum address of register used by the CAN FD controller is within 0x200. Therefore, the message SRAM used by the CAN controller must be used after 0x200. When reading or writing the SRAM content, the CAN controller will automatically subtract the offset of 0x200.

For example:

When the CPU wants to access the location of SRAM address is 0x100, its AHB address should be set as CANFD Base address + 0x200+0x100.

The Message RAM has a width of 32 bits. The CAN FD controller can be configured to allocate up to 2K words in the Message RAM. It is not necessary to configure each of the sections listed in Figure 6.33-11, nor is there any restriction with respect to the sequence of the sections.

When operated in CAN FD mode the required Message RAM size strongly depends on the element size configured for Rx FIFO0, Rx FIFO1, Rx Buffers, and Tx Buffers via F0DS (CANFD_RXESC[2:0], F1DS (CANFD_RXESC[6:4]), RBDS (CANFD_RXESC[10:8]), and TBDS (CANFD_TXESC[2:0]).



Note: The base address of 1st CANFD = 0x403C0200.

Figure 6.33-11 Message RAM Configuration

When the controller addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored.

Rx Buffer and FIFO Element

Two Rx FIFOs and up to 64 Rx Buffers can be configured in the Message RAM. Each Rx FIFO section can be configured to store up to 64 received messages. The structure of a Rx Buffer / FIFO element is shown in below. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register CANFD_RXESC.

Bit	31			24	23				16	15	8	7	0
R0	ESI	XTD	RTR	ID[28:0]									
R1	ANMF	FIDX[6:0]			-	-	FDF	BRS	DLC[3:0]	RXTS[15:0]			
R2	DB3[7:0]				DB2[7:0]					DB1[7:0]		DB0[7:0]	
R3	DB7[7:0]				DB6[7:0]					DB5[7:0]		DB4[7:0]	

Rn	DBm[7:0]				DBm-1[7:0]					DBm-2[7:0]		DBm-3[7:0]	

Table 6.33-6 Rx Buffer and FIFO Element

Bit Field	Description
R0 Bit 31 ESI	Error State Indicator 0 = Transmitting node is error active 1 = Transmitting node is error passive
R0 Bit 30 XTD	Extended Identifier Signals to the MCU whether the received frame has a standard or extended identifier. 0 = 11-bit standard identifier 1 = 29-bit extended identifier
R0 Bit 29 RTR	Remote Transmission Request Signals to the Host whether the received frame is a data frame or a remote frame. 0 = Received frame is a data frame 1 = Received frame is a remote frame Note: There are no remote frames in CAN FD format. In CAN FD frames (FDF = 1'), the dominant RRS (Remote Request Substitution) bit replaces bit RTR (Remote Transmission Request)
R0 Bits 28:0 ID[28:0]	Identifier Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].
R1 Bit 31 ANMF	Accepted Non-matching Frame Acceptance of non-matching frames may be enabled via ANFS (CANFD_GFC[5:4]) and ANFE (CANFD_GFC[3:2]). 0 = Received frame matching filter index FIDX

	1 = Received frame did not match any Rx filter element
R1 Bits 30:24 FIDX[6:0]	Filter Index 0-127=Index of matching Rx acceptance filter element (invalid if ANMF = '1'). Note: Range is 0 to LSS (CANFD_SIDFC[23:16]) – 1 for standard and 0 to LSE (CANFD_XIDFC[22:16]) – 1 for extended.
R1 Bit 21 FDF	FD Format 0 = Classical CAN frame format 1 = CAN FD frame format (new DLC-coding and CRC)
R1 Bit 20 BRS	Bit Rate Switch 0 = Frame received without bit rate switching 1 = Frame received with bit rate switching
R1 Bits 19:16 DLC[3:0]	Data Length Code 0-8 = CAN + CAN FD: received frame has 0-8 data bytes 9-15 = CAN: received frame has 8 data bytes 9-15 = CAN FD: received frame has 12/16/20/24/32/48/64 data bytes
R1 Bits 15:0 RXTS[15:0]	Rx Timestamp Timestamp Counter value captured on start of frame reception. Resolution depending on configuration of the Timestamp Counter Prescaler TCP (CANFD_TSCC[19:16]).
R2 Bits 31:24 DB3[7:0]	Data Byte 3
R2 Bits 23:16 DB2[7:0]	Data Byte 2
R2 Bits 15:8 DB1[7:0]	Data Byte 1
R2 Bits 7:0 DB0[7:0]	Data Byte 0
R3 Bits 31:24 DB7[7:0]	Data Byte 7
R3 Bits 23:16 DB6[7:0]	Data Byte 6
R3 Bits 15:8 DB5[7:0]	Data Byte 5
R3 Bits 7:0 DB4[7:0]	Data Byte 4
…	…
Rn Bits 31:24 DBm[7:0]	Data Byte mm
Rn Bits 23:16 DBm-1[7:0]	Data Byte m-1
Rn Bits 15:8 DBm-2[7:0]	Data Byte m-2
Rn Bits 7:0 DBm-3[7:0]	Data Byte m-3 Note: Depending on the configuration of the element size (RXESC), between two and sixteen 32-bit words

	(Rn = 2 ..17) are used for storage of a CAN message's data field

Table 6.33-7 Rx Buffer and FIFO Element Description

Tx Buffer Element

The Tx Buffers section can be configured to hold dedicated Tx Buffers as well as a Tx FIFO / Tx Queue. In case that the Tx Buffers section is shared by dedicated Tx buffers and a Tx FIFO / Tx Queue, the dedicated Tx Buffers start at the beginning of the Tx Buffers section followed by the buffers assigned to the Tx FIFO or Tx Queue. The Tx Handler distinguishes between dedicated Tx Buffers and Tx FIFO / Tx Queue by evaluating the Tx Buffer configuration TFQS (CANFD_TXBC[29:24]) and NDTB (CANFD_TXBC[21:16]). The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register CANFD_TXESC.

Bit	31			24	23					16	15	8	7	0
T0	ESI	XTD	RTR	ID[28:0]										
T1	MM[7:0]				EFC	-	FDF	BRS	DLC[3:0]		MM[15:8]		-	
T2	DB3[7:0]				DB2[7:0]						DB1[7:0]		DB0[7:0]	
T3	DB7[7:0]				DB6[7:0]						DB5[7:0]		DB4[7:0]	

Tn	DBm[7:0]				DBm-1[7:0]						DBm-2[7:0]		DBm-3[7:0]	

Table 6.33-8 Tx Buffer and FIFO Element

Bit Field	Description
T0 Bit 31 ESI	Error State Indicator 0 = ESI bit in CAN FD format depends only on error passive flag 1 = ESI bit in CAN FD format transmitted recessive Note: The ESI bit of the transmit buffer is OR-ed with the error passive flag to decide the value of the ESI bit in the transmitted FD frame. As required by the CAN FD protocol specification, an error active node may optionally transmit the ESI bit recessive, but an error passive node will always transmit the ESI bit recessive.
T0 Bit 30 XTD	Extended Identifier 0 = 11-bit standard identifier 1 = 29-bit extended identifier
T0 Bit 29 RTR	Remote Transmission Request 0 = Transmit data frame 1 = Transmit remote frame Note: When RTR = 1, the controller transmits a remote frame according to ISO 11898-1:2015, even if FDOE (CANFD_CCCR[8]) enables the transmission in CAN FD format.
T0 Bits 28:0 ID[28:0]	Identifier Standard or extended identifier depending on bit XTD. A standard identifier has to be written to ID[28:18].
T1 Bits 31:24 MM[7:0]	Message Marker Written by CPU during Tx Buffer configuration. Copied into Tx Event FIFO element for identification of Tx message status.
T1 Bit 23 EFC	Event FIFO Control 0 = Don't store Tx events 1 = Store Tx events

T1 Bit 21 FDF	FD Format 0 = Frame transmitted in Classic CAN format 1 = Frame transmitted in CAN FD format
T1 Bit 20 BRS	Bit Rate Switch 0 = CAN FD frames transmitted without bit rate switching 1 = CAN FD frames transmitted with bit rate switching Note: Bits ESI, FDF, and BRS are only evaluated when CAN FD operation is enabled FDOE (CANFD_CCCR[8]) = 1. Bit BRS is only evaluated when in addition BRSE (CANFD_CCCR[9]) = 1.
T1 Bits 19:16 DLC[3:0]	Data Length Code 0-8 = CAN + CAN FD: transmit frame has 0-8 data bytes 9-15 = CAN: transmit frame has 8 data bytes 9-15 = CAN FD: transmit frame has 12/16/20/24/32/48/64 data bytes
T2 Bits 31:24 DB3[7:0]	Data Byte 3
T2 Bits 23:16 DB2[7:0]	Data Byte 2
T2 Bits 15:8 DB1[7:0]	Data Byte 1
T2 Bits 7:0 DB0[7:0]	Data Byte 0
T3 Bits 31:24 DB7[7:0]	Data Byte 7
T3 Bits 23:16 DB6[7:0]:	Data Byte 6
T3 Bits 15:8 DB5[7:0]	Data Byte 5
T3 Bits 7:0 DB4[7:0]	Data Byte 4
... ...	... ...
Tn Bits 31:24 DBm[7:0]	Data Byte m
Tn Bits 23:16 DBm-1[7:0]	Data Byte m-1
Tn Bits 15:8 DBm-2[7:0]	Data Byte m-2
Tn Bits 7:0 DBm-3[7:0]	Data Byte m-3 Note: Depending on the configuration of the element size (TXESC), between two and sixteen 32-bit words (Tn = 2 ..17) are used for storage of a CAN message's data field.

Table 6.33-9 Tx Buffer Element Description

Tx Event FIFO Element

Each element stores information about transmitted messages. By reading the Tx Event FIFO the MCU gets this information in the order the messages were transmitted. Status information about the Tx Event FIFO can be obtained from register CANFD_TXEFS.

Bit	31			24	23			16	15	8	7	0
E0	ESI	XTD	RTR	ID[28:0]								
E1	MM[7:0]				ET [1:0]	FDF	BRS	DLC[3:0]	TXTS[15:0]			

Table 6.33-10 Tx Event FIFO Element

Bit Field	Description
E0 Bit 31 ESI	Error State Indicator 0 = Transmitting node is error active 1 = Transmitting node is error passive
E0 Bit 30 XTD	Extended Identifier 0 = 11-bit standard identifier 1 = 29-bit extended identifier
E0 Bit 29 RTR	Remote Transmission Request 0 = Data frame transmitted 1 = Remote frame transmitted
E0 Bits 28:0 ID[28:0]	Identifier Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].
E1 Bits 31:24 MM[7:0]	Message Marker Copied from Tx Buffer into Tx Event FIFO element for identification of Tx message status.
E1 Bit 23:22 ET[1:0]	Event Type 00 = Reserved 01 = Tx event 10 = Transmission in spite of cancellation (always set for transmissions in DAR mode) 11 = Reserved
E1 Bit 21 FDF	FD Format 0 = Classical CAN frame format 1 = CAN FD frame format (new DLC-coding and CRC)
E1 Bit 20 BRS	Bit Rate Switch 0 = Frame transmitted without bit rate switching 1 = Frame transmitted with bit rate switching
E1 Bits 19:16 DLC[3:0]	Data Length Code 0-8 = CAN + CAN FD: frame with 0-8 data bytes transmitted 9-15 = CAN: frame with 8 data bytes transmitted 9-15 = CAN FD: frame with 12/16/20/24/32/48/64 data bytes transmitted
E1 Bits 15:0 TXTS[15:0]	Tx Timestamp Timestamp Counter value captured on start of frame transmission. Resolution depending on configuration of the Timestamp Counter Prescaler TCP (CANFD_TSCC[19:16]).

Table 6.33-11 Tx Event FIFO Element Description

Standard Message ID Filter Element

Up to 128 filter elements can be configured for 11-bit IDs. When accessing a Standard Message ID Filter element, its address is the Filter List Standard Start Address FLSSA (CANFD_SIDFC[15:2]) plus the index of the filter element (0…127).

Bit	31		24	23	16	15	8	7	0
S0	SFT [1:0]	SFEC [2:0]	SFID1[10:0]			-	SFID2[10:0]		

Table 6.33-12 Standard Message ID Filter Element

Bit Field		Description
Bits 31:30 SFT[1:0]		Standard Filter Type 00 = Range filter from SFID1 to SFID2 (SFID2 ⩾ SFID1) 01 = Dual ID filter for SFID1 or SFID2 10 = Classic filter: SFID1 = filter, SFID2 = mask 11 = Filter element disabled Note: With SFT = 11 the filter element is disabled and the acceptance filtering continues (same behavior as with SFEC = 000)
Bit 29:27 SFEC[2:0]		Standard Filter Element Configuration All enabled filter elements are used for acceptance filtering of 11-bit ID frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If SFEC = 100, 101, or 110 a match sets interrupt flag HPM (CANFD_IR[8]) and, if enabled, an interrupt is generated. In this case register HPMS is updated with the status of the priority match. 000 = Disable filter element 001 = Store in Rx FIFO 0 if filter matches 010 = Store in Rx FIFO 1 if filter matches 011 = Reject ID if filter matches, not intended to be used with Sync messages 100 = Set priority if filter matches, not intended to be used with Sync messages, no storage 101 = Set priority and store in FIFO 0 if filter matches 110 = Set priority and store in FIFO 1 if filter matches 111 = Store into Rx Buffer, configuration of SFT[1:0] ignored
Bits 26:16 SFID1[10:0]		Standard Filter ID 1 First ID of standard ID filter element. When filtering for Rx Buffers, Sync messages, this field defines the ID of the message to be stored. The received identifiers must match exactly, no masking mechanism is used.
Bits 15:0	SFID2[10:0]	Standard filter ID 2 This bit field has a different meaning depending on the configuration of SFEC: SFEC = 001 ... 110 Second ID of standard ID filter element SFEC = 111 Filter for Rx buffers
	SFID2[10:9]	Decides whether the received message is stored into an Rx buffer . 00: Store message into an Rx buffer
	SFID2[8:6]	The bits are used to control the filter event pins at the Extension Interface. A '1' at the respective bit position enables generation of a pulse at the related filter event pin with the duration of one system clock period in case the filter matches. SFID2[8] is used by the calibration unit.
	SFID2[5:0]	Defines the offset to the Rx buffer start address RBSA (CANFD_RXBC[15:2]) for storage of a matching message.

Table 6.33-13 Standard Message ID Filter Element Description

Extended Message ID Filter Element

Up to 64 filter elements can be configured for 29-bit IDs. When accessing an Extended Message ID Filter element, its address is the Filter List Extended Start Address FLESA (CANFD_XIDFC[15:2]) plus two times the index of the filter element (0…63).

Bit	31		24	23	16	15	8	7	0
F0	EFECV [2:0]		EFID1[28:0]						
F0	EFT [1:0]	-	EFID2[28:0]						

Table 6.33-14 Extended Message ID Filter Element

Field		Description
F0 Bit 31:29 EFEC[2:0]		Extended Filter Element Configuration All enabled filter elements are used for acceptance filtering of 29-bit ID frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If EFEC = 100, 101, or 110 a match sets interrupt flag HPM (CANFD_IR[8]) and, if enabled, an interrupt is generated. In this case register HPMS is updated with the status of the priority match. 000 = Disable filter element 001 = Store in Rx FIFO 0 if filter matches 010 = Store in Rx FIFO 1 if filter matches 011 = Reject ID if filter matches 100 = Set priority if filter matches 101 = Set priority and store in FIFO 0 if filter matches 110 = Set priority and store in FIFO 1 if filter matches 111 = Store into Rx Buffer, configuration of EFT[1:0] ignored
F0 Bits 28:0 EFID1[28:0]		Extended Filter ID 1 First ID of extended ID filter element. When filtering for Rx Buffers, Sync messages, this field defines the ID of an extended message to be stored. The received identifiers must match exactly, only CANFD_XIDAM masking mechanism (refer to Extended Message ID Filtering) is used.
F1 Bits 31:30 EFT[1:0]		Extended Filter Type 00 = Range filter from EFID1 to EFID2 (EFID2 >- EFID1) 01 = Dual ID filter for EFID1 or EFID2 10 = Classic filter: EFID1 = filter, EFID2 = mask 11 = Range filter from EFID1 to EFID2 (EFID2 >= EFID1), CANFD_XIDAM mask not applied.
F1 Bits 28:0	EFID2[10:0]	Extended filter ID 2 This bit field has a different meaning depending on the configuration of EFEC: – SFEC = 001 ... 110 Second ID of extended ID filter element – SFEC = 111 Filter for Rx buffers
	EFID2[10:9]	Decides whether the received message is stored into an Rx Buffer. 00 = Store message into an Rx Buffer
	EFID2[8:6]	Is used to control the filter event pins at the Extension Interface. A one at the respective bit position enables generation of a pulse at the related filter event pin with the duration of one system clock period in case the filter matches.
	EFID2[5:0]	Defines the offset to the Rx Buffer Start Address RBSA (CANFD_RXBC[15:2]) for storage of a matching message.

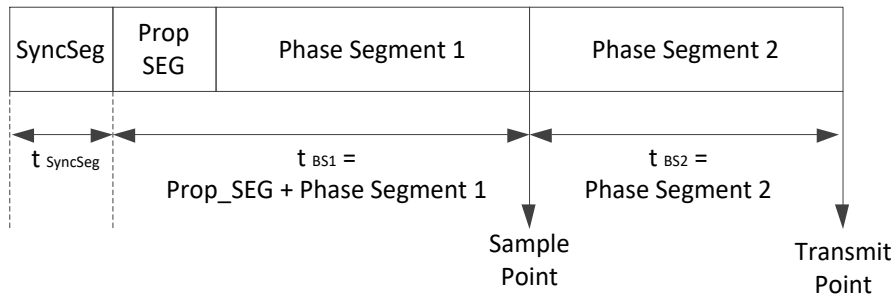
Table 6.33-15 Extended Message ID Filter Element Description

6.33.5.8 Bit Timing

The bit timing logic monitors the serial bus-line and performs sampling and adjustment of the sample point by synchronizing on the start-bit edge and resynchronizing on the following edges.

As shown in Figure 6.33-12, its operation may be explained simply by splitting the bit time in three segments, as follows:

- Synchronization segment (SYNC_SEG): a bit change is expected to occur within this time segment, having a fixed length of one time quantum (1 x tq).
- Bit segment 1 (BS1): defines the location of the sample point. It includes the PROP_SEG and PHASE_SEG1 of the CAN standard. Its duration is programmable between 1 and 16 time quanta, but may be automatically lengthened to compensate for positive phase drifts due to differences in the frequency of various nodes of the network.
- Bit segment 2 (BS2): defines the location of the transmit point. It represents the PHASE_SEG2 of the CAN standard, its duration is programmable between one and eight time quanta, but may also be automatically shortened to compensate for negative phase drifts.



Note: BS = Bit segment

Figure 6.33-12 Bit Timing

The baud rate is the inverse of bit time (baud rate = 1 / bit time), which, in turn, is the sum of three components. Figure 6.33-12 indicates that bit time = $t_{SyncSeg} + t_{BS1} + t_{BS2}$, where:

for the nominal bit time

– tq = (NBRP (CANFD_NBTP[24:16])+ 1) * $t_{canfd_tq_clk}$

– $t_{SyncSeg}$ = 1 tq

– t_{BS1} = tq * (NTSEG1(CANFD_NBTP[15:8]) + 1)

– t_{BS2} = tq * (NTSEG2(CANFD_NBTP[6:0]) + 1)

for the data bit time

– tq = (DBRP(CANFD_DBTP[20:16]) + 1) * $t_{canfd_tq_clk}$

– $t_{SyncSeg}$ = 1 tq

– t_{BS1} = tq * (DTSEG1(CANFD_DBTP[12:8]) + 1)

– t_{BS2} = tq * (DTSEG2(CANFD_DBTP[7:4]) + 1)

The (Re)Synchronization Jump Width (SJW) defines an upper bound for the amount of lengthening or shortening of the bit segments. It is programmable between one and four time quanta.

A valid edge is defined as the first transition in a bit time from dominant to recessive bus level, provided the controller itself does not send a recessive bit.

If a valid edge is detected in BS1 instead of SYNC_SEG. BS1 is extended by up to SJW so that the sample point is delayed.

Conversely, if a valid edge is detected in BS2 instead of SYNC_SEG. BS2 is shortened by up to SJW so that the transmit point is moved earlier.

As a safeguard against programming errors, the configuration of the Bit Timing register is only possible while the device is in Standby mode. Registers CANFD_DBTP and CANFD_NBTP (dedicated, respectively, to data and nominal bit timing) are only accessible when CCE (CANFD_CCCR[1]) and INIT (CANFD_CCCR[0]) are set.

Note: For a detailed description of the CAN bit timing and resynchronization mechanism, refer to the ISO 11898-1 standard.

6.33.6 Register Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
CANFD Base Address: **CANFDn_BA = 0x403C_0000+ (0x10000 *n)** **n=0, 1, 2, 3**				
CANFD_DBTP	CANFDn_BA+0x00C	R	Data Bit Timing and Prescaler Register (P*)	0x0000_0A33
CANFD_TEST	CANFDn_BA+0x010	R	Test Register (P*)	0x0000_0000
CANFD_RWD	CANFDn_BA+0x014	R	RAM Watchdog (P*)	0x0000_0000
CANFD_CCCR	CANFDn_BA+0x018	R/W	CC Control Register (Pp*)	0x0000_0001
CANFD_NBTP	CANFDn_BA+0x01C	R	Nominal Bit Timing and Prescaler Register (P*)	0x0600_0A03
CANFD_TSCC	CANFDn_BA+0x020	R	Timestamp Counter Configuration (P*)	0x0000_0000
CANFD_TSCV	CANFDn_BA+0x024	R	Timestamp Counter Value (C*)	0x0000_0000
CANFD_TOCC	CANFDn_BA+0x028	R	Timeout Counter Configuration (P*)	0xFFFF_0000
CANFD_TOCV	CANFDn_BA+0x02C	R	Timeout Counter Value (C*)	0x0000_FFFF
CANFD_ECR	CANFDn_BA+0x040	R	Error Counter Register (X*)	0x0000_0000
CANFD_PSR	CANFDn_BA+0x044	R	Protocol Status Register (XS*)	0x0000_0707
CANFD_TDCR	CANFDn_BA+0x048	R	Transmitter Delay Compensation Register (P*)	0x0000_0000
CANFD_IR	CANFDn_BA+0x050	R/W	Interrupt Register	0x0000_0000
CANFD_IE	CANFDn_BA+0x054	R/W	Interrupt Enable	0x0000_0000
CANFD_ILS	CANFDn_BA+0x058	R/W	Interrupt Line Select	0x0000_0000
CANFD_ILE	CANFDn_BA+0x05C	R/W	Interrupt Line Enable	0x0000_0000
CANFD_GFC	CANFDn_BA+0x080	R	Global Filter Configuration (P*)	0x0000_0000
CANFD_SIDFC	CANFDn_BA+0x084	R	Standard ID Filter Configuration (P*)	0x0000_0000
CANFD_XIDFC	CANFDn_BA+0x088	R	Extended ID Filter Configuration (P*)	0x0000_0000
CANFD_XIDAM	CANFDn_BA+0x090	R	Extended ID AND Mask (P*)	0x1FFF_FFFF
CANFD_HPMS	CANFDn_BA+0x094	R	High Priority Message Status	0x0000_0000
CANFD_NDAT1	CANFDn_BA+0x098	R/W	New Data 1	0x0000_0000
CANFD_NDAT2	CANFDn_BA+0x09C	R/W	New Data 2	0x0000_0000
CANFD_RXF0C	CANFDn_BA+0x0A0	R	Rx FIFO 0 Configuration (P*)	0x0000_0000
CANFD_RXF0S	CANFDn_BA+0x0A4	R	Rx FIFO 0 Status	0x0000_0000
CANFD_RXF0A	CANFDn_BA+0x0A8	R/W	Rx FIFO 0 Acknowledge	0x0000_0000

CANFD_RXBC	CANFDn_BA+0x0AC	R	Rx Buffer Configuration (P*)	0x0000_0000
CANFD_RXF1C	CANFDn_BA+0x0B0	R	Rx FIFO 1 Configuration (P*)	0x0000_0000
CANFD_RXF1S	CANFDn_BA+0x0B4	R	Rx FIFO 1 Status	0x0000_0000
CANFD_RXF1A	CANFDn_BA+0x0B8	R/W	Rx FIFO 1 Acknowledge	0x0000_0000
CANFD_RXESC	CANFDn_BA+0x0BC	R	Rx Buffer / FIFO Element Size Configuration (P*)	0x0000_0000
CANFD_TXBC	CANFDn_BA+0x0C0	R	Tx Buffer Configuration (P*)	0x0000_0000
CANFD_TXFQS	CANFDn_BA+0x0C4	R	Tx FIFO/Queue Status	0x0000_0000
CANFD_TXESC	CANFDn_BA+0x0C8	R	Tx Buffer Element Size Configuration (P*)	0x0000_0000
CANFD_TXBRP	CANFDn_BA+0x0CC	R	Tx Buffer Request Pending	0x0000_0000
CANFD_TXBAR	CANFDn_BA+0x0D0	R/W	Tx Buffer Add Request	0x0000_0000
CANFD_TXBCR	CANFDn_BA+0x0D4	R/W	Tx Buffer Cancellation Request	0x0000_0000
CANFD_TXBTO	CANFDn_BA+0x0D8	R	Tx Buffer Transmission Occurred	0x0000_0000
CANFD_TXBCF	CANFDn_BA+0x0DC	R	Tx Buffer Cancellation Finished	0x0000_0000
CANFD_TXBTIE	CANFDn_BA+0x0E0	R/W	Tx Buffer Transmission Interrupt Enable	0x0000_0000
CANFD_TXBCIE	CANFDn_BA+0x0E4	R/W	Tx Buffer Cancellation Finished Interrupt Enable	0x0000_0000
CANFD_TXEFC	CANFDn_BA+0x0F0	R	Tx Event FIFO Configuration (P*)	0x0000_0000
CANFD_TXEFS	CANFDn_BA+0x0F4	R	Tx Event FIFO Status	0x0000_0000
CANFD_TXEFA	CANFDn_BA+0x0F8	R/W	Tx Event FIFO Acknowledge	0x0000_0000

Note:

R = Read, S = Set on read, X = Reset on read, W = Write, P = Protected write, p = Protected set, C = Clear/preset on write, r = release, d = date

6.33.7 Register Description

Data Bit Timing and Prescaler Register (CANFD_DBTP)

Register	Offset	R/W	Description	Reset Value
CANFD_DBTP	CANFDn_BA+0x00C	R	Data Bit Timing and Prescaler Register (P*)	0x0000_0A33

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TDC	Reserved		DBRP				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			DTSEG1				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DTSEG2				DSJW			

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23]	**TDC**	**Transmitter Delay Compensation** *0*=Transmitter Delay Compensation disabled. *1*=Transmitter Delay Compensation enabled.
[22:21]	**Reserved**	Reserved.
[20:16]	**DBRP**	**Data Bit Rate Prescaler** The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Bit Rate Prescaler are 0 to 31. When **TDC** = '1', the range is limited to 0,1. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.
[15:13]	**Reserved**	Reserved.
[12:8]	**DTSEG1**	**Data time segment before sample point** Valid values are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. t_{BS1} = (DTSEG1 + 1) x tq.
[7:4]	**DTSEG2**	**Data time segment after sample point** Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. t_{BS2} = (DTSEG2 + 1) x tq.
[3:0]	**DSJW**	**Data Re-Synchronization Jump Width** Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. t_{SJW} = (DSJW + 1) x tq.

TEST Register (CANFD_TEST)

Register	Offset	R/W	Description	Reset Value
CANFD_TEST	CANFDn_BA+0x010	R	Test Register (P*)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RX	TX		LBCK	Reserved			

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**RX**	**Receive Pin** Monitors the actual value of pin CANx_RXD 0 = The CAN bus is dominant (CANx_RXD = 0). 1 = The CAN bus is recessive (CANx_RXD = 1).
[6:5]	**TX**	**Control of Transmit Pin** 00 = Reset value, CANx_TXD controlled by the CAN Core, updated at the end of the CAN bit time. 01 = Sample Point can be monitored at pin CANx_TXD. 10 = Dominant (‘0’) level at pin CANx_TXD. 11 = Recessive (‘1’) level at pin CANx_TXD.
[4]	**LBCK**	**Loop Back Mode** 0 = Reset value, Loop Back Mode is disabled. 1 = Loop Back Mode is enabled (refer to 6.33.5.1 TEST Mode).
[3:0]	**Reserved**	Reserved.

RAM Watchdog Register (CANFD_RWD)

The RAM watchdog monitors the READY output of the message RAM. A message RAM access starts the message RAM watchdog counter with the value configured by the WDC (CANFD_RWD[7:0]) bits. The counter is reloaded with WDC bits when the message RAM signals successful completion by activating its READY output. In case there is no response from the message RAM until the counter has counted down to zero, the counter stops and interrupt flag WDI (CANFD_IR[26]) bit is set.

Register	Offset	R/W	Description	Reset Value
CANFD_RWD	CANFDn_BA+0x014	R	RAM Watchdog (P*)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
WDV							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WDC							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:8]	**WDV**	**Watchdog Value** Actual Message RAM Watchdog Counter Value.
[7:0]	**WDC**	**Watchdog Configuration** Start value of the Message RAM Watchdog Counter. With the reset value of 00 the counter is disabled.

CC Control Register (CANFD_CCCR)

Register	Offset	R/W	Description	Reset Value
CANFD_CCCR	CANFDn_BA+0x018	R/W	CC Control Register (Pp*)	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
NISO	TXP	EFBI	PXHD	Reserved		BRSE	FDOE
7	6	5	4	3	2	1	0
TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15]	**NISO**	**Non ISO Operation** If this bit is set, the CAN FD controller uses the CAN FD frame format as specified by the Bosch CAN FD Specification V1.0. 0 = CAN FD frame format according to ISO 11898-1:2015. 1 = CAN FD frame format according to Bosch CAN FD Specification V1.0.
[14]	**TXP**	**Transmit Pause** If this bit is set, the CAN FD controller pauses for two CAN bit times before starting the next transmission after itself has successfully transmitted a frame (refer to 6.33.5.5). 0 = Transmit pause disabled. 1 = Transmit pause enabled.
[13]	**EFBI**	**Edge Filtering during Bus Integration** 0 = Edge filtering disabled. 1 = Two consecutive dominant tq required to detect an edge f or hard synchronization.
[12]	**PXHD**	**Protocol Exception Handling Disable** 0 = Protocol exception handling enabled. 1 = Protocol exception handling disabled. **Note:** When protocol exception handling is disabled, the controller will transmit an error frame when it detects a protocol exception condition.
[11:10]	**Reserved**	Reserved.
[9]	**BRSE**	**Bit Rate Switch Enable** 0 = Bit rate switching for transmissions disabled. 1 = Bit rate switching for transmissions enabled. **Note:** When CAN FD operation is disabled FDOE = 0, BRSE is not evaluated.
[8]	**FDOE**	**FD Operation Enable** 0 = FD operation disabled. 1 = FD operation enabled.

[7]	**TEST**	**Test Mode Enable** 0 = Normal operation, register TEST holds reset values. 1 = Test Mode, write access to register TEST enabled.
[6]	**DAR**	**Disable Automatic Retransmission** 0 = Automatic retransmission of messages not transmitted successfully enabled. 1 = Automatic retransmission disabled.
[5]	**MON**	**Bus Monitoring Mode** Bit MON can only be set by the Host when both CCE and INIT are set to 1. The bit can be reset by the Host at any time. 0 = Bus Monitoring Mode is disabled. 1 = Bus Monitoring Mode is enabled.
[4]	**CSR**	**Clock Stop Request** 0 = No clock stop is requested. 1 = Clock stop requested. When clock stop is requested, first INIT and then CSA will be set after all pending transfer requests have been completed and the CAN bus reached idle.
[3]	**CSA**	**Clock Stop Acknowledge** 0 = No clock stop acknowledged. 1 = The Controller may be set in power down by stopping AHB clock and CAN Core clock.
[2]	**ASM**	**Restricted Operation Mode** Bit ASM can only be set by the Host when both CCE and INIT are set to 1. The bit can be reset by the software at any time. This bit will be set automatically set to 1 when the Tx handler was not able to read data from the message RAM in time. For a description of the Restricted Operation Mode refer to **Restricted Operation Mode**. 0 = Normal CAN operation. 1 = Restricted Operation Mode active.
[1]	**CCE**	**Configuration Change Enable** 0 = The CPU has no write access to the protected configuration registers. 1 = The CPU has write access to the protected configuration registers (while INIT (CANFD_CCCR[0]) = 1).
[0]	**INIT**	**Initialization** 0 = Normal Operation. 1 = Initialization is started. **Note:** Due to the synchronization mechanism between the two clock domains, there may be a delay until the value written to INIT can be read back. Therefore the programmer has to assure that the previous value written to INIT has been accepted by reading INIT before setting INIT to a new value.

Nominal Bit Timing and Prescaler Register (CANFD_NBTP)

Register	Offset	R/W	Description	Reset Value
CANFD_NBTP	CANFDn_BA+0x01C	R	Nominal Bit Timing and Prescaler Register (P*)	0x0600_0A03

31	30	29	28	27	26	25	24
NSJW							NBRP
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
NBRP							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
NTSEG1							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	NTSEG2						

Bits	Description	
[31:25]	**NSJW**	**Nominal Re-Synchronization Jump Width** Valid values are 0 to 127. Should be smaller than NTSEG2. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used. t_{SJW} = (NSJW + 1) x tq.
[24:16]	**NBRP**	**Nominal Bit Rate Prescaler** The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Bit Rate Prescaler are 0 to 511. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used
[15:8]	**NTSEG1**	**Nominal Time segment before sample point** Valid values are 1 to 255. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. t_{BS1} = (NTSEG1 + 1) x tq.
[7]	**Reserved**	Reserved.
[6:0]	**NTSEG2**	**Nominal Time segment after sample point** 0x01-0x7F Valid values are 1 to 127. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. t_{BS2} = (NTSEG2 + 1) x tq. **Note:** With a CAN Core clock (cclk) of 8 MHz, the reset value of 0x06000A03 configures the controller for a bit rate of 500 kBit/s.

Timestamp Counter Configuration Register (CANFD_TSCC)

Register	Offset	R/W	Description	Reset Value
CANFD_TSCC	CANFDn_BA+0x020	R	Timestamp Counter Configuration (P*)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				TCP			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						TSS	

Bits	Description	
[31:20]	**Reserved**	Reserved.
[19:16]	**TCP**	**Timestamp Counter Prescaler** Configures the timestamp and timeout counters time unit in multiples of CAN bit times [1…16]. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.
[15:2]	**Reserved**	Reserved.
[1:0]	**TSS**	**Timestamp Select** 00 = Timestamp counter value always 0x0000. 01 = Timestamp counter value incremented according to TCP. 10 = Reserved 11 = Same as “00”.

Timestamp Counter Register (CANFD_TSCV)

Register	Offset	R/W	Description	Reset Value
CANFD_TSCV	CANFDn_BA+0x024	R	Timestamp Counter Value (C*)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TSC							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TSC							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**TSC**	**Timestamp Counter** The internal Timestamp Counter value is captured on start of frame (both Rx and Tx). When TSS (CANFD_TSCC[[1:0]) = 2'b01, the Timestamp Counter is incremented in multiples of CAN bit times [1...16] depending on the configuration of TCP (CANFD_TSCC[19:16]). A wrap around sets interrupt flag TSW (CANFD_IR[16]). Write access resets the counter to zero. **Note:** A "wrap around" is a change of the Timestamp Counter value from non-zero to zero not caused by write access to CANFD_TSCV.

Timeout Counter Configuration Register (CANFD_TOCC)

Register	Offset	R/W	Description	Reset Value
CANFD_TOCC	CANFDn_BA+0x028	R	Timeout Counter Configuration (P*)	0xFFFF_0000

31	30	29	28	27	26	25	24
TOP							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TOP							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					TOS		ETOC

Bits	Description	
[31:16]	**TOP**	**Timeout Period** Start value of the Timeout Counter (down-counter). Configures the Timeout Period.
[15:3]	**Reserved**	Reserved.
[2:1]	**TOS**	**Timeout Select** When operating in Continuous mode, a write to CANFD_TOCV presets the counter to the value configured by TOP (CANFD_TOCC[31:16]) and continues down-counting. When the Timeout Counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by TOP (CANFD_TOCC[31:16]). Down-counting is started when the first FIFO element is stored. 00 = Continuous operation. 01 = Timeout controlled by Tx Event FIFO. 10 = Timeout controlled by Rx FIFO 0. 11 = Timeout controlled by Rx FIFO 1.
[0]	**ETOC**	**Enable Timeout Counter** 0 = Timeout Counter disabled. 1 = Timeout Counter enabled. **Note:** For use of timeout function with CAN FD refer to 6.33.5.3.

Timeout Counter Value Register (CANFD_TOCV)

Register	Offset	R/W	Description	Reset Value
CANFD_TOCV	CANFDn_BA+0x02C	R	Timeout Counter Value (C*)	0x0000_FFFF

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TOC							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TOC							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**TOC**	**Timeout Counter** The filed is decremented in multiples of CAN bit times [1...16] depending on the configuration of TCP (CANFD_TSCC[19:16]). When decremented to zero, interrupt flag TOO (CANFD_IR[18]) is set and the timeout counter is stopped. Start and reset/restart conditions are configured via TOS (CANFD_TOCC[1:0]).

Error Counter Register (CANFD_ECR)

Register	Offset	R/W	Description	Reset Value
CANFD_ECR	CANFDn_BA+0x040	R	Error Counter Register (X*)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CEL							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RP	REC						
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TEC							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**CEL**	**CAN Error Logging** The counter is incremented each time when a CAN protocol error causes the 8-bit Transmit Error Counter TEC or the 7-bit Receive Error Counter REC to be incremented. The counter is also incremented when the Bus_Off limit is reached. It is not incremented when only RP is set without changing REC. The increment of CEL follows after the increment of REC or TEC. The counter is reset by read access to CEL. The counter stops at 0xFF; the next increment of TEC or REC sets interrupt flag ELO (CANFD_IR[22]).
[15]	**RP**	**Receive Error Passive** 0 = The Receive Error Counter is below the error passive level of 128. 1 = The Receive Error Counter has reached the error passive level of 128.
[14:8]	**REC**	**Receive Error Counter** Actual state of the Receive Error Counter, values between 0 and 127.
[7:0]	**TEC**	**Transmit Error Counter** Actual state of the Transmit Error Counter, values between 0 and 255. **Note:** When ASM (CANFD_CCCR[2]) is set, the CAN protocol controller does not increment TEC and REC when a CAN protocol error is detected, but CEL is still incremented.

Protocol Status Register (CANFD_PSR)

Register	Offset	R/W	Description	Reset Value
CANFD_PSR	CANFDn_BA+0x044	R	Protocol Status Register (XS*)	0x0000_0707

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved	TDCV						
15	14	13	12	11	10	9	8
Reserved	PXE	RFDF	RBRS	RESI	DLEC		
7	6	5	4	3	2	1	0
BO	EW	EP	ACT		LEC		

Bits	Description	
[31:23]	**Reserved**	Reserved.
[22:16]	**TDCV**	**Transmitter Delay Compensation Value** Position of the secondary sample point, defined by the sum of the measured delay from CANx_TXD to CANx_RXD and TDCO (CANFD_TDCR[[14:8]). The SSP position is, in the data phase, the number of minimum time quata (mtq) between the start of the transmitted bit and the secondary sample point. Valid values are 0 to 127 mtq.
[15]	**Reserved**	Reserved.
[14]	**PXE**	**Protocol Exception Event** 0 = No protocol exception event occurred since last read access. 1 = Protocol exception event occurred.
[13]	**RFDF**	**Received a CAN FD Message** This bit is set independent of acceptance filtering. 0 = Since this bit was reset by the CPU, no CAN FD message has been received. 1 = Message in CAN FD format with FDF flag set has been received. **Note:** Byte access: Reading byte 0 will reset RFDF, reading bytes 3/2/1 has no impact.
[12]	**RBRS**	**BRS flag of last received CAN FD Message** This bit is set together with RFDF, independent of acceptance filtering. 0 = Last received CAN FD message did not have its BRS flag set. 1 = Last received CAN FD message had its BRS flag set. **Note:** Byte access: Reading byte 0 will reset RBRS, reading bytes 3/2/1 has no impact.
[11]	**RESI**	**ESI flag of last received CAN FD Message** This bit is set together with RFDF, independent of acceptance filtering. 0 = Last received CAN FD message did not have its ESI flag set. 1 = Last received CAN FD message had its ESI flag set.
[10:8]	**DLEC**	**Data Phase Last Error Code** Type of last error that occurred in the data phase of a CAN FD format frame with its BRS flag set. Coding is the same as for LEC. This field will be cleared to zero when a CAN FD format frame with its BRS flag set has been transferred (reception or transmission) without error.

[7]	**BO**	**Bus_Off Status** 0 = The CAN FD controller is not Bus_Off. 1 = The CAN FD controller is in Bus_Off state.
[6]	**EW**	**Warning Status** 0 = Both error counters are below the Error_Warning limit of 96. 1 = At least one of error counter has reached the Error_Warning limit of 96.
[5]	**EP**	**Error Passive** 0 = The CAN FD controller is in the Error_Active state. It normally takes part in bus communication and sends an active error flag when an error has been detected. 1 = The CAN FD controller is in the Error_Passive state.
[4:3]	**ACT**	**Activity** Monitors the module's CAN communication state. 00 = Synchronizing - node is synchronizing on CAN communication. 01 = Idle - node is neither receiver nor transmitter. 10 = Receiver - node is operating as receiver. 11 = Transmitter - node is operating as transmitter.
[2:0]	**LEC**	**Last Error Code** The LEC indicates the type of the last error to occur on the CAN bus. This field will be cleared to 0 when a message has been transferred (reception or transmission) without error. 000 = No Error: No error occurred since LEC has been reset by successful reception or transmission. 001 = Stuff Error: More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed. 010 = Form Error: A fixed format part of a received frame has the wrong format. 011 = AckError: The message transmitted by the CANFD CONTROLLER was not acknowledged by another node. 100 = Bit1Error: During the transmission of a message (with the exception of the arbitration field), the device wanted to send a recessive level (bit of logical value 1), but the monitored bus value was dominant. 101 = Bit0Error : During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), the device wanted to send a dominant level (data or identifier bit logical value 0), but the monitored bus value was recessive. During Bus_Off recovery this status is set each time a sequence of 11 recessive bits has been monitored. This enables the CPU to monitor the proceeding of the Bus_Off recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed). 110 = CRCError: The CRC check sum of a received message was incorrect. The CRC of an incoming message does not match with the CRC calculated from the received data. 111 = NoChange: Any read access to the Protocol Status Register re-initializes the LEC to 7.When the LEC shows the value 7, no CAN bus event was detected since the last CPU read access to the Protocol Status Register.

Note: When a frame in CAN FD format has reached the data phase with BRS flag set, the next CAN event (error or valid frame) will be shown in DLEC instead of LEC. An error in a fixed stuff bit of a CAN FD CRC sequence will be shown as a Form Error, not Stuff Error.

Note: The Bus_Off recovery sequence (refer to ISO 11898-1:2015) cannot be shortened by setting or resetting INIT (CANFD_CCCR[0]). If the device goes Bus_Off, it will set INIT (CANFD_CCCR[0]) of its own accord, stopping all bus activities. Once INIT (CANFD_CCCR[0]) has been cleared by the CPU, the device will then wait for 129 occurrences of Bus Idle (129 * 11 consecutive recessive bits) before resuming normal operation. At the end of the Bus_Off recovery sequence, the Error Management Counters will be reset. During the waiting time after the resetting of INIT (CANFD_CCCR[0]), each time a sequence of 11 recessive bits has been monitored, a Bit0Error code is written to LEC (CANFD_PSR[2:0]), enabling the CPU to readily check up whether the CAN bus is stuck at dominant or continuously disturbed and to monitor the Bus_Off recovery sequence. REC(CANFD_ECR[14:8]) is used to count these sequences.

Transmitter Delay Compensation Register (CANFD_TDCR)

Register	Offset	R/W	Description	Reset Value
CANFD_TDCR	CANFDn_BA+0x048	R	Transmitter Delay Compensation Register (P*)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**TDCO**						
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**TDCF**						

Bits	Description	
[31:15]	**Reserved**	Reserved.
[14:8]	**TDCO**	**Transmitter Delay Compensation SSP Offset** Offset value defining the distance between the measured delay from CANx_TXD to CANx_RXD and the secondary sample point. Valid values are 0 to 127 mtq.
[7]	**Reserved**	Reserved.
[6:0]	**TDCF**	**Transmitter Delay Compensation Filter Window Length** Defines the minimum value for the SSP position, dominant edges on CANx_RXD that would result in an earlier SSP position are ignored for transmitter delay measurement. The feature is enabled when TDCF is configured to a value greater than TDCO. Valid values are 0 to 127 mtq.

Interrupt Register (CANFD_IR)

The flags are set when one of the listed conditions is detected (edge-sensitive). A flag is cleared by writing a 1 to the corresponding bit position. Writing a 0 has no effect. A hard reset will clear the register.

Register	Offset	R/W	Description	Reset Value
CANFD_IR	CANFDn_BA+0x050	R/W	Interrupt Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		ARA	PED	PEA	WDI	BO	EW
23	22	21	20	19	18	17	16
EP	ELO	Reserved		DRX	TOO	MRAF	TSW
15	14	13	12	11	10	9	8
TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM
7	6	5	4	3	2	1	0
RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29]	**ARA**	**Access to Reserved Address** 0 = No access to reserved address occurred. 1 = Access to reserved address occurred.
[28]	**PED**	**Protocol Error in Data Phase** 0 = No protocol error in data phase. 1= Protocol error in data phase detected (DLEC (CANFD_PSR[10:8]) no equal 0 or 7). **Note:** Data bit time is used
[27]	**PEA**	**Protocol Error in Arbitration Phase** 0 = No protocol error in arbitration phase. 1 = Protocol error in arbitration phase detected (LEC (CANFD_PSR[2:0]) no equal 0 or 7). **Note:** Nominal bit time is used
[26]	**WDI**	**Watchdog Interrupt** 0 = No Message RAM Watchdog event occurred. 1 = Message RAM Watchdog event due to missing READY.
[25]	**BO**	**Bus_Off Status** 0 = Bus_Off status unchanged. 1 = Bus_Off status changed.
[24]	**EW**	**Warning Status** 0 = Error_Warning status unchanged. 1 = Error_Warning status changed.
[23]	**EP**	**Error Passive**

		0 = Error_Passive status unchanged. 1 = Error_Passive status changed.
[22]	**ELO**	**Error Logging Overflow** 0 = CAN Error Logging Counter did not overflow. 1 = Overflow of CAN Error Logging Counter occurred.
[21:20]	**Reserved**	Reserved.
[19]	**DRX**	**Message stored to Dedicated Rx Buffer** The flag is set whenever a received message has been stored into a dedicated Rx Buffer. 0 = No Rx Buffer updated. 1 = At least one received message stored into an Rx Buffer.
[18]	**TOO**	**Timeout Occurred** 0 = No timeout. 1 = Timeout reached.
[17]	**MRAF**	**Message RAM Access Failure** The flag is set, when the Rx Handler • Has not completed acceptance filtering or storage of an accepted message until the arbitration field of the following message has been received. In this case acceptance filtering or message storage is aborted and the Rx Handler starts processing of the following message. • Was not able to write a message to the Message RAM. In this case message storage is aborted. In both cases the FIFO put index is not updated resp. the New Data flag for a dedicated Rx Buffer is not set, a partly stored message is overwritten when the next message is stored to this location. The flag is also set when the Tx Handler was not able to read a message from the Message RAM in time. In this case message transmission is aborted. In case of a Tx Handler access failure the CAN FD controller is switched into Restricted Operation Mode (refer to **Restricted Operation Mode**). To leave Restricted Operation Mode, the Host CPU has to reset ASM (CANFD_CCCR[2]). 0 = No Message RAM access failure occurred. 1 = Message RAM access failure occurred.
[16]	**TSW**	**Timestamp Wraparound** 0 = No timestamp counter wrap-around. 1 = Timestamp counter wrapped around.
[15]	**TEFL**	**Tx Event FIFO Element Lost** 0 = No Tx Event FIFO element lost. 1 = Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero.
[14]	**TEFF**	**Tx Event FIFO Full** 0 = Tx Event FIFO is not full. 1 = Tx Event FIFO is full.
[13]	**TEFW**	**Tx Event FIFO Watermark Reached** 0 = Tx Event FIFO fill level below watermark. 1 = Tx Event FIFO fill level reached watermark.
[12]	**TEFN**	**Tx Event FIFO New Entry** 0 = Tx Event FIFO unchanged. 1 = Tx Handler wrote Tx Event FIFO element.
[11]	**TFE**	**Tx FIFO Empty** 0 = Tx FIFO non-empty. 1 = Tx FIFO empty.
[10]	**TCF**	**Transmission Cancellation Finished**

		0 = No transmission cancellation finished. 1 = Transmission cancellation finished.
[9]	**TC**	**Transmission Completed** 0 = No transmission completed. 1 = Transmission completed.
[8]	**HPM**	**High Priority Message** 0 = No high priority message received. 1 = High priority message received.
[7]	**RF1L**	**Rx FIFO 1 Message Lost** 0 = No Rx FIFO 1 message lost. 1 = Rx FIFO 1 message lost, also set after write attempt to Rx FIFO 1 of size zero.
[6]	**RF1F**	**Rx FIFO 1 Full** 0 = Rx FIFO 1 is not full. 1 = Rx FIFO 1 is full.
[5]	**RF1W**	**Rx FIFO 1 Watermark Reached** 0 = Rx FIFO 1 fill level below watermark. 1 = Rx FIFO 1 fill level reached watermark.
[4]	**RF1N**	**Rx FIFO 1 New Message** 0 = No new message written to Rx FIFO 1. 1 = New message written to Rx FIFO 1.
[3]	**RF0L**	**Rx FIFO 0 Message Lost** 0 = No Rx FIFO 0 message lost. 1 = Rx FIFO 0 message lost, also set after write attempt to Rx FIFO 0 of size zero.
[2]	**RF0F**	**Rx FIFO 0 Full** 0 = Rx FIFO 0 is not full. 1 = Rx FIFO 0 is full.
[1]	**RF0W**	**Rx FIFO 0 Watermark Reached** 0 = Rx FIFO 0 fill level below watermark. 1 = Rx FIFO 0 fill level reached watermark.
[0]	**RF0N**	**Rx FIFO 0 New Message** 0 = No new message written to Rx FIFO 0. 1 = New message written to Rx FIFO 0.

Interrupt Enable Register (CANFD_IE)

Register	Offset	R/W	Description	Reset Value
CANFD_IE	CANFDn_BA+0x054	R/W	Interrupt Enable	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		ARAE	PEDE	PEAE	WDIE	BOE	EWE
23	22	21	20	19	18	17	16
EPE	ELOE	BEUE	BECE	DRXE	TOOE	MRAFE	TSWE
15	14	13	12	11	10	9	8
TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME
7	6	5	4	3	2	1	0
RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29]	**ARAE**	**Access to Reserved Address Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.
[28]	**PEDE**	**Protocol Error in Data Phase Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.
[27]	**PEAE**	**Protocol Error in Arbitration Phase Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.
[26]	**WDIE**	**Watchdog Interrupt Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.
[25]	**BOE**	**Bus_Off Status Interrupt Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.
[24]	**EWE**	**Warning Status Interrupt Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.
[23]	**EPE**	**Error Passive Interrupt Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.
[22]	**ELOE**	**Error Logging Overflow Interrupt Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.
[21]	**BEUE**	**Bit Error Uncorrected Interrupt Enable**

		0 = Interrupt is Disabled. 1 = Interrupt is Enabled.
[20]	**BECE**	**Bit Error Corrected Interrupt Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.
[19]	**DRXE**	**Message stored to Dedicated Rx Buffer Interrupt Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.
[18]	**TOOE**	**Timeout Occurred Interrupt Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.
[17]	**MRAFE**	**Message RAM Access Failure Interrupt Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.
[16]	**TSWE**	**Timestamp Wraparound Interrupt Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.
[15]	**TEFLE**	**Tx Event FIFO Event Lost Interrupt Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.
[14]	**TEFFE**	**Tx Event FIFO Full Interrupt Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.
[13]	**TEFWE**	**Tx Event FIFO Watermark Reached Interrupt Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.
[12]	**TEFNE**	**Tx Event FIFO New Entry Interrupt Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.
[11]	**TFEE**	**Tx FIFO Empty Interrupt Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.
[10]	**TCFE**	**Transmission Cancellation Finished Interrupt Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.
[9]	**TCE**	**Transmission Completed Interrupt Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.
[8]	**HPME**	**High Priority Message Interrupt Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.
[7]	**RF1LE**	**Rx FIFO 1 Message Lost Interrupt Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.

[6]	**RF1FE**	**Rx FIFO 1 Full Interrupt Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.
[5]	**RF1WE**	**Rx FIFO 1 Watermark Reached Interrupt Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.
[4]	**RF1NE**	**Rx FIFO 1 New Message Interrupt Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.
[3]	**RF0LE**	**Rx FIFO 0 Message Lost Interrupt Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.
[2]	**RF0FE**	**Rx FIFO 0 Full Interrupt Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.
[1]	**RF0WE**	**Rx FIFO 0 Watermark Reached Interrupt Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.
[0]	**RF0NE**	**Rx FIFO 0 New Message Interrupt Enable** 0 = Interrupt is Disabled. 1 = Interrupt is Enabled.

Interrupt Line Select Register (CANFD_ILS)

This register assigns an interrupt generated by a specific interrupt flag from the CANFD_IR register to one of the two module interrupt lines.

Register	Offset	R/W	Description	Reset Value
CANFD_ILS	CANFDn_BA+0x058	R/W	Interrupt Line Select	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		ARAL	PEDL	PEAL	WDIL	BOL	EWL
23	22	21	20	19	18	17	16
EPL	ELOL	Reserved		DRXL	TOOL	MRAFL	TSWL
15	14	13	12	11	10	9	8
TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML
7	6	5	4	3	2	1	0
RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29]	**ARAL**	**Access to Reserved Address Line**
[28]	**PEDL**	**Protocol Error in Data Phase Line**
[27]	**PEAL**	**Protocol Error in Arbitration Phase Line**
[26]	**WDIL**	**Watchdog Interrupt Line**
[25]	**BOL**	**Bus_Off Status Interrupt Line**
[24]	**EWL**	**Warning Status Interrupt Line**
[23]	**EPL**	**Error Passive Interrupt Line**
[22]	**ELOL**	**Error Logging Overflow Interrupt Line**
[21:20]	**Reserved**	Reserved.
[19]	**DRXL**	**Message stored to Dedicated Rx Buffer Interrupt Line**
[18]	**TOOL**	**Timeout Occurred Interrupt Line**
[17]	**MRAFL**	**Message RAM Access Failure Interrupt Line**
[16]	**TSWL**	**Timestamp Wraparound Interrupt Line**
[15]	**TEFLL**	**Tx Event FIFO Event Lost Interrupt Line**
[14]	**TEFFL**	**Tx Event FIFO Full Interrupt Line**
[13]	**TEFWL**	**Tx Event FIFO Watermark Reached Interrupt Line**
[12]	**TEFNL**	**Tx Event FIFO New Entry Interrupt Line**
[11]	**TFEL**	**Tx FIFO Empty Interrupt Line**

[10]	**TCFL**	**Transmission Cancellation Finished Interrupt Line**
[9]	**TCL**	**Transmission Completed Interrupt Line**
[8]	**HPML**	**High Priority Message Interrupt Line**
[7]	**RF1LL**	**Rx FIFO 1 Message Lost Interrupt Line**
[6]	**RF1FL**	**Rx FIFO 1 Full Interrupt Line**
[5]	**RF1WL**	**Rx FIFO 1 Watermark Reached Interrupt Line**
[4]	**RF1NL**	**Rx FIFO 1 New Message Interrupt Line**
[3]	**RF0LL**	**Rx FIFO 0 Message Lost Interrupt Line**
[2]	**RF0FL**	**Rx FIFO 0 Full Interrupt Line**
[1]	**RF0WL**	**Rx FIFO 0 Watermark Reached Interrupt Line**
[0]	**RF0NL**	**Rx FIFO 0 New Message Interrupt Line** 0 = Interrupt assigned to CAN interrupt line 0. 1 = Interrupt assigned to CAN interrupt line 1.

Interrupt Line Register (CANFD_ILE)

Register	Offset	R/W	Description	Reset Value
CANFD_ILE	CANFDn_BA+0x05C	R/W	Interrupt Line Enable	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						EINT1	EINT0

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**EINT1**	**Enable Interrupt Line 1** 0 = Interrupt line m_can_int1 disabled. 1 = Interrupt line m_can_int1 enabled.
[0]	**EINT0**	**Enable Interrupt Line 0** 0 = Interrupt line m_can_int0 disabled. 1 = Interrupt line m_can_int0 enabled.

Global Filter Configuration Register (CANFD_GFC)

Register	Offset	R/W	Description	Reset Value
CANFD_GFC	CANFDn_BA+0x080	R	Global Filter Configuration (P*)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**ANFS**		**ANFE**		**RRFS**	**RRFE**

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5:4]	**ANFS**	**Accept Non-matching Frames Standard** Defines how received messages with 11-bit IDs that do not match any element of the filter list are treated. 00 = Accept in Rx FIFO 0. 01 = Accept in Rx FIFO 1. 10 = Reject. 11 = Reject.
[3:2]	**ANFE**	**Accept Non-matching Frames Extended** Defines how received messages with 29-bit IDs that do not match any element of the filter list are treated. 00 = Accept in Rx FIFO 0. 01 = Accept in Rx FIFO 1. 10 = Reject. 11 = Reject.
[1]	**RRFS**	**Reject Remote Frames Standard** 0= Filter remote frames with 11-bit standard IDs. 1= Reject all remote frames with 11-bit standard IDs.
[0]	**RRFE**	**Reject Remote Frames Extended** 0= Filter remote frames with 29-bit extended IDs. 1= Reject all remote frames with 29-bit extended IDs.

Standard ID Filter Configuration Register (CANFD_SIDFC)

Register	Offset	R/W	Description	Reset Value
CANFD_SIDFC	CANFDn_BA+0x084	R	Standard ID Filter Configuration (P*)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
LSS							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
FLSSA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FLSSA						**Reserved**	

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**LSS**	**List Size Standard** 0= No standard Message ID filter. 1-128 = Number of standard Message ID filter elements. >128= Values greater than 128 are interpreted as 128.
[15:2]	**FLSSA**	**Filter List Standard Start Address** Start address of standard Message ID filter list (32-bit word address, refer to Figure 6.33-11).
[1:0]	**Reserved**	Reserved.

Extended ID Filter Configuration Register (CANFD_XIDFC)

Register	Offset	R/W	Description	Reset Value
CANFD_XIDFC	CANFDn_BA+0x088	R	Extended ID Filter Configuration (P*)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved	**LSE**						
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
FLESA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FLESA						**Reserved**	

Bits	Description	
[31:23]	**Reserved**	Reserved.
[22:16]	**LSE**	**List Size Extended** 0= No extended Message ID filter. 1-64= Number of extended Message ID filter elements. >64= Values greater than 64 are interpreted as 64.
[15:2]	**FLESA**	**Filter List Extended Start Address** Start address of extended Message ID filter list (32-bit word address, refer to Figure 6.33-11)
[1:0]	**Reserved**	Reserved.

Extended ID and Mask Register (CANFD_XIDAM)

Register	Offset	R/W	Description	Reset Value
CANFD_XIDAM	CANFDn_BA+0x090	R	Extended ID AND Mask (P*)	0x1FFF_FFFF

31	30	29	28	27	26	25	24
Reserved			EIDM				
23	22	21	20	19	18	17	16
EIDM							
15	14	13	12	11	10	9	8
EIDM							
7	6	5	4	3	2	1	0
EIDM							

Bits	Description	
[31:29]	**Reserved**	Reserved.
[28:0]	**EIDM**	**Extended ID Mask** For acceptance filtering of extended frames the Extended ID AND Mask is ANDed with the Message ID of a received frame. Intended for masking of 29-bit IDs in SAE J1939. With the reset value of all bits set to one the mask is not active.

High Priority Message Status Register (CANFD_HPMS)

Register	Offset	R/W	Description	Reset Value
CANFD_HPMS	CANFDn_BA+0x094	R	High Priority Message Status	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
FLST	FIDX						
7	6	5	4	3	2	1	0
MSI		BIDX					

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15]	**FLST**	**Filter List** Indicates the filter list of the matching filter element. 0 = Standard Filter List. 1 = Extended Filter List.
[14:8]	**FIDX**	**Filter Index** Index of matching filter element. Range is 0 to LSS (CANFD_SIDFC[23:16]) - 1 or LSE (CANFD_XIDFC[22:16]) – 1
[7:6]	**MSI**	**Message Storage Indicator** 00 = No FIFO selected. 01 = FIFO message lost. 10 = Message stored in FIFO 0. 11 = Message stored in FIFO 1.
[5:0]	**BIDX**	**Buffer Index** Index of Rx FIFO element to which the message was stored. Only valid when MSI[1] = 1.

New Data 1 Register (CANFD_NDAT1)

Register	Offset	R/W	Description	Reset Value
CANFD_NDAT1	CANFDn_BA+0x098	R/W	New Data 1	0x0000_0000

31	30	29	28	27	26	25	24
NDn							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
NDn							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
NDn							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NDn							

Bits	Description	
[31:0] n=0,1..31	**NDn**	**New Data** The register holds the New Data flags of Rx Buffers 0 to 31. The flags are set when the respective Rx Buffer has been updated from a received frame. The flags remain set until the Host clears them. A flag is cleared by writing a 1 to the corresponding bit position. Writing a 0 has no effect. A hard reset will clear the register. 0 = Rx Buffer not updated. 1 = Rx Buffer updated from new message.

New Data 2 Register (CANFD_NDAT2)

Register	Offset	R/W	Description	Reset Value
CANFD_NDAT2	CANFDn_BA+0x09C	R/W	New Data 2	0x0000_0000

31	30	29	28	27	26	25	24
NDn							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
NDn							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
NDn							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NDn							

Bits	Description	
[31:0] n=32,33..63	**NDn**	**New Data** The register holds the New Data flags of Rx Buffers 32 to 63. The flags are set when the respective Rx Buffer has been updated from a received frame. The flags remain set until the Host clears them. A flag is cleared by writing a 1 to the corresponding bit position. Writing a 0 has no effect. A hard reset will clear the register. 0 = Rx Buffer not updated. 1 = Rx Buffer updated from new message.

Rx FIFO 0 Configuration Register (CANFD_RXF0C)

Register	Offset	R/W	Description	Reset Value
CANFD_RXF0C	CANFDn_BA+0x0A0	R	Rx FIFO 0 Configuration (P*)	0x0000_0000

31	30	29	28	27	26	25	24
F0OM	**F0WM**						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved	**F0S**						
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
F0SA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
F0SA						**Reserved**	

Bits	Description	
[31]	**F0OM**	**FIFO 0 Operation Mode** FIFO 0 can be operated in blocking or in overwrite mode (refer to **Rx FIFOs**). 0 = FIFO 0 blocking mode. 1 = FIFO 0 overwrite mode.
[30:24]	**F0WM**	**Rx FIFO 0 Watermark** 0= Watermark interrupt disabled. 1-64 = Level for Rx FIFO 0 watermark interrupt (RF0W(CANFD_IR[1]). >64 = Watermark interrupt disabled.
[23]	**Reserved**	Reserved.
[22:16]	**F0S**	**Rx FIFO 0 Size** 0= No Rx FIFO 0. 1-64= Number of Rx FIFO 0 elements. >64= Values greater than 64 are interpreted as 64. The Rx FIFO 0 elements are indexed from 0 to F0S-1
[15:2]	**F0SA**	**Rx FIFO 0 Start Address** Start address of Rx FIFO 0 in Message RAM (32-bit word address).
[1:0]	**Reserved**	Reserved.

Rx FIFO 0 Status Register (CANFD_RXF0S)

Register	Offset	R/W	Description	Reset Value
CANFD_RXF0S	CANFDn_BA+0x0A4	R	Rx FIFO 0 Status	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						RF0L	F0F
23	22	21	20	19	18	17	16
Reserved		F0PI					
15	14	13	12	11	10	9	8
Reserved		F0GI					
7	6	5	4	3	2	1	0
Reserved	F0FL						

Bits	Description	
[31:26]	**Reserved**	Reserved.
[25]	**RF0L**	**Rx FIFO 0 Message Lost** This bit is a copy of interrupt flagRF0L (CANFD_IR[3]). When CANFD_IR.RF0L is reset, this bit is also reset. 0 = No Rx FIFO 0 message lost. 1 = Rx FIFO 0 message lost, also set after write attempt to Rx FIFO 0 of size zero. **Note:** Overwriting the oldest message when F0OM (CANFD_RXF0C[31]) = 1 will not set this flag.
[24]	**F0F**	**Rx FIFO 0 Full** 0= Rx FIFO 0 is not full. 1= Rx FIFO 0 is full.
[23:22]	**Reserved**	Reserved.
[21:16]	**F0PI**	**Rx FIFO 0 Put Index** Rx FIFO 0 write index pointer, range 0 to 63.
[15:14]	**Reserved**	Reserved.
[13:8]	**F0GI**	**Rx FIFO 0 Get Index** Rx FIFO 0 read index pointer, range 0 to 63.
[7]	**Reserved**	Reserved.
[6:0]	**F0FL**	**Rx FIFO 0 Fill Level** Number of elements stored in Rx FIFO 0, range 0 to 64

Rx FIFO 0 Acknowledge Register (CANFD_RXF0A)

Register	Offset	R/W	Description	Reset Value
CANFD_RXF0A	CANFDn_BA+0x0A8	R/W	Rx FIFO 0 Acknowledge	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		F0A					

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5:0]	**F0A**	**Rx FIFO 0 Acknowledge Index** After the Host has read a message or a sequence of messages from Rx FIFO 0 it has to write the buffer index of the last element read from Rx FIFO 0 to F0AI. This will set the Rx FIFO 0 Get Index F0GI (CANFD_RXF0S[13:8]) to F0AI (CANFD_RXF0A[5:0]) + 1 and update the FIFO 0 Fill Level F0FL(CANFD_RXF0S[6:0]).

Rx Buffer Configuration Register (CANFD_RXBC)

Register	Offset	R/W	Description	Reset Value
CANFD_RXBC	CANFDn_BA+0x0AC	R	Rx Buffer Configuration (P*)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RBSA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RBSA						Reserved	

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:2]	**RBSA**	**Rx Buffer Start Address** Configures the start address of the Rx Buffers section in the Message RAM (32-bit word address).
[1:0]	**Reserved**	Reserved.

Rx FIFO 1 Configuration Register (CANFD_RXF1C)

Register	Offset	R/W	Description	Reset Value
CANFD_RXF1C	CANFDn_BA+0x0B0	R	Rx FIFO 1 Configuration (P*)	0x0000_0000

31	30	29	28	27	26	25	24
F1OM	**F1WM**						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved	**F1S**						
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
F1SA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
F1SA						**Reserved**	

Bits	Description	
[31]	**F1OM**	**FIFO 1 Operation Mode** FIFO 1 can be operated in blocking or in overwrite mode (refer to **Rx FIFOs**). 0= FIFO 1 blocking mode. 1= FIFO 1 overwrite mode.
[30:24]	**F1WM**	**Rx FIFO 1 Watermark** 0= Watermark interrupt disabled. 1-64 = Level for Rx FIFO 1 watermark interrupt (RF1W(CANFD_IR[5])). >64 = Watermark interrupt disabled.
[23]	**Reserved**	Reserved.
[22:16]	**F1S**	**Rx FIFO 1 Size** 0= No Rx FIFO 1. 1-64 = Number of Rx FIFO 1 elements. >64 = Values greater than 64 are interpreted as 64. The Rx FIFO 1 elements are indexed from 0 to F1S - 1
[15:2]	**F1SA**	**Rx FIFO 1 Start Address** Start address of Rx FIFO 1 in Message RAM (32-bit word address, refer to Figure 6.33-11).
[1:0]	**Reserved**	Reserved.

Rx FIFO 1 Status Register (CANFD_RXF1S)

Register	Offset	R/W	Description	Reset Value
CANFD_RXF1S	CANFDn_BA+0x0B4	R	Rx FIFO 1 Status	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						RF1L	F1F
23	22	21	20	19	18	17	16
Reserved		F1PI					
15	14	13	12	11	10	9	8
Reserved		F1GI					
7	6	5	4	3	2	1	0
Reserved	F1FL						

Bits	Description	
[31:26]	**Reserved**	Reserved.
[25]	**RF1L**	**Rx FIFO 1 Message Lost** This bit is a copy of interrupt flag RF1L(CANFD_IR[7]). When RF1L(CANFD_IR[7]) is reset, this bit is also reset. 0= No Rx FIFO 1 message lost. 1= Rx FIFO 1 message lost, also set after write attempt to Rx FIFO 1 of size zero. **Note:** Overwriting the oldest message when F1OM (CANFD_RXF1C[31]) = 1 will not set this flag.
[24]	**F1F**	**Rx FIFO 1 Full** 0 = Rx FIFO 1 is not full. 1 = Rx FIFO 1 is full.
[23:22]	**Reserved**	**Rx FIFO 1 Get Index** Rx FIFO 1 read index pointer, range 0 to 63.
[21:16]	**F1PI**	**Rx FIFO 1 Fill Level** Number of elements stored in Rx FIFO 1, range 0 to 64.
[15:14]	**Reserved**	Reserved.
[13:8]	**F1GI**	**Rx FIFO 1 Get Index** Rx FIFO 1 read index pointer, range 0 to 63.
[7]	**Reserved**	Reserved.
[6:0]	**F1FL**	**Rx FIFO 1 Fill Level** Number of elements stored in Rx FIFO 1, range 0 to 64

Rx FIFO 1 Acknowledge Register (CANFD_RXF1A)

Register	Offset	R/W	Description	Reset Value
CANFD_RXF1A	CANFDn_BA+0x0B8	R/W	Rx FIFO 1 Acknowledge	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		F1AI					

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5:0]	**F1AI**	**Rx FIFO 1 Acknowledge Index** After the Host has read a message or a sequence of messages from Rx FIFO 1 it has to write the buffer index of the last element read from Rx FIFO 1 to F1AI. This will set the Rx FIFO 1 Get Index F1GI (CANFD_RXF1S[13:8]) to F1AI (CANFD_RXF1A[5:0]) + 1 and update the FIFO 1 Fill Level F1FL (CANFD_RXF1S[6:0]).

Rx Buffer / FIFO Element Size Configuration Register (CANFD_RXESC)

Register	Offset	R/W	Description	Reset Value
CANFD_RXESC	CANFDn_BA+0x0BC	R	Rx Buffer / FIFO Element Size Configuration (P*)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved					RBDS		
7	6	5	4	3	2	1	0
Reserved	F1DS			Reserved	F0DS		

Bits	Description	
[31:11]	**Reserved**	Reserved.
[10:8]	**RBDS**	**Rx Buffer Data Field Size** 000 = 8 byte data field. 001 = 12 byte data field. 010 = 16 byte data field. 011 = 20 byte data field. 100 = 24 byte data field. 101 = 32 byte data field. 110 = 48 byte data field. 111 = 64 byte data field.
[7]	**Reserved**	Reserved.
[6:4]	**F1DS**	**Rx FIFO 1 Data Field Size** 000 = 8 byte data field. 001 = 12 byte data field. 010 = 16 byte data field. 011 = 20 byte data field. 100 = 24 byte data field. 101 = 32 byte data field. 110 = 48 byte data field. 111 = 64 byte data field.
[3]	**Reserved**	Reserved.
[2:0]	**F0DS**	**Rx FIFO 0 Data Field Size** 000 = 8 byte data field. 001 = 12 byte data field. 010 = 16 byte data field. 011 = 20 byte data field. 100 = 24 byte data field.

		101 = 32 byte data field. 110 = 48 byte data field. 111 = 64 byte data field. **Note:** In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx Buffer or Rx FIFO, only the number of bytes as configured by CANFD_RXESC are stored to the Rx Buffer resp. Rx FIFO element. The rest of the frame data field is ignored.

Tx Buffer Configuration Register (CANFD_TXBC)

Register	Offset	R/W	Description	Reset Value
CANFD_TXBC	CANFDn_BA+0x0C0	R	Tx Buffer Configuration (P*)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	TFQM	TFQS					
23	22	21	20	19	18	17	16
Reserved		NDTB					
15	14	13	12	11	10	9	8
TBSA							
7	6	5	4	3	2	1	0
TBSA						Reserved	

Bits	Description	
[31]	**Reserved**	Reserved.
[30]	**TFQM**	**Tx FIFO/Queue Mode** 0= Tx FIFO operation. 1= Tx Queue operation.
[29:24]	**TFQS**	**Transmit FIFO/Queue Size** 0= No Tx FIFO/Queue. 1-32= Number of Tx Buffers used for Tx FIFO/Queue. >32= Values greater than 32 are interpreted as 32.
[23:22]	**Reserved**	Reserved.
[21:16]	**NDTB**	**Number of Dedicated Transmit Buffers** 0= No Dedicated Tx Buffers. 1-32= Number of Dedicated Tx Buffers. >32= Values greater than 32 are interpreted as 32.
[15:2]	**TBSA**	**Tx Buffers Start Address** Start address of Tx Buffers section in Message RAM (32-bit word address, refer to Figure 6.33-11). **Note:** The sum of TFQS and NDTB may be not greater than 32. There is no check for erroneous configurations. The Tx Buffers section in the Message RAM starts with the dedicated Tx Buffers
[1:0]	**Reserved**	Reserved.

Tx FIFO / Queue Status Register (CANFD_TXFQS)

Register	Offset	R/W	Description	Reset Value
CANFD_TXFQS	CANFDn_BA+0x0C4	R	Tx FIFO/Queue Status	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved		TFQF	TFQPI				
15	14	13	12	11	10	9	8
Reserved			TFGI				
7	6	5	4	3	2	1	0
Reserved		TFFL					

Bits	Description	
[31:22]	**Reserved**	Reserved.
[21]	**TFQF**	**Tx FIFO/Queue Full** 0= Tx FIFO/Queue is not full. 1= Tx FIFO/Queue is full.
[20:16]	**TFQPI**	**Tx FIFO/Queue Put Index** Tx FIFO/Queue write index pointer, range 0 to 31.
[15:13]	**Reserved**	Reserved.
[12:8]	**TFGI**	**Tx FIFO Get Index** Tx FIFO read index pointer, range 0 to 31. Read as zero when Tx Queue operation is configured (TFQM (CANFD_TXBC[30]) = 1).
[7:6]	**Reserved**	Reserved.
[5:0]	**TFFL**	**Tx FIFO Free Level** Number of consecutive free Tx FIFO elements starting from TFGI, range 0 to 32. Read as zero when Tx Queue operation is configured (TFQM (CANFD_TXBC[30]) = 1). **Note:** In case of mixed configurations where dedicated Tx Buffers are combined with a Tx FIFO or a Tx Queue, the Put and Get Indices indicate the number of the Tx Buffer starting with the first dedicated Tx Buffers. Example: For a configuration of 12 dedicated Tx Buffers and a Tx FIFO of 20 Buffers a Put Index of 15 points to the fourth buffer of the Tx FIFO.

Tx Buffer Element Size Configuration Register (CANFD_TXESC)

Register	Offset	R/W	Description	Reset Value
CANFD_TXESC	CANFDn_BA+0x0C8	R	Tx Buffer Element Size Configuration (P*)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					TBDS		

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2:0]	**TBDS**	**Tx Buffer Data Field Size** 000 = 8 byte data field. 001 = 12 byte data field. 010 = 16 byte data field. 011 = 20 byte data field. 100 = 24 byte data field. 101 = 32 byte data field. 110 = 48 byte data field. 111 = 64 byte data field. **Note:** In case the data length code DLC of a Tx Buffer element is configured to a value higher than the Tx Buffer data field size TBDS(CANFD_TXESC[2:0]), the bytes not defined by the Tx Buffer are transmitted as 0xCC (padding bytes).

Tx Buffer Request Pending Register (CANFD_TXBRP)

Register	Offset	R/W	Description	Reset Value
CANFD_TXBRP	CANFDn_BA+0x0CC	R	Tx Buffer Request Pending	0x0000_0000

31	30	29	28	27	26	25	24
TRPn							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TRPn							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TRPn							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TRPn							

Bits	Description	
[31:0]	**TRPn**	**Transmission Request Pending** **Each Tx Buffer has its own Transmission Request Pending bit The bits are set via register CANFD_TXBAR The bits are reset after a requested transmission has completed or has been cancelled via register CANFD_TXBCR** CANFD_TXBRP bits are set only for those Tx Buffers configured via CANFD_TXBC. After a CANFD_TXBRP bit has been set, a Tx scan (refer to 6.33.5.5, Tx Handling) is started to check for the pending Tx request with the highest priority (Tx Buffer with lowest Message ID). A cancellation request resets the corresponding transmission request pending bit of register CANFD_TXBRP. In case a transmission has already been started when a cancellation is requested, this is done at the end of the transmission, regardless whether the transmission was successful or not. The cancellation request bits are reset directly after the corresponding CANFD_TXBRP bit has been reset. After a cancellation has been requested, a finished cancellation is signaled via CANFD_TXBCF • after successful transmission together with the corresponding CANFD_TXBTO bit • when the transmission has not yet been started at the point of cancellation • when the transmission has been aborted due to lost arbitration • when an error occurred during frame transmission In DAR mode all transmissions are automatically cancelled if they are not successful. The corresponding CANFD_TXBCF bit is set for all unsuccessful transmissions. 0 = No transmission request pending. 1 = Transmission request pending. **Note:** CANFD_TXBRP bits which are set while a Tx scan is in progress are not considered during this particular Tx scan. In case a cancellation is requested for such a Tx Buffer, this Add Request is cancelled immediately, the corresponding CANFD_TXBRP bit is reset.

Tx Buffer Add Request Register (CANFD_TXBAR)

Register	Offset	R/W	Description	Reset Value
CANFD_TXBAR	CANFDn_BA+0x0D0	R/W	Tx Buffer Add Request	0x0000_0000

31	30	29	28	27	26	25	24
ARn							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ARn							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ARn							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ARn							

Bits	Description	
[31:0]	**ARn**	**Add Request** **Each Tx Buffer has its own Add Request bit Writing a 1 will set the corresponding Add Request bit; writing a 0 has no impact This enables the Host to set transmission requests for multiple Tx Buffers with one write to CANFD_TXBAR CANFD_TXBAR bits are set only for those Tx Buffers configured via CANFD_TXBC When no Tx scan is running, the bits are reset immediately, else the bits remain set until the Tx scan process has completed** 0 = No transmission request added. 1 = Transmission requested added. **Note:** If an add request is applied for a Tx Buffer with pending transmission request (corresponding CANFD_TXBRP bit already set), this add request is ignored.

Tx Buffer Cancellation Request Register (CANFD_TXBCR)

Register	Offset	R/W	Description	Reset Value
CANFD_TXBCR	CANFDn_BA+0x0D4	R/W	Tx Buffer Cancellation Request	0x0000_0000

31	30	29	28	27	26	25	24
CRn							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CRn							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CRn							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CRn							

Bits	Description	
[31:0]	**CRn**	**Cancellation Request** Each Tx Buffer has its own Cancellation Request bit. Writing a 1 will set the corresponding Cancellation Request bit; writing a 0 has no impact. This enables the Host to set cancellation requests for multiple Tx Buffers with one write to CANFD_TXBCR. CANFD_TXBCR bits are set only for those Tx Buffers configured via CANFD_TXBC. The bits remain set until the corresponding bit of CANFD_TXBRP is reset. 0 = No cancellation pending. 1 = Cancellation pending.

Tx Buffer Transmission Occurred Register (CANFD_TXBTO)

Register	Offset	R/W	Description	Reset Value
CANFD_TXBTO	CANFDn_BA+0x0D8	R	Tx Buffer Transmission Occurred	0x0000_0000

31	30	29	28	27	26	25	24
TOn							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TOn							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TOn							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TOn							

Bits	Description	
[31:0]	**TOn**	**Transmission Occurred** Each Tx Buffer has its own Transmission Occurred bit. The bits are set when the corresponding CANFD_TXBRP bit is cleared after a successful transmission. The bits are reset when a new transmission is requested by writing a 1 to the corresponding bit of register CANFD_TXBAR. 0 = No transmission occurred. 1 = Transmission occurred.

Tx Buffer Cancellation Finished Register (CANFD_TXBCF)

Register	Offset	R/W	Description	Reset Value
CANFD_TXBCF	CANFDn_BA+0x0DC	R	Tx Buffer Cancellation Finished	0x0000_0000

31	30	29	28	27	26	25	24
CFn							
23	22	21	20	19	18	17	16
CFn							
15	14	13	12	11	10	9	8
CFn							
7	6	5	4	3	2	1	0
CFn							

Bits	Description	
[31:0]	CFn	**Cancellation Finished** Each Tx Buffer has its own Cancellation Finished bit. The bits are set when the corresponding CANFD_TXBRP bit is cleared after a cancellation was requested via CANFD_TXBCR. In case the corresponding CANFD_TXBRP bit was not set at the point of cancellation, CF is set immediately. The bits are reset when a new transmission is requested by writing a 1 to the corresponding bit of register CANFD_TXBAR. 0 = No transmit buffer cancellation. 1 = Transmit buffer cancellation finished.

Tx Buffer Transmission Interrupt Enable Register (CANFD_TXBTIE)

Register	Offset	R/W	Description	Reset Value
CANFD_TXBTIE	CANFDn_BA+0x0E0	R/W	Tx Buffer Transmission Interrupt Enable	0x0000_0000

31	30	29	28	27	26	25	24
TIEn							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TIEn							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TIEn							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TIEn							

Bits	Description	
[31:0]	**TIEn**	**Transmission Interrupt Enable** Each Tx Buffer has its own Transmission Interrupt enable bit. 0 = Transmission interrupt disabled. 1 = Transmission interrupt enable.

Tx Buffer Cancellation Finished Interrupt Enable Register (CANFD_TXBCIE)

Register	Offset	R/W	Description	Reset Value
CANFD_TXBCIE	CANFDn_BA+0x0E4	R/W	Tx Buffer Cancellation Finished Interrupt Enable	0x0000_0000

31	30	29	28	27	26	25	24
CFIEn							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CFIEn							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CFIEn							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CFIEn							

Bits	Description	
[31:0]	**CFIEn**	**Cancellation Finished Interrupt Enable** Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 = Cancellation finished interrupt disabled. 1 = Cancellation finished interrupt enabled.

Tx Event FIFO Configuration Register (CANFD_TXEFC)

Register	Offset	R/W	Description	Reset Value
CANFD_TXEFC	CANFDn_BA+0x0F0	R	Tx Event FIFO Configuration (P*)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		EFWM					
23	22	21	20	19	18	17	16
Reserved		EFS					
15	14	13	12	11	10	9	8
EFSA							
7	6	5	4	3	2	1	0
EFSA						Reserved	

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29:24]	**EFWM**	**Event FIFO Watermark** 0 = Watermark interrupt disabled. 1-32= Level for Tx Event FIFO watermark interrupt (TEFW (CANFD_IR[13])). >32= Watermark interrupt disabled.
[23:22]	**Reserved**	Reserved.
[21:16]	**EFS**	**Event FIFO Size** 0= Tx Event FIFO disabled. 1-32= Number of Tx Event FIFO elements. >32= Values greater than 32 are interpreted as 32. The Tx Event FIFO elements are indexed from 0 to EFS - 1
[15:2]	**EFSA**	**Event FIFO Start Address** Start address of Tx Event FIFO in Message RAM (32-bit word address, refer to Figure 6.33-11).
[1:0]	**Reserved**	Reserved.

Tx Event FIFO Status Register (CANFD_TXEFS)

Register	Offset	R/W	Description	Reset Value
CANFD_TXEFS	CANFDn_BA+0x0F4	R	Tx Event FIFO Status	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						TEFL	EFF
23	22	21	20	19	18	17	16
Reserved			EFPI				
15	14	13	12	11	10	9	8
Reserved			EFGI				
7	6	5	4	3	2	1	0
Reserved		EFFL					

Bits	Description	
[31:26]	**Reserved**	Reserved.
[25]	**TEFL**	**Tx Event FIFO Element Lost** This bit is a copy of interrupt flag TEFL (CANFD_IR[15]). When TEFL is reset, this bit is also reset. 0= No Tx Event FIFO element lost. 1= Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero.
[24]	**EFF**	**Event FIFO Full** 0= Tx Event FIFO is not full. 1= Tx Event FIFO is full.
[23:21]	**Reserved**	Reserved.
[20:16]	**EFPI**	**Event FIFO Put Index** Tx Event FIFO write index pointer, range 0 to 31
[15:13]	**Reserved**	Reserved.
[12:8]	**EFGI**	**Event FIFO Get Index** Tx Event FIFO read index pointer, range 0 to 31
[7:6]	**Reserved**	Reserved.
[5:0]	**EFFL**	**Event FIFO Fill Level** Number of elements stored in Tx Event FIFO, range 0 to 32

Tx Event FIFO Acknowledge Register (CANFD_TXEFA)

Register	Offset	R/W	Description	Reset Value
CANFD_TXEFA	CANFDn_BA+0x0F8	R/W	Tx Event FIFO Acknowledge	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			EFAI				

Bits	Description	
[31:5]	**Reserved**	Reserved.
[4:0]	**EFAI**	**Event FIFO Acknowledge Index** After the Host has read an element or a sequence of elements from the Tx Event FIFO it has to write the index of the last element read from Tx Event FIFO to EFAI. This will set the Tx Event FIFO Get Index EFGI (CANFD_TXEFS[12:8]) to EFAI + 1 and update the Event FIFO Fill Level EFFL (CANFD_TXEFS[5:0])

6.34 Gigabit Media Access Controller (GMAC)

6.34.1 Overview

Portions Copyright © Synopsys, Inc. Used with permission. All rights reserved.

The GMAC enables a host to transmit and receive data over Ethernet in compliance with the IEEE 802.3-2008 standard. This chip only supports Reduced Gigabit Media Independent Interface (RGMII) and Reduced Media Independent Interface (RMII) interfaces.

The GMAC is compliant with the following standards:

- IEEE 802.3-2008 for Ethernet MAC
- IEEE 1588-2008 standard for precision networked clock synchronization
- IEEE 802.3az-2010, for Energy Efficient Ethernet (EEE)
- AMBA 2.0 for AHB slave port
- AMBA 3.0 for AXI master port
- RGMII specification version 2.6 from HP/Marvell
- RMII specification version 1.2 from RMII consortium

6.34.2 Features

- 10, 100, and 1000 Mbps data transfer rates with the following PHY interfaces:
  - RGMII interface to communicate with an external gigabit PHY
  - RMII interface to communicate with an external Fast Ethernet PHY
- Full-duplex operation:
  - IEEE 802.3x flow control automatic transmission of zero-quanta Pause frame on flow control input de-assertion
  - Forwarding of received Pause frames to the user application
- Half-duplex operation:
  - CSMA/CD Protocol support
  - Flow control using backpressure support
  - Frame bursting and frame extension in 1000 Mbps half-duplex operation
- Preamble and start of frame data (SFD) insertion in Transmit path
- Preamble and SFD deletion in the Receive path
- Automatic CRC and pad generation controllable on a per-frame basis
- Automatic Pad and CRC Stripping options for receive frames
- Programmable frame length to support Standard or Jumbo Ethernet frames with up to 16 Kbytes of size
- Programmable Interframe Gap (IFG) (40-96 bit times in steps of 8)
- Option to transmit frames with reduced preamble size
- Separate 32-bit status for transmit and receive packets
- IEEE 802.1Q VLAN tag detection for reception frames
- Separate transmission, reception, and control interfaces to the application
- Receive module for checksum off-load for received IPv4 and TCP packets encapsulated by

the Ethernet frame (Type 1)

- Enhanced Receive module for checking IPv4 header checksum and TCP, UDP, or ICMP checksum encapsulated in IPv4 or IPv6 datagrams (Type 2)
- Support Ethernet frame timestamping as described in IEEE 1588-2002 and IEEE 1588-2008

  The 64-bit timestamps are given in the transmit or receive status of each frame
- MDIO master interface for PHY device configuration and management
- Standard IEEE 802.3az-2010 for Energy Efficient Ethernet
- Flexibility to control the Pulse-Per-Second (PPS) output signal (ptp_pps_o)
- CRC replacement, Source Address field insertion or replacement, and VLAN insertion, replacement, and deletion in transmitted frames with per-frame control
- Programmable watchdog timeout limit in the receive path

6.34.3 Architecture

6.34.3.1 Block Diagram

The GMAC is composed of 4 main blocks:

- The **Direct Memory Access (DMA)** interface
- The **MAC Transaction Layer (MTL)** transfers data to system memory through the AXI master interface.
- The host CPU uses the AHB slave interface to access the GMAC subsystem's **Control and Status registers (CSRs) and Operating Mode Registers (OMRs)**
- The **Media Access Control (MAC)** module

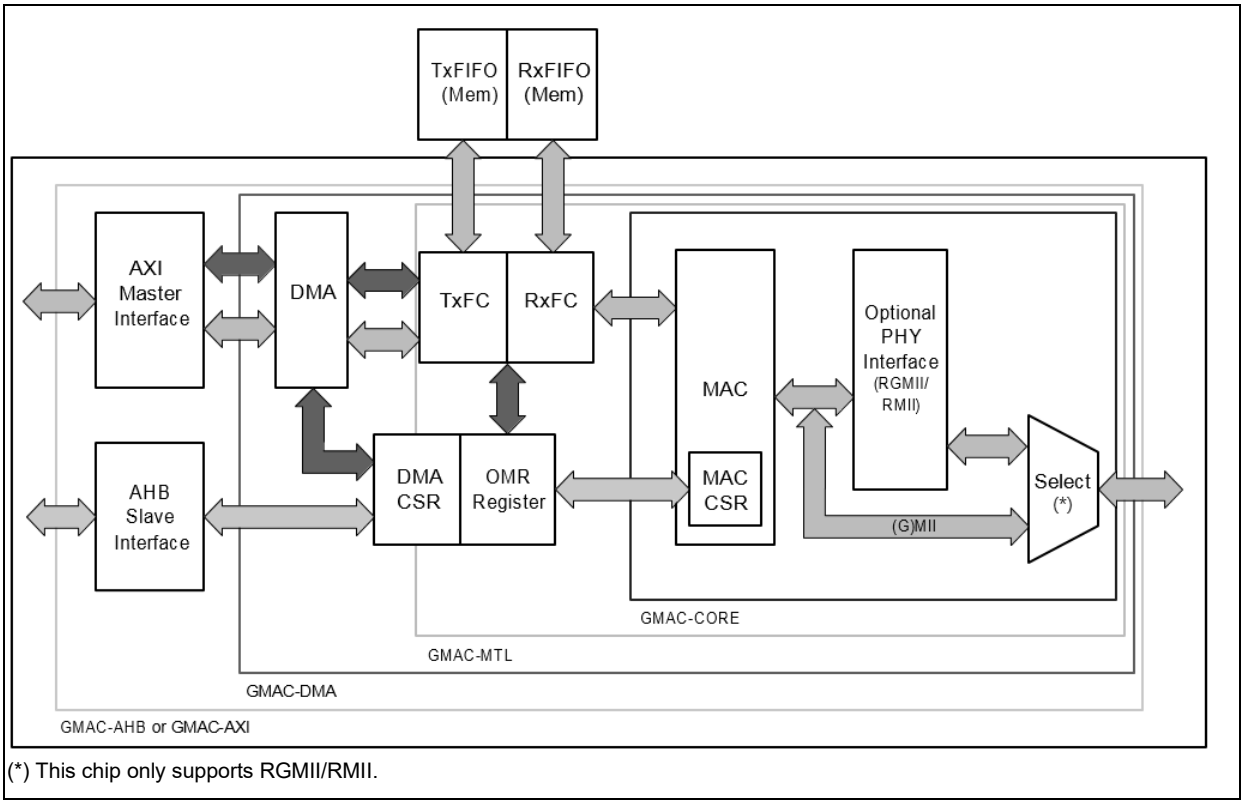


Figure 6.34-1 GMAC Block Diagram

6.34.3.2 DMA Controller

The DMA has independent Transmit and Receive engines, and a CSR space. The Transmit engine transfers data from system memory to the device port (MTL), while the Receive engine transfers data from the device port to the system memory. The controller use descriptors to efficiently move data from source to destination with minimal Host CPU intervention. The DMA is designed for packet-oriented data transfers such as frames in Ethernet. The controller can be programmed to interrupt the Host CPU for situations such as Frame Transmit and Receive transfer completion, and other normal/error conditions.

The DMA and the Host driver communicate through two data structures:

- Control and Status registers (CSR)
- Descriptor lists and data buffers

Control and Status registers are described in detail in “Register Description”. Descriptors are described in detail in “Descriptors”.

The DMA transfers data frames received by the MAC to the Receive Buffer in the Host memory, and Transmit data frames from the Transmit Buffer in the Host memory. Descriptors that reside in the Host memory act as pointers to these buffers.

There are two descriptor lists; one for reception, and one for transmission. The base address of each list is written into Register 1003 (Receive Descriptor List Address Register) and Register 1004 (Transmit Descriptor List Address Register), respectively. A descriptor list is forward linked (either implicitly or explicitly). The last descriptor may point back to the first entry to create a ring structure. Explicit chaining of descriptors is accomplished by setting the second address chained in both Receive and Transmit descriptors (RDES1[24] and TDES1[24]). The descriptor lists resides in the Host physical memory address space. Each descriptor can point to a maximum of two buffers. This enables two buffers to be used, physically addressed, rather than contiguous buffers in memory.

A data buffer resides in the Host physical memory space, and consists of an entire frame or part of a frame, but cannot exceed a single frame. Buffers contain only data, buffer status is maintained in the descriptor. Data chaining refers to frames that span multiple data buffers. However, a single descriptor cannot span multiple frames. The DMA skips to the next frame buffer when the end of frame is detected. Data chaining can be enabled or disabled.

The descriptor ring and chain structure is shown in Figure 6.34-2.

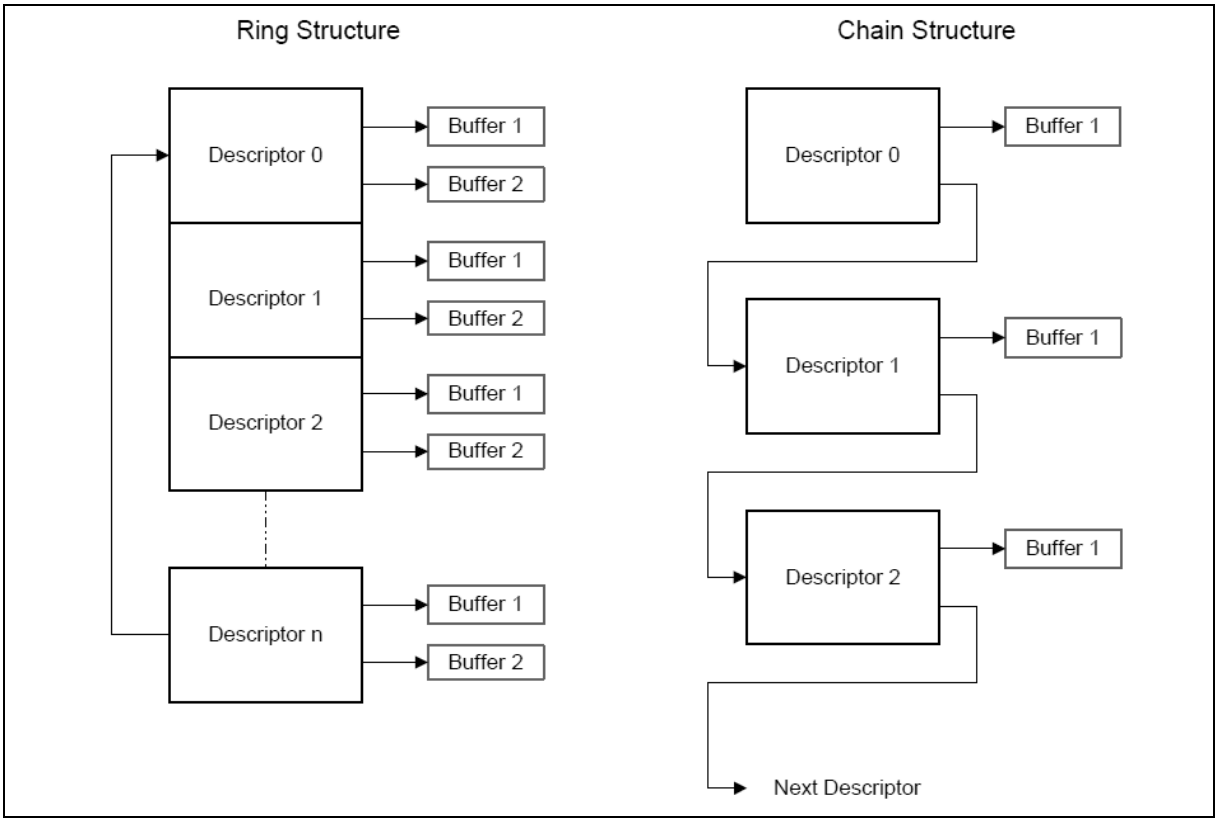


Figure 6.34-2 Descriptor Ring and Chain Structure

Initialization

Initialization for the GMAC is as follows:

1. Write to Register 1000 (Bus Mode Register) to set Host bus access parameters.
2. Write to Register 1007 (Interrupt Enable Register) to mask unnecessary interrupt causes.
3. Create the Transmit and Receive descriptor lists, and then write to DMA Register 1003 (Receive Descriptor List Address Register) and Register 1004 (Transmit Descriptor List Address Register), providing the DMA with the starting address of each list.

5. Write to Register 0 (MAC Configuration Register) to configure the operating mode and enable the transmit operation (Bit 3).
6. Program Bit 15 (PS) and Bit 11 (DM) of Register 0 (MAC Configuration Register) based on the line status received or read from the PHY status registers after auto-negotiation.
7. Write to Register 1006 (Operation Mode Register) to set Bits 13 and 1 to start transmission and reception.
8. Write to Register 0 (MAC Configuration Register) to enable the Receive operation (Bit 2).

   The Transmit and Receive engines enter the Running state and attempt to acquire descriptors

from the respective descriptor lists. The Receive and Transmit engines then begin processing Receive and Transmit operations. The Transmit and Receive processes are independent of each other and can be started or stopped separately.

Host Bus Burst Access

The DMA attempts to execute fixed-length Burst transfers on the AHB or AXI master interface if configured to do so through FB bit of Register 1000 (Bus Mode Register). The maximum Burst length is indicated and limited by the PBL field (Bits [13:8]) Register 1000 (Bus Mode Register). The Receive and Transmit descriptors are always accessed in the maximum possible (limited by PBL or 16 * 8/bus width) burst-size for the 16- bytes to be read.

The Transmit DMA initiates a data transfer only when sufficient space to accommodate the configured burst is available in MTL Transmit FIFO or the number of bytes till the end of frame (when it is less than the configured burst length). The DMA indicates the start address and the number of transfers required to the AHB or AXI master interface. When the AHB or AXI Interface is configured for fixed-length burst, then it transfers data using the best combination of INCR4, 8, or 16 and SINGLE transactions. Otherwise (no fixed- length burst), it transfers data using INCR (undefined length) and SINGLE transactions.

The Receive DMA initiates a data transfer only when sufficient data to accommodate the configured burst is available in MTL Receive FIFO or when the end of frame (when it is less than the configured burst length) is detected in the Receive FIFO. The DMA indicates the start address and the number of transfers required to the AHB or AXI master interface. When the AHB or AXI Interface is configured for fixed-length burst, then it transfers data using the best combination of INCR4, 8, or 16 and SINGLE transactions. If the end-of frame is reached before the fixed-burst ends on the AHB or AXI interface, then dummy transfers are performed in- order to complete the fixed-burst. Otherwise (FB bit of Register 1000 (Bus Mode Register) is reset), it transfers data using INCR (undefined length) and SINGLE transactions.

When the AHB or AXI interface is configured for address-aligned beats, both DMA engines ensure that the first burst transfer the AHB or AXI initiates is less than or equal to the size of the configured PBL. Thus, all subsequent beats start at an address that is aligned to the configured PBL. The DMA can only align the address for beats up to size 16 (for PBL > 16), because the AHB or AXI interface does not support more than INCR16.

Host Data Buffer Alignment

The Transmit and Receive data buffers do not have any restrictions on start address alignment. For example, in systems with 32-bit memory, the start address for the buffers can be aligned to any of the four bytes. However, the DMA always initiates write transfers, with address aligned to the bus width, with dummy data (old data) in the byte lanes that are not valid. This typically happens during the transfer of the beginning or end of an Ethernet frame. The software driver should discard the dummy bytes based on the start address of the buffer and size of the frame.

Example 3-1 Buffer Read

If the Transmit buffer address is 32’h00000FF2 (for 32-bit data bus), and 15 bytes need to be transferred, then the DMA reads five full words from address 32’h00000FF0, but when transferring data to the MTL Transmit FIFO, the extra bytes (the first two bytes) are dropped or ignored. Similarly, the last 3 bytes of the last transfer are also ignored. The DMA always ensures that it transfers a full 32-bit data to the MTL Transmit FIFO, unless it is the end of frame.

Example 3-2 Buffer Write

If the Receive buffer address is 32’h0000FF2 (for 64-bit data bus) and 16 bytes of a received frame need to be transferred, then the DMA writes 3 full words from address 32’h00000FF0. But the first 2 bytes of first transfer and the last 6 bytes of the third transfer have dummy data.

The DMA considers the offset address only if it is the first Receive buffer of the frame. The DMA ignores the offset address and performs full word writes for the middle and the last Receive buffer of the frame.

Buffer Size Calculations

The DMA does not update the size fields in the Transmit and Receive descriptors. The DMA updates only the status fields (RDES and TDES) of the descriptors. The driver has to perform the size calculations.

The Transmit DMA transfers the exact number of bytes (indicated by buffer size field of TDES1) towards the MAC. If a descriptor is marked as first (FS bit of TDES1 is set), then the DMA marks the first transfer from the buffer as the start of frame. If a descriptor is marked as last (LS bit of TDES1), then the DMA marks the last transfer from that data buffer as the end-of frame to the MTL.

The Receive DMA transfers data to a buffer until the buffer is full or the end-of frame is received from the MTL. If a descriptor is not marked as last (LS bit of RDES0), then the descriptor's corresponding buffer(s) are full and the amount of valid data in a buffer is accurately indicated by its buffer size field minus the data buffer pointer offset when the FS bit of that descriptor is set. The offset is zero when the data buffer pointer is aligned to the data bus width. If a descriptor is marked as last, then the buffer may not be full (as indicated by the buffer size in RDES1). To compute the amount of valid data in this final buffer, the driver must read the frame length (FL bits of RDES0[29: 16]) and subtract the sum of the buffer sizes of the preceding buffers in this frame. The Receive DMA always transfers the start of next frame with a new descriptor.

Even when the start address of a receive buffer is not aligned to the data width of system bus, the system should allocate a receive buffer of a size aligned to the system bus width. For example, if the system allocates a 1,024-byte (1 Kbytes) receive buffer starting from address 0x1000, the software can program the buffer start address in the Receive descriptor to have a 0x1002 offset. The Receive DMA writes the frame to this buffer with dummy data in the first two locations (0x1000 and 0x1001). The actual frame is written from location 0x1002. Thus, the actual useful space in this buffer is 1,022 bytes, even though the buffer size is programmed as 1,024 bytes, because of the start address offset.

DMA Arbiter

The arbiter inside the DMA module performs the arbitration between the Transmit and Receive channel accesses to the AHB master interface. Two types of arbitrations are possible: round-robin, and fixed-priority.

When round-robin arbitration is selected (DA bit of Register 1000 (Bus Mode Register) is reset), the arbiter allocates the data bus in the ratio set by the PR bits of Register 1000 (Bus Mode Register), when both Transmit and Receive DMAs are requesting for access simultaneously. When the DA bit is set, the Receive DMA always gets priority over the Transmit DMA for data access by default. When Bit 27 (TXPR) of Register 1000 (Bus Mode Register) is also set, then the Transmit DMA gets priority over the Receive DMA.

Transmission

Tx DMA Operation: Default (Non-OSF) Mode

The Transmit DMA engine by default mode proceeds as follows:

1. The Host sets up the transmit descriptor (TDES0-TDES3) and sets the Own bit (TDES0[31]) after setting up the corresponding data buffer(s) with Ethernet Frame data.
2. When Bit 13 (ST) of Register 1006 (Operation Mode Register) is set, the DMA enters the Run state.
3. While in the Run state, the DMA polls the Transmit Descriptor list for frames requiring transmission. After polling starts, it continues in either sequential descriptor ring order or chained order. If the DMA detects a descriptor flagged as owned by the Host (TDES0[31] = 1'b0), or if an error condition occurs, transmission is suspended and both the Bit 2 (Transmit Buffer Unavailable) and Bit 16 (Normal Interrupt Summary) of the Register 1005 (Status Register) are set. The Transmit Engine proceeds to step 9.
4. If the acquired descriptor is flagged as owned by DMA (TDES0[31] = 1'b1), the DMA decodes the Transmit Data Buffer address from the acquired descriptor.
5. The DMA fetches the Transmit data from the Host memory and transfers the data to the MTL

for transmission.

6. If an Ethernet frame is stored over data buffers in multiple descriptors, the DMA closes the intermediate descriptor and fetches the next descriptor. Steps 3, 4, and 5 are repeated until the end-of- Ethernet-frame data is transferred to the MTL.
7. When frame transmission is complete, if IEEE 1588 timestamping was enabled for the frame (as indicated in the transmit status) the timestamp value obtained from MTL is written to the transmit descriptor (TDES2 and TDES3) that contains the end-of-frame buffer. The status information is then written to this transmit descriptor (TDES0). Because the Own bit is cleared during this step, the Host now owns this descriptor. If timestamping was not enabled for this frame, the DMA does not alter the contents of TDES2 and TDES3.
8. Bit 0 (Transmit Interrupt) of Register 1005 (Status Register) is set after completing transmission of a frame that has Interrupt on Completion (TDES1[31]) set in its Last Descriptor. The DMA engine then returns to step 3.
9. In the Suspend state, the DMA tries to reacquire the descriptor (and thereby return to step 3) when it receives a Transmit Poll demand and the Underflow Interrupt Status bit is cleared. If the host stopped the DMA by clearing Bit 13 (ST) of Register 1006 (Operation Mode Register), the DMA enters the STOP state.

The Tx DMA transmission flow by default mode is shown in Figure 6.34-3.

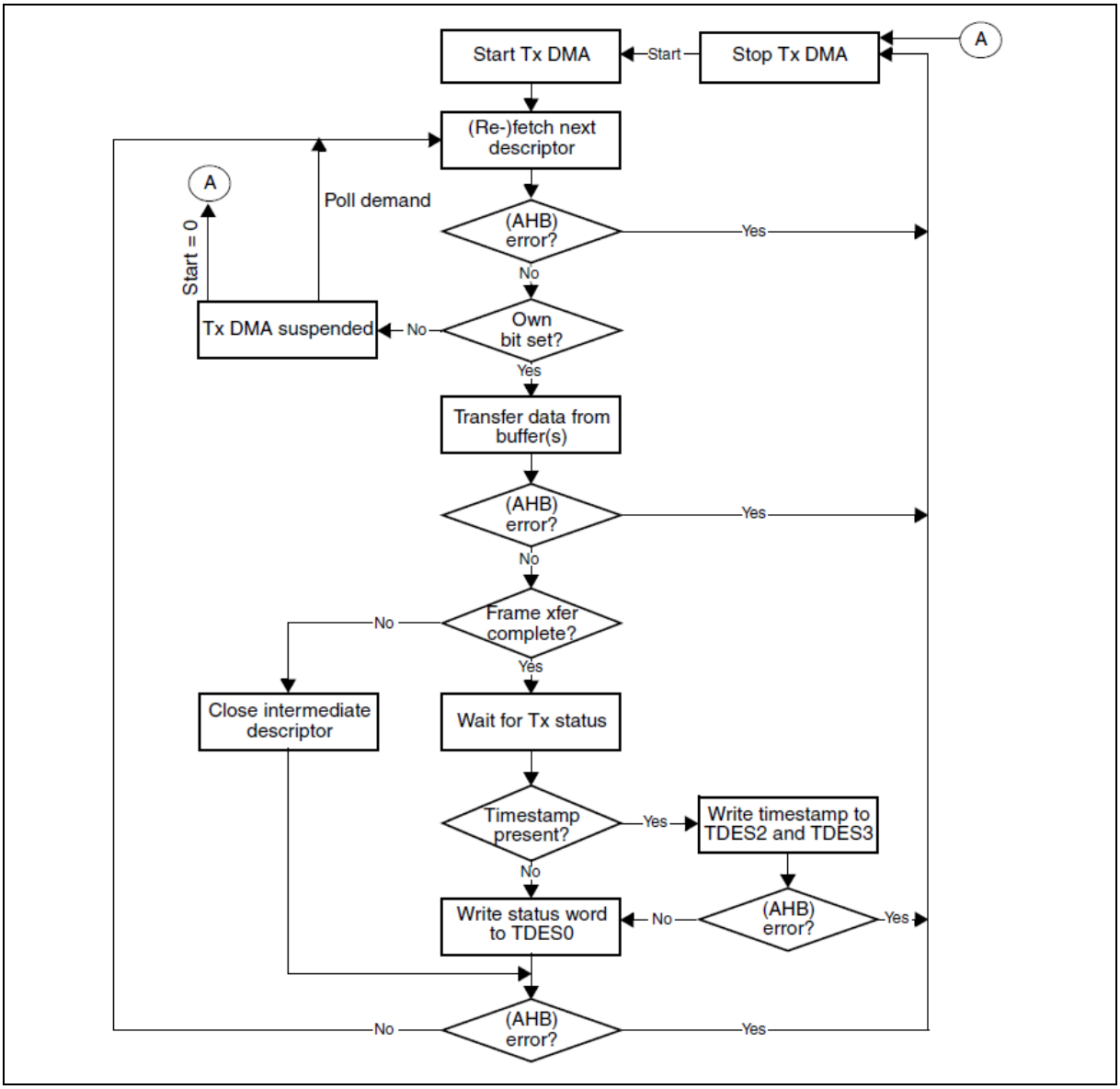


Figure 6.34-3 Tx DMA Operation by default Mode

Tx DMA Operation: OSF Mode

While in the Run state, the transmit process can simultaneously acquire two frames without closing the Status descriptor of the first [if Bit 2 (OSF) in Register 1006 (Operation Mode Register) is set]. As the transmit process finishes transferring the first frame, it immediately polls the Transmit Descriptor list for the second frame. If the second frame is valid, the transmit process transfers this frame before writing the first frame's status information.

In OSF mode, the Run state Transmit DMA operates in the following sequence:

1. The DMA operates as described in steps 1–6 of the Tx DMA (default mode).
2. Without closing the previous frame's last descriptor, the DMA fetches the next descriptor.
3. If the DMA owns the acquired descriptor, the DMA decodes the transmit buffer address in this descriptor. If the DMA does not own the descriptor, the DMA goes into Suspend mode and skips to step 7.
4. The DMA fetches the Transmit frame from the Host memory and transfers the frame to the MTL

until the end-of-frame data is transferred, closing the intermediate descriptors if this frame is split across multiple descriptors.

5. The DMA waits for the previous frame's frame transmission status and timestamp. When the status is available, the DMA writes the timestamp to TDES2 and TDES3, if such timestamp was captured (as indicated by a status bit). The DMA then writes the status, with a cleared Own bit, to the corresponding TDES0, thus closing the descriptor. If timestamping was not enabled for the previous frame, the DMA does not alter the contents of TDES2 and TDES3.
6. If enabled, the Transmit interrupt is set, the DMA fetches the next descriptor, then proceeds to step 3 (when Status is normal). If the previous transmission status shows an underflow error, the DMA goes into Suspend mode (step 7).
7. In Suspend mode, if a pending status and timestamp are received from the MTL, the DMA writes the timestamp (if enabled for the current frame) to TDES2 and TDES3, then writes the status to the corresponding TDES0. It then sets relevant interrupts and returns to Suspend mode.

   If no status is pending and the host stopped the DMA by clearing Bit 13 (ST) of Register 1006 (Operation Mode Register), the DMA enters the STOP state.
8. The DMA can exit Suspend mode and enter the Run state (go to step 1 or step 2 depending on pending status) only after receiving a Transmit Poll demand (Register 1001 (Transmit Poll Demand Register)).

The DMA fetches the next descriptor in advance before closing the current descriptor. Therefore, the descriptor chain should have more than two different descriptors for correct and proper operation.

The basic flow is described in Figure 6.34-4.

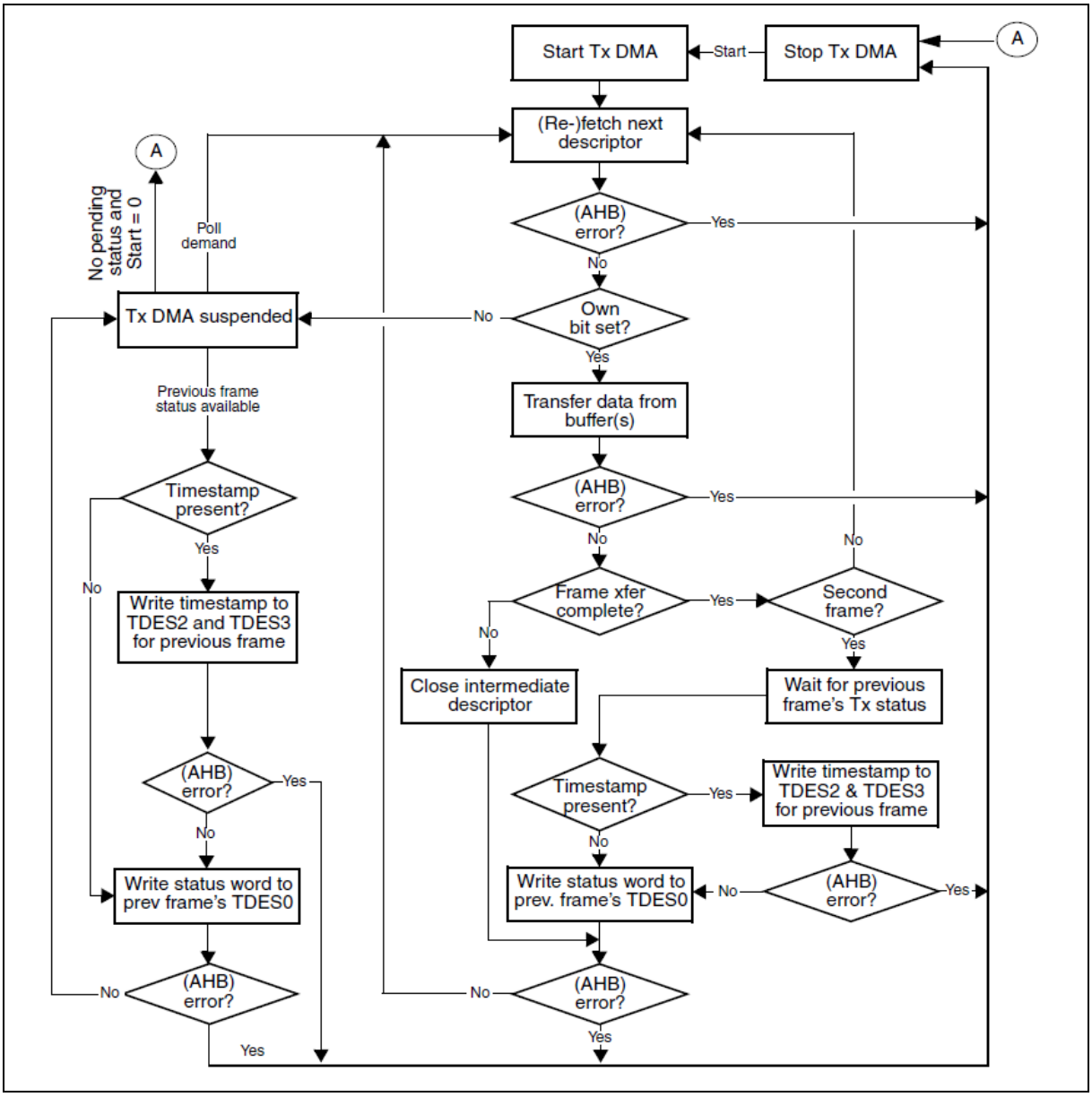


Figure 6.34-4 Tx DMA Operation in OSF Mode

Transmit Frame Processing

The Transmit DMA expects that the data buffers contain complete Ethernet frames, excluding preamble, pad bytes, and FCS fields. The DA, SA, and Type/Length fields contain valid data. If the Transmit Descriptor indicates that the MAC must disable CRC or PAD insertion, the buffer must have complete Ethernet frames (excluding preamble), including the CRC bytes.

Frames can be data-chained and can span several buffers. Frames must be delimited by the First Descriptor (TDES1[29]) and the Last Descriptor (TDES1[30]), respectively.

As transmission starts, the First Descriptor must have (TDES1[29]) set. When this occurs, frame data transfers from the Host buffer to the MTL Transmit FIFO. Concurrently, if the current frame has the Last Descriptor (TDES1[30]) clear, the Transmit Process attempts to acquire the Next Descriptor. The Transmit Process expects this descriptor to have TDES1[29] clear. If TDES1[30] is clear, it indicates an intermediary buffer. If TDES1[30] is set, it indicates the last buffer of the frame.

After the last buffer of the frame has been transmitted, the DMA writes back the final status information

to the Transmit Descriptor 0 (TDES0) word of the descriptor that has the last segment set in Transmit Descriptor 1 (TDES1[30]). At this time, if Interrupt on Completion (TDES1[31]) is set, the Bit 0 (Transmit Interrupt) of Register 1005 (Status Register) is set, the Next Descriptor is fetched, and the process repeats.

The actual frame transmission begins after the MTL Transmit FIFO has reached either a programmable transmit threshold (Bits [16:14] of Register 1006 (Operation Mode Register)), or a full frame is contained in the FIFO. There is also an option for Store and Forward Mode (Bit 21 of Register 1006 (Operation Mode Register)). Descriptors are released (Own bit TDES0[31] clears) when the DMA finishes transferring the frame.

To ensure proper transmission of a frame and the next frame, you must specify a non-zero buffer size for the transmit descriptor that has the Last Descriptor (TDES1[30]) set.

Transmit Polling Suspended

Transmit polling can be suspended by either of the following conditions:

- The DMA detects a descriptor owned by the Host (TDES0[31]=0). To resume, the driver must give descriptor ownership to the DMA and then issue a Poll Demand command.
- A frame transmission is aborted when a transmit error because of underflow is detected. The appropriate Transmit Descriptor 0 (TDES0) bit is set.

If the DMA goes into SUSPEND state because of the first condition, then both Bit 16 (Normal Interrupt Summary) and Bit 2 (Transmit Buffer Unavailable) of Register 1005 (Status Register) are set. If the second condition occurs, both Bit 15 (Abnormal Interrupt Summary) and Bit 5 (Transmit Underflow) of Register 1005 (Status Register) are set, and the information is written to Transmit Descriptor 0, causing the suspension.

In both cases, the position in the Transmit List is retained. The retained position is that of the descriptor following the Last Descriptor closed by the DMA.

The driver must explicitly issue a Transmit Poll Demand command after rectifying the suspension cause.

Reception

The Receive DMA engine's reception sequence is depicted in Figure 6.34-5 and proceeds as follows:

1. The host sets up Receive descriptors (RDES0-RDES3) and sets the Own bit (RDES0[31]).
2. When Bit 1 (SR) of Register 1006 (Operation Mode Register) is set, the DMA enters the Run state. While in the Run state, the DMA polls the Receive Descriptor list, attempting to acquire free descriptors. If the fetched descriptor is not free (is owned by the host), the DMA enters the Suspend state and jumps to step 9.
3. The DMA decodes the receive data buffer address from the acquired descriptors.
4. Incoming frames are processed and placed in the acquired descriptor's data buffers.
5. When the buffer is full or the frame transfer is complete, the Receive engine fetches the next descriptor.
6. If the current frame transfer is complete, the DMA proceeds to step 7. If the DMA does not own the next fetched descriptor and the frame transfer is not complete (EOF is not yet transferred), the DMA sets the Descriptor Error bit in the RDES0 (unless flushing is disabled in Bit 24 of Register 1006 (Operation Mode Register)). The DMA closes the current descriptor (clears the Own bit) and marks it as intermediate by clearing the Last Segment (LS) bit in the RDES0 value (marks it as Last Descriptor if flushing is not disabled), then proceeds to step 8. If the DMA does own the next descriptor but the current frame transfer is not complete, the DMA closes the current descriptor as intermediate and reverts to step 4.
7. If IEEE 1588 timestamping is enabled, the DMA writes the timestamp (if available) to the current descriptor's RDES2 and RDES3. It then takes the receive frame's status from the MTL and writes the status word to the current descriptor's RDES0, with the Own bit cleared and the Last

Segment bit set.

If the host stopped the Rx DMA by clearing Bit 1 (SR) of Register 1006 (Operation Mode Register), the DMA goes to the Stop state. Otherwise, the Rx DMA proceeds to the step 8.

8. The Receive engine checks the latest descriptor's Own bit. If the host owns the descriptor (Own bit is 1'b0), the Bit 7 (Receive Buffer Unavailable) of Register 1005 (Status Register) is set and the DMA Receive engine enters the Suspended state (step 9). If the DMA owns the descriptor, the engine returns to step 4 and awaits the next frame.
9. Before the Receive engine enters the Suspend state, partial frames are flushed from the Receive FIFO. You can control flushing using Bit 24 of Register 1006 (Operation Mode Register).
10. The Receive DMA goes to the STOP state if the host had cleared Bit 1 (SR) of Register 1006 (Operation Mode Register). Otherwise, it exits the Suspend state when a Receive Poll demand is given or the start of next frame is available from the Receive FIFO of the MTL. The engine proceeds to step 2 and refetches the next descriptor.

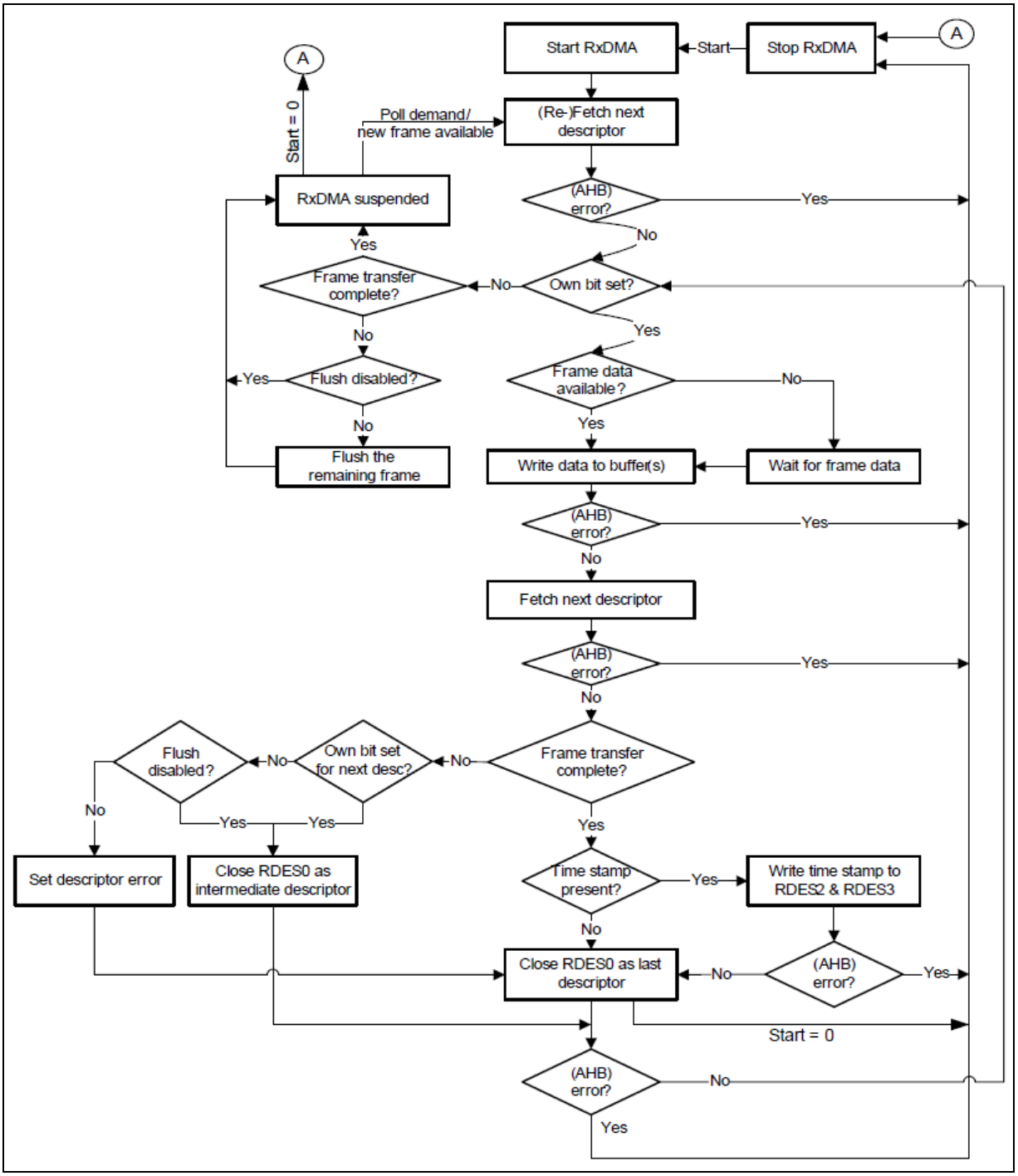


Figure 6.34-5 Receive DMA Operation

The DMA does not acknowledge accepting the status from the MTL until it has completed the timestamp write-back and is ready to perform status write-back to the descriptor.

If software has enabled timestamping through CSR, when a valid timestamp value is not available for the frame (for example, because the receive FIFO was full before the timestamp could be written to it), the DMA writes all-ones to RDES2 and RDES3. Otherwise (that is, if timestamping is not enabled), the RDES2 and RDES3 remain unchanged.

Receive Descriptor Acquisition

The Receive Engine always attempts to acquire an extra descriptor in anticipation of an incoming frame. Descriptor acquisition is attempted if any of the following conditions is satisfied:

- Bit 1 (Start or Stop Receive) of Register 1006 (Operation Mode Register) has been set immediately after being placed in the Run state.
- The data buffer of current descriptor is full before the frame ends for the current transfer.
- The controller has completed frame reception, but the current Receive Descriptor is not yet closed.
- The receive process has been suspended because of a host-owned buffer (RDES0[31] = 0) and a new frame is received.
- A Receive poll demand has been issued.

Receive Frame Processing

The MAC transfers the received frames to the Host memory only when the frame passes the address filter and frame size is greater than or equal to configurable threshold bytes set for the Receive FIFO of MTL, or when the complete frame is written to the FIFO in the store-and-forward mode.

If the frame fails the address filtering, it is dropped in the MAC block itself (unless Bit 31 (Receive All) of Register 1 (MAC Frame Filter) is set). Frames that are shorter than 64 bytes, because of collision or premature termination, can be removed from the MTL Receive FIFO.

After 64 (configurable threshold) bytes have been received, the MTL block requests the DMA block to begin transferring the frame data to the Receive Buffer pointed by the current descriptor. The DMA sets First Descriptor (RDES0[9]) after the DMA Host Interface (AHB/AXI or MDC) becomes ready to receive a data transfer (if DMA is not fetching transmit data from the host), to delimit the frame. The descriptors are released when the Own (RDES[31]) bit is reset to 1'b0, either as the Data buffer fills up or as the last segment of the frame is transferred to the Receive buffer. If the frame is contained in a single descriptor, both Last Descriptor (RDES[8]) and First Descriptor (RDES[9]) are set.

The DMA fetches the next descriptor, sets the Last Descriptor (RDES[8]) bit, and releases the RDES0 status bits in the previous frame descriptor. Then the DMA sets the Bit 6 (Receive Interrupt) of Register 1005 (Status Register). The same process repeats unless the DMA encounters a descriptor flagged as being owned by the host. If this occurs, the Receive Process sets the Bit 7 (Receive Buffer Unavailable) of Register 1005 (Status Register) and then enters the Suspend state. The position in the receive list is retained.

Receive Process Suspended

If a new Receive frame arrives while the Receive Process is in Suspend state, the DMA refetches the current descriptor in the Host memory. If the descriptor is now owned by the DMA, the Receive Process re-enters the Run state and starts frame reception. If the descriptor is still owned by the host, by default, the DMA discards the current frame at the top of the MTL Rx FIFO and increments the missed frame counter. If more than one frame is stored in the MTL Rx FIFO, the process repeats.

The discarding or flushing of the frame at the top of the MTL Rx FIFO can be avoided by disabling Flushing (Bit 24 of Register 1006 (Operation Mode Register)). In such conditions, the receive process sets the Receive Buffer Unavailable status and returns to the Suspend state.

Interrupts

Interrupts can be generated as a result of various events. The DMA Register 1005 (Status Register) contains all the bits that might cause an interrupt. Register 1007 (Interrupt Enable Register) contains an enable bit for each of the events that can cause an interrupt.

There are two groups of interrupts, Normal and Abnormal, as described in Register 1005 (Status Register). Interrupts are cleared by writing a 1'b1 to the corresponding bit position. When all the enabled interrupts within a group are cleared, the corresponding summary bit is cleared. When both the summary bits are cleared, the interrupt signal sbd_intr_o is de-asserted. If the MAC is the cause for assertion of the interrupt, then any of the GLI, GMI, GPI, TTI, or GLPII bits of Register 1005 (Status Register) are

set high.

The Register 1005 (Status Register) is the (interrupt) status register. The interrupt pin is asserted because of any event in this status register only if the corresponding interrupt enable bit is set in Register 1007 (Interrupt Enable Register).

Interrupts are not queued and if the interrupt event occurs before the driver has responded to it, no additional interrupts are generated. For example, Receive Interrupt [Bit 6 of Register 1005 (Status Register)] indicates that one or more frames were transferred to the Host buffer. The driver must scan all descriptors, from the last recorded position to the first one owned by the DMA.

An interrupt is generated only once for simultaneous, multiple events. The driver must scan the Register 1005 (Status Register) for the cause of the interrupt. The interrupt is not generated again unless a new interrupting event occurs, after the driver has cleared the appropriate bit in Register 1005 (Status Register). For example, the controller generates a Receive interrupt [Bit 6 of Register 1005 (Status Register)] and the driver begins reading Register 1005 (Status Register). Next, Receive Buffer Unavailable [Bit 7 of Register 1005 (Status Register)] occurs. The driver clears the Receive interrupt. Even then, the interrupt pin (sbd_intr_o) signal is not de-asserted, because of the active or pending Receive Buffer Unavailable interrupt.

An interrupt timer RIWT (Bits 7:0 in Register 1009 (Receive Interrupt Watchdog Timer Register)) is given for flexible control of Receive Interrupt. When this Interrupt timer is programmed with a non-zero value, it gets activated as soon as the Rx DMA completes a transfer of a received frame to system memory without asserting the Receive Interrupt because it is not enabled in the corresponding Receive Descriptor (RDES1[31] in Table 6.34-21). When this timer runs out as per the programmed value, RI bit is set and the interrupt is asserted if the corresponding RI is enabled in Register 1007 (Interrupt Enable Register). This timer gets disabled before it runs out, when a frame is transferred to memory and the RI is set because it is enabled for that descriptor.

Error Response to DMA

For any data transfer initiated by a DMA channel, if the slave replies with an error response, that DMA stops all operations and updates the error bits and the Fatal Bus Error bit in the Register 1005 (Status Register). The DMA controller can resume operation only after soft resetting or hard resetting the GMAC and re-initializing the DMA.

6.34.4 Basic Configuration

6.34.4.1 GMAC0 Basic Configuration

- Clock Source Configuration
  - Enable GMAC0 controller clock in GMAC0CKEN (CLK_SYSCLK0[27])
- DDR AXI Configuration
  - Enable DDR AXI port3 in port_en (PCTRL_3[0])
- Reset Configuration
  - Reset GMAC0 controller in GMAC0RST (SYS_IPRST0[16])
- Pin Configuration

Group	Pin Name	GPIO	MFP
RGMII0	RGMII0_MDC	PE.0	MFP8
	RGMII0_MDIO	PE.1	MFP8
	RGMII0_TXCTL	PE.2	MFP8
	RGMII0_TXD0	PE.3	MFP8

	RGMII0_TXD1	PE.4	MFP8
	RGMII0_RXCLK	PE.5	MFP8
	RGMII0_RXCTL	PE.6	MFP8
	RGMII0_RXD0	PE.7	MFP8
	RGMII0_RXD1	PE.8	MFP8
	RGMII0_RXD2	PE.9	MFP8
	RGMII0_RXD3	PE.10	MFP8
		PF.0	MFP6
	RGMII0_TXCLK	PE.11	MFP8
		PF.1	MFP6
	RGMII0_TXD2	PE.12	MFP8
		PF.2	MFP6
	RGMII0_TXD3	PE.13	MFP8
		PF.3	MFP6
RMII0	RMII0_MDC	PE.0	MFP9
	RMII0_MDIO	PE.1	MFP9
	RMII0_TXEN	PE.2	MFP9
	RMII0_TXD0	PE.3	MFP9
	RMII0_TXD1	PE.4	MFP9
	RMII0_REFCLK	PE.5	MFP9
	RMII0_CRSDV	PE.6	MFP9
	RMII0_RXD0	PE.7	MFP9
	RMII0_RXD1	PE.8	MFP9
	RMII0_RXERR	PE.9	MFP9

6.34.4.2 GMAC1 Basic Configuration

- Clock Source Configuration
  - Enable GMAC1 controller clock in GMAC1CKEN (CLK_SYSCLK0[28])
- DDR AXI Configuration
  - Enable DDR AXI port3 in port_en (PCTRL_3[0])
- Reset Configuration
  - Reset GMAC1 controller in GMAC1RST (SYS_IPRST0[17])
- Pin Configuration

Group	Pin Name	GPIO	MFP
RGMII1	RGMII1_MDC	PF.0	MFP8
	RGMII1_MDIO	PF.1	MFP8

	RGMII1_TXCTL	PF.2	MFP8
	RGMII1_TXD0	PF.3	MFP8
	RGMII1_TXD1	PF.4	MFP8
	RGMII1_RXCLK	PF.5	MFP8
	RGMII1_RXCTL	PF.6	MFP8
	RGMII1_RXD0	PF.7	MFP8
	RGMII1_RXD1	PF.8	MFP8
	RGMII1_RXD2	PF.9	MFP8
	RGMII1_RXD3	PF.10	MFP8
	RGMII1_TXCLK	PF.11	MFP8
	RGMII1_TXD2	PF.12	MFP8
	RGMII1_TXD3	PF.13	MFP8
RMII1	RMII1_MDC	PF.0	MFP9
	RMII1_MDIO	PF.1	MFP9
	RMII1_TXEN	PF.2	MFP9
	RMII1_TXD0	PF.3	MFP9
	RMII1_TXD1	PF.4	MFP9
	RMII1_REFCLK	PF.5	MFP9
	RMII1_CRSDV	PF.6	MFP9
	RMII1_RXD0	PF.7	MFP9
	RMII1_RXD1	PF.8	MFP9
	RMII1_RXERR	PF.9	MFP9

6.34.5 Functional Description

6.34.5.1 IEEE 1588-2002 Timestamps

The IEEE 1588-2002 standard defines a protocol, Precision Time Protocol (PTP), which enables precise synchronization of clocks in measurement and control systems implemented with technologies such as network communication, local computing, and distributed objects. The PTP applies to systems communicating by local area networks supporting multicast messaging, including (but not limited to) Ethernet. This protocol enables heterogeneous systems that include clocks of varying inherent precision, resolution, and stability to synchronize. The protocol supports system-wide synchronization accuracy in the sub-microsecond range with minimal network and local clock computing resources.

The PTP is transported over UDP/IP. The system or network is classified into master and slave nodes for distributing the timing and clock information. Figure 6.34-6 shows the process that PTP uses for synchronizing a slave node to a master node by exchanging PTP messages.

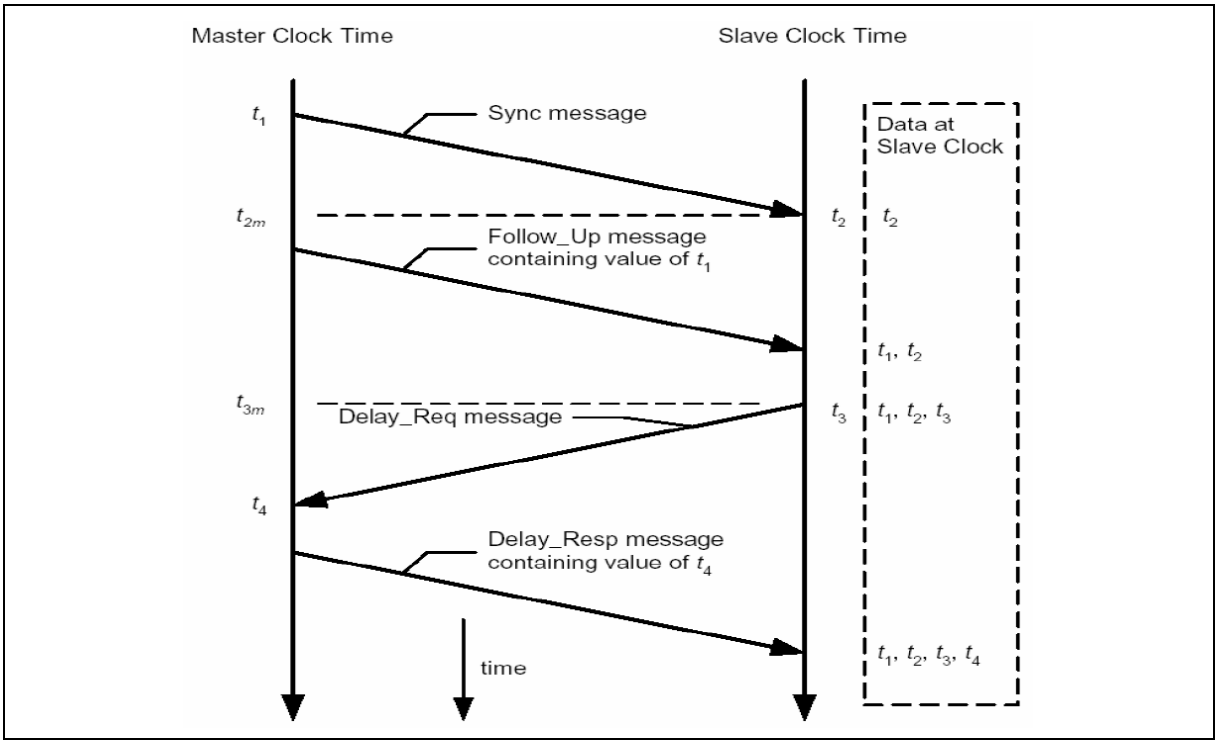


Figure 6.34-6 Networked Time Synchronization

As shown in Figure 6.34-6, the PTP uses the following process:

1. The master broadcasts the PTP Sync messages to all its nodes. The Sync message contains the master's reference time information. The time at which this message leaves the master's system is t1. This time must be captured, for Ethernet ports, at GMII or MII.
2. The slave receives the Sync message and also captures the exact time, t2, using its timing reference.
3. The master sends a Follow_up message to the slave, which contains t1 information for later use.
4. The slave sends a Delay_Req message to the master, noting the exact time, t3, at which this frame leaves the GMII/MII.
5. The master receives the message, capturing the exact time, t4, at which it enters its system.
6. The master sends the t4 information to the slave in the Delay_Resp message.
7. The slave uses the four values of t1, t2, t3, and t4 to synchronize its local timing reference to the master's timing reference.

Most of the PTP implementation is done in the software above the UDP layer. However, the hardware support is required to capture the exact time when specific PTP packets enter or leave the Ethernet port at the GMII/MII. This timing information must be captured and returned to the software for the proper implementation of PTP with high accuracy.

Reference Timing Source

To get a snapshot of the time, the MAC requires a reference time in 64-bit format as defined in the IEEE 1588 specification. The GMAC provides the following two options for using the reference timing source in a node:

- External Timestamp Input Option

  This option takes an external 64-bit timing reference and its clock as input. The clock input

is used to synchronize the timing reference to the MAC clock domain. The 64-bit timing reference is split into the following two 32-bit signals:

- Upper 32-bits providing the time in seconds
- Lower 32-bits providing the time in nanoseconds

This timing reference is used to timestamp the frames.

- Internal Reference Time Option

  This option takes only the reference clock input and uses it to generate the Reference time (also called the System Time) internally and capture timestamps. The generation, update, and modification of the System Time are described in “System Time Register Module”.

System Time Register Module

The System Time Generator module is optional and is not available if external time updating is enabled. The 64-bit time is maintained in this module and updated using the input reference clock (clk_ptp_ref_i).

This time is the source for taking snapshots (timestamps) of Ethernet frames being transmitted or received at the GMII.

The System Time counter can be initialized or corrected using the coarse correction method. In this method, the initial value or the offset value is written to the Timestamp Update register (“IEEE 1588 Timestamp Registers”). For initialization, the System Time counter is written with the value in the Timestamp Update registers, while for system time correction, the offset value is added to or subtracted from the system time.

In the fine correction method, a slave clock’s (clk_ptp_ref_i) frequency drift with respect to the master clock (as defined in IEEE 1588) is corrected over a period of time instead of in one clock, as in coarse correction. This helps maintain linear time and does not introduce drastic changes (or a large jitter) in the reference time between PTP Sync message intervals. In this method, an accumulator sums up the contents of the Addend register, as shown in Figure 6.34-7. The arithmetic carry that the accumulator generates is used as a pulse to increment the system time counter. The accumulator and the addend are 32-bit registers. Here, the accumulator acts as a high-precision frequency multiplier or divider.

This algorithm is depicted in Figure 6.34-7.

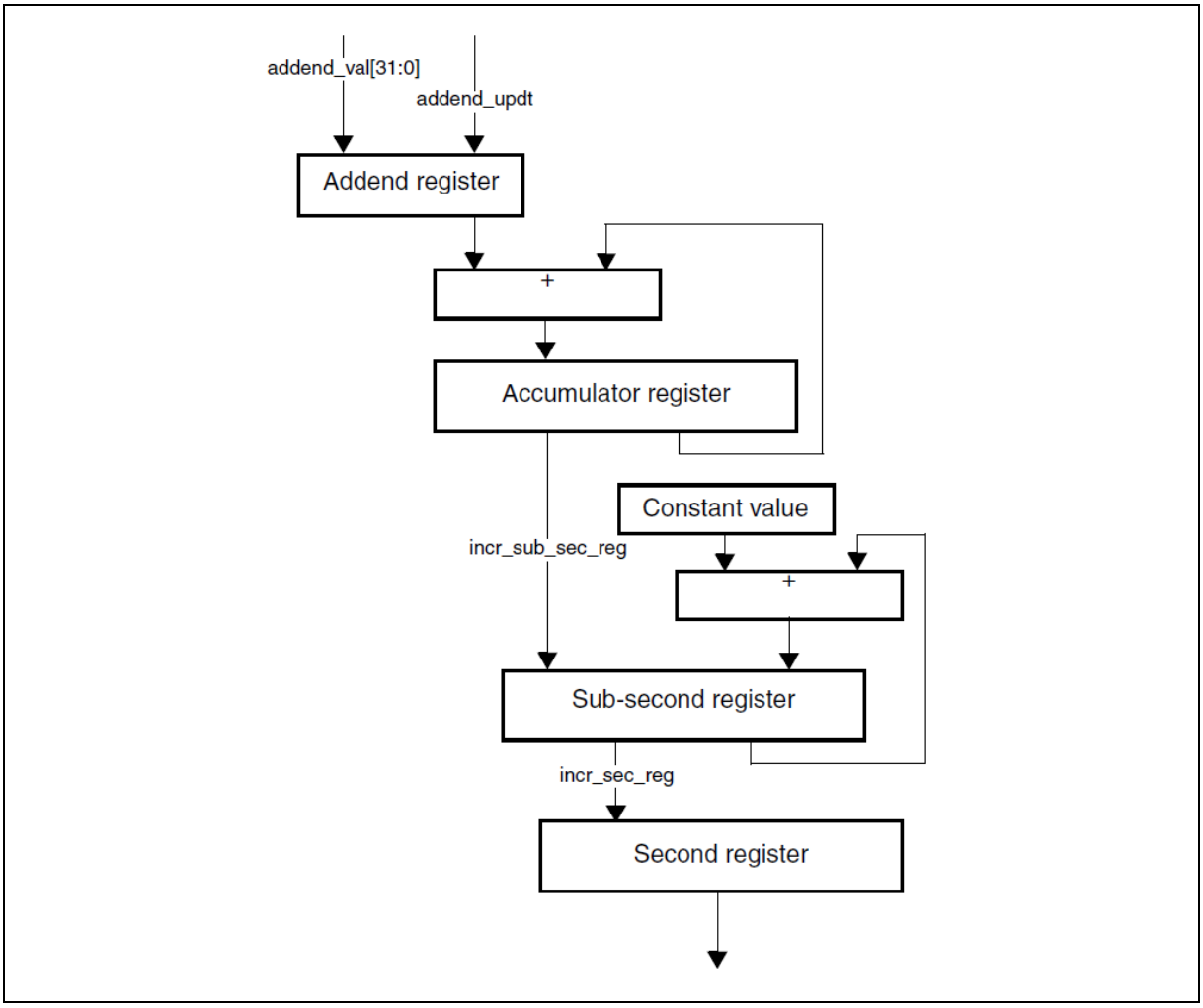


Figure 6.34-7 System Time Update Using Fine Method

The System Time Update logic requires a 50- MHz clock frequency to achieve 20-ns accuracy. The frequency division is the ratio of the reference clock frequency to the required clock frequency. Hence, if the reference clock (clk_ptp_ref_i) is, for example, 66 MHz, this ratio is calculated as 66 MHz / 50 MHz = 1.32. Hence, the default addend value to be set in the register is 2^{32} / 1.32, 0xC1F07C1F.

If the reference clock drifts lower, to 65 MHz for example, the ratio is 65 / 50, or 1.3 and the value to set in the addend register is 2^{32} / 1.30, or 0xC4EC4EC4. If the clock drifts higher, to 67 MHz for example, the addend register must be set to 0xBF0B7672. When the clock drift is nil, the default addend value of 0xC1F07C1F (2^{32} / 1.32) must be programmed.

In Figure 6.34-7, the constant value used to accumulate the sub-second register is decimal 43, which achieves an accuracy of 20 ns in the system time (in other words, it is incremented in 20-ns steps).

The software must calculate the drift in frequency based on the Sync messages and update the Addend register accordingly.

Initially, the slave clock is set with $FreqCompensationValue_0$ in the Addend register. This value is as follows:

$$FreqCompensationValue_0 = 2^{32} / FreqDivisionRatio$$

If MasterToSlaveDelay is initially assumed to be the same for consecutive Sync messages, the algorithm described below must be applied. After a few Sync cycles, frequency lock occurs. The slave clock can then determine a precise MasterToSlaveDelay value and re-synchronize with the master using the new

value.

The algorithm is as follows:

- At time $MasterSyncTime_n$ the master sends the slave clock a Sync message. The slave receives this message when its local clock is $SlaveClockTime_n$ and computes $MasterClockTime_n$ as:

  $MasterClockTime_n = MasterSyncTime_n + MasterToSlaveDelay_n$

- The master clock count for current Sync cycle, $MasterClockCount_n$ is given by:

  $MasterClockCount_n = MasterClockTime_n – MasterClockTime_{n-1}$ (assuming that MasterToSlaveDelay is the same for Sync cycles n and n – 1)

- The slave clock count for current Sync cycle, $SlaveClockCount_n$ is given by:

  $SlaveClockCount_n = SlaveClockTime_n – SlaveClockTime_{n-1}$

- The difference between master and slave clock counts for current Sync cycle, $ClockDiffCount_n$ is given by:

  $ClockDiffCount_n = MasterClockCount_n – SlaveClockCount_n$

- The frequency-scaling factor for slave clock, FreqScaleFactorn is given by:

  $FreqScaleFactor_n = (MasterClockCount_n + ClockDiffCount_n) / SlaveClockCount_n$

- The frequency compensation value for Addend register, $FreqCompensationValue_n$ is given by:

  $FreqCompensationValue_n = FreqScaleFactor_n * FreqCompensationValue_{n-1}$

In theory, this algorithm achieves lock in one Sync cycle; however, it may take several cycles, because of changing network propagation delays and operating conditions.

This algorithm is self-correcting: if for any reason the slave clock is initially set to a value from the master that is incorrect, the algorithm corrects it at the cost of more Sync cycles.

Transmit Path Functions

The MAC captures a timestamp when the Start Frame Delimiter (SFD) of a frame is sent on GMII/MII. The frames for which you want to capture timestamps are controllable on a per-frame basis. In other words, each transmit frame can be marked to indicate whether a timestamp should be captured for that frame.

The MAC does not process the transmitted frames to identify the PTP frames. You need to specify the frames for which you want to capture timestamps. To return the captured timestamps to the application, the MAC uses a separate 64-bit bus. The MAC gives the timestamp, along with the Transmit status of the frame, on this bus.

You can use the control bits in the transmit descriptor (see “Descriptor Format With IEEE 1588-2005 Timestamps Enabled”). The MAC returns the timestamp to the software inside the corresponding transmit descriptor, thus connecting the timestamp automatically to the specific PTP frame. The 64-bit timestamp information is written to the TDES2 and TDES3 fields. The TDES2 field holds the 32 least significant bits of the timestamp, except as described in “Transmit Timestamp Field”.

When the alternate (enhanced) descriptor is selected, the MAC writes the 64-bit timestamp in TDES6 and TDES7, respectively.

Receive Path Functions

The MAC captures the timestamp of all frames received on the GMII or MII interface.

The DMA returns the timestamp to the software in the corresponding receive descriptor. The 64-bit timestamp information is written back to the RDES2 and RDES3 fields. The RDES2 holds the 32 least significant bits of the timestamp, except as mentioned in “Receive Timestamp”. The timestamp is written

only to that receive descriptor for which the Last Descriptor status field has been set to 1 (the EOF marker). When the timestamp is not available (for example, because of an Rx FIFO overflow), an all-ones pattern is written to the descriptors (RDES2 and RDES3), indicating that timestamp is not correct. If the software uses a control register bit to disable timestamping, the DMA does not alter RDES2 or RDES3.

When the alternate (enhanced) descriptor is selected, the 64-bit timestamp is written in RDES6 and RDES7, respectively. RDES0[7] indicates whether the timestamp is updated in RDES6 and RDES77 or not.

Timestamp Error Margin

According to the IEEE 1588 specifications, a timestamp must be captured at the SFD of the transmitted and received frames at the GMII or MII interface. Because the reference timing source (the PTP clock, clk_ptp_ref_i) is taken as different from the GMII or MII clocks, a small error margin is introduced, because of the transfer of information across asynchronous clock domains.

In the transmit path, the captured and reported timestamp has a maximum error margin of 2 PTP clocks. It means that the captured timestamp has the reference timing source value that is given within 2 clocks after the SFD has been transmitted on the GMII.

Similarly, in the receive path, the error margin is 3 GMII or MII clocks, plus up to 2 PTP clocks. You can ignore the error margin because of the three GMII or MII clocks by assuming that this constant delay is present in the system (or link) before the SFD data reaches the GMII or MII interface of MAC.

Frequency Range of Reference Timing Clock

The timestamp information is transferred across asynchronous clock domains, that is, from MAC clock domain to application clock domain. Therefore, a minimum delay is required between two consecutive timestamp captures. This delay is 4 clock cycles of GMII or MII and 3 clock cycles of PTP clocks. If the delay between two timestamp captures is less than this delay, the MAC does not take a timestamp snapshot for the second frame.

The maximum PTP clock frequency is limited by the maximum resolution of the reference time (1 ns resulting in 1 GHz) and the timing constraints achievable for logic operating on the PTP clock. In addition, the resolution, or granularity, of the reference time source determines the accuracy of the synchronization. Therefore, a higher PTP clock frequency gives better system performance.

The minimum PTP clock frequency depends on the time required between two consecutive SFD bytes and the time taken for synchronizing the time to the GMII or MII clock domain. This relationship is given by the following equation:

```
3*PTP clock period + 4*GMII/MII clock period <= Minimum gap between two SFDs
```

Because the GMII or MII clock frequency is fixed by IEEE specification, the minimum PTP clock frequency required for proper operation depends upon the operating mode and operating speed of the MAC as shown in Table 6.34-1.

Mode	Minimum gap between two SFDs	Minimum PTP frequency with external timestamp input	Minimum PTP frequency with internal timestamp
10-Mbps full duplex	168 MII clocks (128 clocks for a 64-byte frame + 24 clocks of min IFG + 16 clocks of preamble)	~45 kHz	5 MHz
10-Mbps half duplex	48 MII clocks (8 clocks for a JAM pattern sent just after SFD because of collision + 24 IFG + 16 preamble)	170 kHz	5 MHz
100-Mbps full duplex	168 MII clocks (128 clocks for a 64-byte frame + 24 clocks	~0.5 MHz	5 MHz

	of min IFG + 16 clocks of preamble)		
100-Mbps half duplex	48 MII clocks (8 clocks for a JAM pattern sent just after SFD because of collision + 24 IFG + 16 preamble)	4.55 MHz	5 MHz
1000-Mbps full duplex	84 GMII clocks (64 clocks for a 64-byte frame + 12 clocks of min IFG + 8 clocks of preamble)	~4.7 MHz	5 MHz
100-Mbps half duplex	24 GMII clocks (4 clocks for a JAM pattern sent just after SFD because of collision + 12 IFG + 8 preamble)	18.75 MHz	18.75 MHz

Table 6.34-1 Minimum PTP Clock Frequency Example

When IEEE1588 timestamping is enabled with internal timestamp, use PTP clock frequency which is greater than 5 MHz. This is because the 8-bit Register 449 (Sub- Second Increment Register) limits the minimum PTP frequency that can be used to ~4 MHz.

6.34.5.2 IEEE 1588-2008 Advanced Timestamps

In addition to the basic timestamp features mentioned in IEEE 1588-2002 Timestamps, the GMAC supports the following advanced timestamp features defined in the IEEE 1588-2008 standard.

- Supports the IEEE 1588-2008 (version 2) timestamp format.
- Provides an option to take snapshot of all frames or only PTP type frames.
- Provides an option to take snapshot of only event messages.
- Provides an option to take the snapshot based on the clock type: ordinary, boundary, end-to-end, and peer-to-peer.
- Provides an option to select the node to be a master or slave for ordinary and boundary clock.
- Identifies the PTP message type, version, and PTP payload in frames sent directly over Ethernet and sends the status.
- Provides an option to measure sub-second time in digital or binary format.

Peer-to-Peer PTP Transparent Clock (P2P TC) Message Support

The IEEE 1588-2008 version supports Peer-to-Peer PTP (Pdelay) message in addition to SYNC, Delay Request, Follow-up, and Delay Response messages. Figure 6.34-8 shows the method to calculate the propagation delay in clocks supporting peer-to-peer path correction.

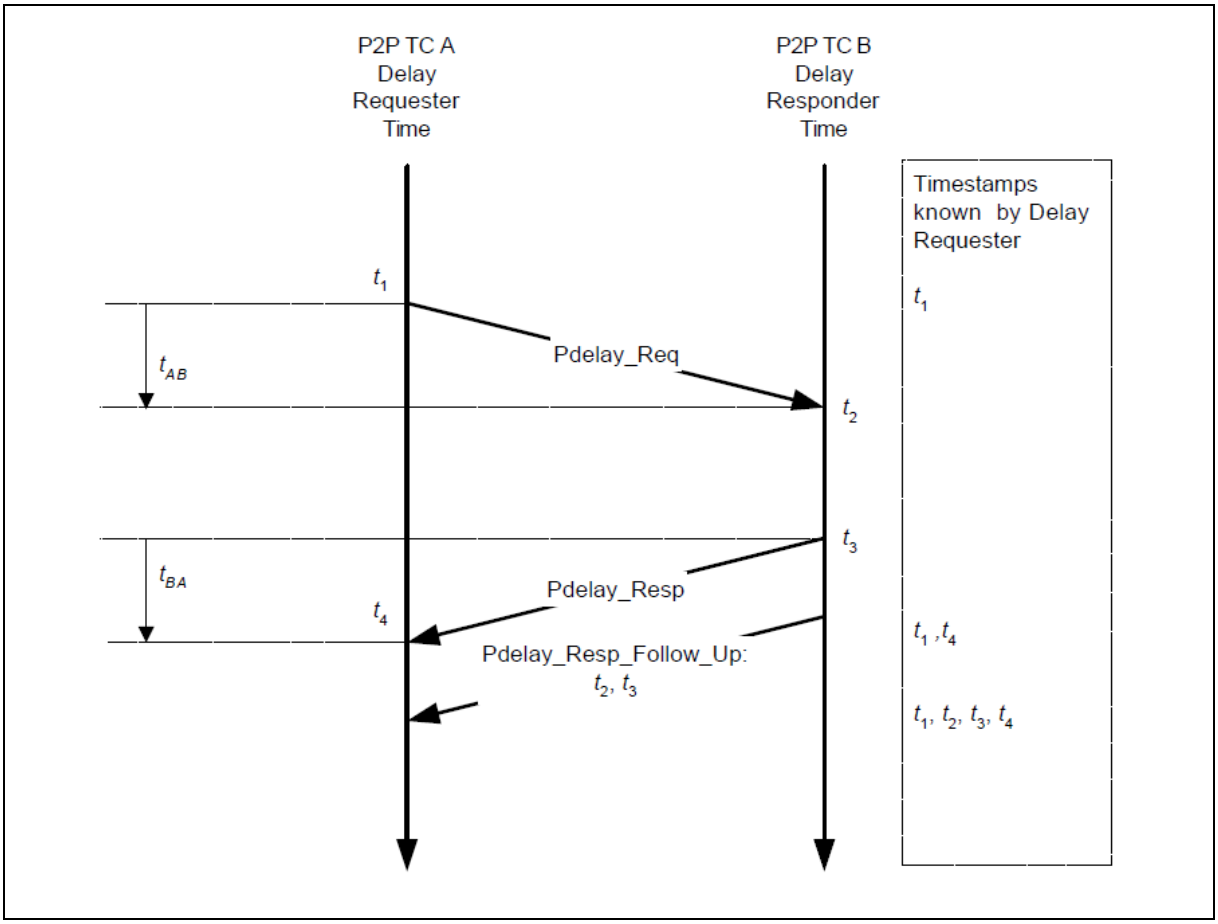


Figure 6.34-8 Propagation Delay Calculation in Clocks Supporting Peer-to-Peer Path Correction

As shown in Figure 6.34-8, the propagation delay is calculated in the following way:

1. Port-1 issues a Pdelay_Req message and generates a timestamp, t1, for the Pdelay_Req message.
2. Port-2 receives the Pdelay_Req message and generates a timestamp, t2, for this message.
3. Port-2 returns a Pdelay_Resp message and generates a timestamp, t3, for this message.
4. To minimize errors because of any frequency offset between the two ports, Port-2 returns the Pdelay_Resp message as quickly as possible after the receipt of the Pdelay_Req message. The Port-2 returns any one of the following:
   - The difference between the timestamps t2 and t3 in the Pdelay_Resp message.
   - The difference between the timestamps t2 and t3 in the Pdelay_Resp_Follow_Up message.
   - The timestamps t2 and t3 in the Pdelay_Resp and Pdelay_Resp_Follow_Up messages respectively.
5. Port-1 generates a timestamp, t4, on receiving the Pdelay_Resp message.
6. Port-1 uses all four timestamps to compute the mean link delay.

Clock Types

The GMAC supports the following clock types defined in the IEEE 1588-2008 standard:

- Ordinary Clock

- Boundary Clock
- End-to-End Transparent Clock
- Peer-to-Peer Transparent Clock

Ordinary Clock

The ordinary clock in a domain supports a single copy of the protocol. The ordinary clock has a single PTP state and a single physical port. In typical industrial automation applications, an ordinary clock is associated with an application device such as a sensor or an actuator. In telecom applications, the ordinary clock can be associated with a timing demarcation device.

The ordinary clock can be a grandmaster or a slave clock. It supports the following features:

- Sends and receives PTP messages. The timestamp snapshot can be controlled as described in "Register 448 (Timestamp Control Register)".
- Maintains the data sets such as timestamp values.

Table 6.34-2 shows the messages for which you can take the timestamp snapshot on the receive side for Master and slave nodes.

Master	Slave
Delay_Req	SYNC

Table 6.34-2 Ordinary Clock: PTP Messages for Snapshot

For an ordinary clock, you can take the snapshot of either of the following PTP message types: version 1 or version 2. You cannot take the snapshots for both PTP message types. You can take the snapshot by setting the control bit (TSVER2ENA) and selecting the snapshot mode in Register 448 (Timestamp Control Register).

Boundary Clock

The boundary clock typically has several physical ports communicating with the network. The messages related to synchronization, master-slave hierarchy, and signaling terminate in the protocol engine of the boundary clock and are not forwarded. The PTP message type status given by the MAC (see "Receive Path Functions") helps you to identify the type of message and take appropriate action.

The boundary clock is similar to the ordinary clock except for the following features:

- The clock data sets are common to all ports of the boundary clock
- The local clock is common to all ports of the boundary clock.

Therefore, the features of the ordinary clock are also applicable to the boundary clock.

End-to-End Transparent Clock

The end-to-end transparent clock supports the end-to-end delay measurement mechanism between slave clocks and the master clock. The end-to-end transparent clock forwards all messages like normal bridge, router, or repeater. The residence time of a PTP packet is the time taken by the PTP packet from the Ingress port to the Egress port.

The residence time of a SYNC packet inside the end-to-end transparent clock is updated in the correction field of the associated Follow_Up PTP packet before it is transmitted. Similarly, the residence time of a Delay_Req packet inside the end-to-end transparent clock is updated in the correction field of the associated Delay_Resp PTP packet before it is transmitted. Therefore, the snapshot needs to be taken at both Ingress and Egress ports only for the messages mentioned in Table 6.34-3. You can take the snapshot by setting the snapshot select bits (SNAPTYPSEL) to 10 in Register 448 (Timestamp Control Register).

PTP Messages

SYNC
Delay_Req

Table 6.34-3 End to End Transparent Clock: PTP Messages for Snapshot

Peer-to-Peer Transparent Clock

The peer-to-peer transparent clock differs from the end-to-end transparent clock in the way it corrects and handles the PTP timing messages. In all other aspects, it is identical to the end-to-end transparent clock.

In the peer-to-peer transparent clock, the computation of the link delay is based on an exchange of Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up messages with the link peer. The residence time of the Pdelay_Req and the associated Pdelay_resp packets is added and inserted into the correction field of the associated Pdely_Resp_Followup packet. Therefore, support for taking snapshot for the event messages related to Pdelay is added as shown in Table 6.34-4.

PTP Messages
SYNC
Pdelay_Req
Pdelay_Resp

Table 6.34-4 Peer-to-Peer Transparent Clock: PTP Messages for Snapshot

You can take the snapshot by setting the snapshot select bits (SNAPTYPESEL) to 11 in Register 448 (Timestamp Control Register).

PTP Processing and Control

Table 6.34-5 shows the common message header for the PTP messages. This format is taken from IEEE standard 1588-2008 (Revision of IEEE Std. 1588-2002).

BITS 7 6 5 4 3 2 1 0		OCTETS	OFFSET
transportSpecific	messageType	1	0
Reserved	versionPTP	1	1
messageLength		2	2
domainNumber		1	4
Reserved		1	5
flagField		2	6
correctionField		8	8
Reserved		4	16
sourcePortIdentity		10	20
sequenceId		2	30
controlField (*)		1	32

logMessageInterval	1	33

(*) – controlField is used in version 1. In version 2, messageType field is used for detecting different message types.

Table 6.34-5 Message Format Defined in IEEE 1588-2008

There are some fields in the Ethernet payload that you can use to detect the PTP packet type and control the snapshot to be taken. These fields are different for the following PTP frames:

- PTP Frames Over IPv4
- PTP Frames Over IPv6
- PTP Frames Over Ethernet

PTP Frames Over IPv4

Table 6.34-6 provides information about the fields that are matched to control snapshot for the PTP packets sent over UDP over IPv4 for IEEE 1588 version 1 and 2. The octet positions for the tagged frames are offset by 4. This is based on Annex D of IEEE 1588-2008 standard and the message format defined in Table 6.34-5.

Field matched	Octet position	Matched value	Description
MAC Frame Type	12, 13	0x0800	IPv4 datagram
IP version and Header Length	14	0x45	IP version is IPv4
Layer 4 Protocol	23	0x11	UDP
IP Multicast Address (IEEE 1588 version 1)	30, 31, 32, 33	0xE0, 0x00, 0x01, 0x81 (or 0x82 or 0x83 or 0x84)	Multicast IPv4 addresses allowed. 224.0.1.129 224.0.1.130 224.0.1.131 224.0.1.132
IP Multicast Address (IEEE 1588 version 2)	30, 31, 32, 33	0xE0, 0x00, 0x01, 0x81 (Hex) 0xE0, 0x00, 0x00, 0x6B (Hex)	PTP-Primary multicast address: 224.0.1.129 PTP-Pdelay multicast address: 224.0.0.107
UDP Destination Port	36, 37	0x013F, 0x0140	0x013F – PTP event message (*) 0x0140 – PTP general messages
PTP Control Field (IEEE version 1)	74	0x00, 0x01, 0x02, 0x03, 0x04	0x00 = SYNC 0x01 = Delay_Req 0x02 = Follow_Up 0x03 = Delay_Resp 0x04 = Management
PTP Message Type Field (IEEE version 2)	42 (nibble)	0x0, 0x1, 0x2, 0x3, 0x8, 0x9, 0xB, 0xC, 0xD	0x0 = SYNC 0x1 = Delay_Req 0x2 = Pdelay_Req 0x3 = Pdelay_Resp 0x8 = Follow_Up 0x9 = Delay_Resp 0xA = Pdelay_Resp_Follow_Up 0xB = Announce

			0xC = Signaling 0xD = Management
PTP Version	43 (nibble)	0x1 or 0x2	0x1 = Supports PTP version 1 0x2 = Supports PTP version 2

(*) PTP event messages are SYNC, Delay_Req (IEEE 1588 version 1 and 2) or Pdelay_Req, Pdelay_Resp (IEEE 1588 version 2 only).

Table 6.34-6 IPv4-UDP PTP Frame Fields Required for Control and Status

PTP Frames Over IPv6

Table 6.34-7 provides information about the fields that are matched to control the snapshots for the PTP packets sent over UDP over IPv6 for IEEE 1588 version 1 and 2. The octet positions for the tagged frames are offset by 4. This is based on Annex D of IEEE 1588-2008 standard and the message format defined in Table 6.34-5.

Field matched	Octet position	Matched value	Description
MAC Frame Type	12, 13	0x86DD	IP datagram
IP Version	14 (Bits [7: 4])	0x6	IP version is IPv6
Layer 4 Protocol	20 (*)	0x11	UDP
PTP Multicast Address	38 – 53	FF0x: 0: 0: 0: 0: 0: 0: 181 (Hex) FF02: 0: 0: 0: 0: 0: 0: 6B (Hex)	PTP-Primary multicast address: FF0x: 0: 0: 0: 0: 0: 0: 0: 0: 181 (Hex) PTP-Pdelay multicast address: FF02: 0: 0: 0: 0: 0: 0: 0: 0: 6B (Hex)
UDP Destination Port	56, 57 (*)	0x013F, 0x140	0x013F = PTP event message 0x0140 = PTP general messages
PTP Control Field (IEEE 1588 version 1)	94 (*)	0x00, 0x01, 0x02, 0x03, 0x04	0x00 = SYNC 0x01 = Delay_Req 0x02 = Follow_Up 0x03 = Delay_Resp 0x04 = Management (version1)
PTP Message Type Field (IEEE version 2)	62 (*) (nibble)	0x0, 0x1, 0x2, 0x3, 0x8, 0x9, 0xB, 0xC, 0xD	0x0 = SYNC 0x1 = Delay_Req 0x2 = Pdelay_Req 0x3 = Pdelay_Resp 0x8 = Follow_Up 0x9 = Delay_Resp 0xA = Pdelay_Resp_Follow_Up 0xB = Announce 0xC = Signaling 0xD = Management
PTP Version	63 (nibble)	0x1 or 0x2	0x1 = Supports PTP version 1 0x2 = Supports PTP version 2

(*) The Extension Header is not defined for PTP packets.

Table 6.34-7 IPv6-UDP PTP Frame Fields Required for Control and Status

PTP Frames Over Ethernet

Table 6.34-8 provides information about the fields that are matched to control the snapshots for the PTP

packets sent over Ethernet for IEEE 1588 version 1 and 2. The octet positions for the tagged frames are offset by 4. This is based on Annex D of the IEEE 1588-2008 standard and the message format defined in Table 6.34-5.

Field matched	Octet position	Matched value	Description
MAC Destination Multicast Address[a]	0-5	01-1B-19-00-00-00 01-80-C2-00-00-0E	All PTP messages can use any of the following multicast addresses[b]: ■01-1B-19-00-00-00 ■01-80-C2-00-00-0E[c]
MAC Frame Type	12, 13	0x88F7	PTP Ethernet frame
PTP Control Field (IEEE version 1)	46	0x00, 0x01, 0x02, 0x03, 0x04	0x00 = SYNC 0x01 = Delay_Req 0x02 = Follow_Up 0x03 = Delay_Resp 0x04 = Management
PTP Message Type Field (IEEE version 2)	14 (nibble)	0x0, 0x1, 0x2, 0x3, 0x8, 0x9, 0xB, 0xC, 0xD	0x0 = SYNC 0x1 = Delay_Req 0x2 = Pdelay_Req 0x3 = Pdelay_Resp 0x8 = Follow_Up 0x9 = Delay_Resp 0xA = Pdelay_Resp_Follow_Up 0xB = Announce 0xC = Signaling 0xD = Management
PTP Version	15 (nibble)	0x1 or 0x2	0x1 = Supports PTP version 1 0x2 = Supports PTP version 2

a. The address match of destination addresses (DA), programmed in MAC address 0 to 31, is used if Bit 18 (TSENMACADDR: Enable MAC address for PTP frame filtering) of the Timestamp Control register is set.

b. IEEE standard 1588-2008, Annex F

c. The MAC does not consider the PTP version 1 messages with Peer delay multicast address (01-80-C2-00-00-0E) as valid PTP messages.

Table 6.34-8 Ethernet PTP Frame Fields Required for Control And Status

Reference Timing Source

The MAC supports the following reference timing source features defined in the IEEE 1588-2008 standard:

- 48-Bit Seconds Field
- Fixed Pulse-Per-Second Output
- Flexible Pulse-Per-Second Output

48-Bit Seconds Field

The MAC supports 80-bit timestamp. The timestamp has the following fields:

- UInteger48 secondsField

  The seconds field is the integer portion of the timestamp in units of seconds and is 48-bits wide. For example, 2.000000001 seconds are represented as secondsField = 0x0000_0000_0002.

- UInteger32 nanosecondsField

  The nanoseconds field is the fractional portion of the timestamp in units of nanoseconds. For example, 2.000000001 nanoseconds are represented as nanoSeconds = 0x0000_0001.

  The nanoseconds field supports the following two modes:

  - Digital rollover mode: In digital rollover mode, the maximum value in the nanoseconds field is 0x3B9A_C9FF, that is, (10e9-1) nanoseconds.
  - Binary rollover mode: In binary rollover mode, the nanoseconds field rolls over and increments the seconds field after value 0x7FFF_FFFF. Accuracy is ~0.466 ns per bit.

  You can set these modes by using Bit 9 (TSCTRLSSR) of Register 448 (Timestamp Control Register).

When you select the advanced timestamp feature, the timestamp maintained in the MAC is still 64-bit wide. The overflow to the upper 16-bits of seconds register happens once in 130 years. You can read the values of the upper 16-bits of the seconds field from the CSR register.

Fixed Pulse-Per-Second Output

The GMAC supports the pulse-per-second (PPS) output that is given to indicate 1 second interval (default). You can change the frequency of the PPS output by setting Bits[3: 0], PPSCTRL, in Register 459 (PPS Control Register).

Flexible Pulse-Per-Second Output

The GMAC also provides the flexibility to program the start or stop time, width, and interval of the pulse generated on the ptp_pps_o output.

The GMAC provides the following features with the Flexible PPS output option:

- Supports programming the start or stop time in terms of system time.
- Supports programming the start point of the single pulse and start and stop points of the pulse train in terms of 64-bit system time. The existing Target Time registers (register 455 and 456) are used to program the start and stop time.
- Supports programming the stop time in advance, that is, you can program the stop time before the actual start time has elapsed.
- Supports programming the width, between the rising edge and corresponding falling edge of PPS signal output, in terms of number of units of sub-second increment value programmed in Register 449 (Sub-Second Increment Register). You can program the width of pulse from 1 to 232-1 units of sub-second increment value.
- Supports programming the interval, between the rising edges of PPS signal, in terms of number of units of sub-second increment value. You can program the interval between pulses from 1 to 232-1 units of sub-second increment value.
- Provides the option to cancel the programmed PPS start or stop request.
- Indicates error if the start or stop time being programmed has already elapsed.

The PTP Reference clock mentioned in the following sections is the clock at which the system time gets updated. When Bit 1 (Timestamp Fine or Coarse Update) of Register 448 (Timestamp Control Register) is set to 0, this clock is similar to the clk_ptp_ref_i clock. In the Fine Correction mode, this is the clock tick at which the system time gets updated (using Bits 7: 0 in Register 449 (Sub-Second Increment Register), which corresponds to incr_sub_sec_reg shown in Figure 6.34-7).

Flexible Pulse-Per-Second Output: PPS Start or Stop Time

You can initially program the start time in the Target Time registers (455 and 456).

If required, you can again program the start or stop time but you can do it only after the earlier programmed value is synchronized to the PTP clock domain. Bit 31 of GMAC registers 456 indicates

that the synchronization is complete. This enables you to program the start or stop time in advance, even before the earlier stop or start time has elapsed.

The start or stop time should be programmed with advanced system time to ensure proper PPS signal output. If the application programs a start or stop time that has already elapsed, then the MAC sets an error status bit indicating the programming error. If enabled, the MAC also sets the Target Time Reached interrupt event. The application can cancel the start or stop request only if the corresponding start or stop time has not elapsed. If the time has elapsed, the cancel command has no effect.

Flexible Pulse-Per-Second Output: PPS Width and Interval

The PPS width and interval are programmed in terms of granularity of system time, that is, number of the units of sub-second increment value. For example, to have a PPS pulse width of 40ns and interval of 100ns, with the PTP reference clock of 50 MHz, you should program the width and interval to values 2 and 5 respectively. You can achieve smaller granularity by using a faster PTP reference clock.

Before giving the command to trigger a pulse or pulse train on the PPS output, you should program or update the interval and width of the PPS signal output.

Transmit Path Functions

The structure of the descriptor changes when you enable the advanced timestamp feature. The advanced timestamp feature is supported only through Alternate (Enhanced) descriptors format. The descriptor is 32-bytes long (8 DWORDS) and the snapshot of the timestamp is written in descriptor TDES6 and TDES7. For detailed information about descriptors, see “Alternate or Enhanced Descriptors”.

Receive Path Functions

When you select the advanced timestamp feature, the MAC processes the received frames to identify valid PTP frames. You can control the snapshot of the time, to be sent to the application, by using the following options of Register 448 (Timestamp Control Register).

- Enable snapshot for all frames.
- Enable snapshot for IEEE 1588 version 2 or version 1 timestamp.
- Enable snapshot for PTP frames transmitted directly over Ethernet or UDP-IP-Ethernet.
- Enable timestamp snapshot for the received frame for IPv4 or IPv6.
- Enable timestamp snapshot only for EVENT messages (SYNC, DELAY_REQ, PDELAY_REQ, or PDELAY_RESP).
- Enable the node to be a master or slave and select the snapshot type. This controls the type of messages for which snapshots are taken.

The DMA returns the timestamp to the software inside the corresponding Transmit and Receive Descriptor. The advanced timestamp feature is supported only with the 32-bytes long Alternate (Enhanced) descriptor. The extended status, containing the timestamp message status and the IPC status, is written in descriptor RDES4 and the snapshot of the timestamp is written in descriptors RDES6 and RDES7. For detailed information about descriptors, see “Alternate or Enhanced Descriptors”.

6.34.5.3 Energy Efficient Ethernet

Energy Efficient Ethernet (EEE) is an optional operational mode that enables the IEEE 802.3 Media Access Control (MAC) sublayer along with a family of Physical layers to operate in the Low-Power Idle (LPI) mode. The EEE operational mode supports the IEEE 802.3 MAC operation at 100 Mbps, 1000 Mbps. The GMAC supports the IEEE Std 802.3az-2010 for EEE.

The LPI mode allows power saving by switching off parts of the communication device functionality when there is no data to be transmitted and received. The systems on both sides of the link can disable some functionalities and save power during the periods of low-link utilization. The MAC controls whether the system should enter or exit the LPI mode and communicates this to the PHY.

The EEE specifies the capabilities negotiation methods that the link partners can use to determine whether EEE is supported and then select the set of parameters that common to both devices.

The EEE feature is not supported when the MAC is configured to use the RMII. You should activate the EEE mode only when the MAC is operating with RGMII interface.

According to the Energy Efficient Ethernet standard (IEEE 802.3az-2010), the LPI mode is supported only in the full-duplex mode. Therefore, you should not enable the LPI mode when the MAC Transmitter is configured for the half-duplex mode.

Transmit Path Functions

In the transmit path, the software must set Bit 16 (LPIEN) of Register 12 (LPI Control and Status Register) at offset 0x0030 to indicate to the MAC to stop transmission and initiate the LPI protocol. The MAC completes the transmission in progress, generates its transmission status, and then starts transmitting the LPI pattern instead of the IDLE pattern during Interframe gap (IFG).

According to the Energy Efficient Ethernet standard (IEEE 802.3az-2010), the PHY should not stop the TxCLK clock during the LPI state in the MII (10 or 100) mode. However, the MAC can stop the GTX_CLK clock during the LPI state in the GMII (1000) mode.

To make the PHY enter the LPI state, the MAC performs the following tasks:

1. De-asserts TX_EN.
2. Asserts TX_ER.
3. Sets TXD[3: 0] to 0x1 (for 100 Mbps) or TXD[7: 0] to 0x01 (for 1000 Mbps).

   Note: The MAC maintains the same state of the TX_EN, TX_ER, and TXD signals for the entire duration during which the PHY remains in the LPI state.
4. Updates the status (Bit 0 of Register 12 (LPI Control and Status Register) at offset 0x0030) and generates an interrupt.

To bring the PHY out of the LPI state, that is, when the software resets the LPIEN bit, the MAC performs the following tasks:

1. Stops transmitting the LPI pattern and starts transmitting the IDLE pattern.
2. Starts the LPI TW TIMER.

   The MAC cannot start the transmission until the wake-up time specified for the PHY expires. The auto-negotiated wake-up interval is programmed in Bits [15: 0], TWT, of Register 13 (LPI Timers Control Register) at offset 0x0034.
3. Updates the LPI exit status (Bit 1 of Register 12 (LPI Control and Status Register) at 0x0030) and generates an interrupt.

If the MAC is in the Tx LPI mode and the Tx clock is stopped, the application should not write to CSR registers that are synchronized to Tx clock domain. If the MAC is in the LPI mode and the host issues a soft reset or hard reset, the MAC transmitter comes out of the LPI mode.

Figure 6.34-9 shows the behavior of TX_EN, TX_ER, and TXD[3: 0] signals during the LPI mode transitions.

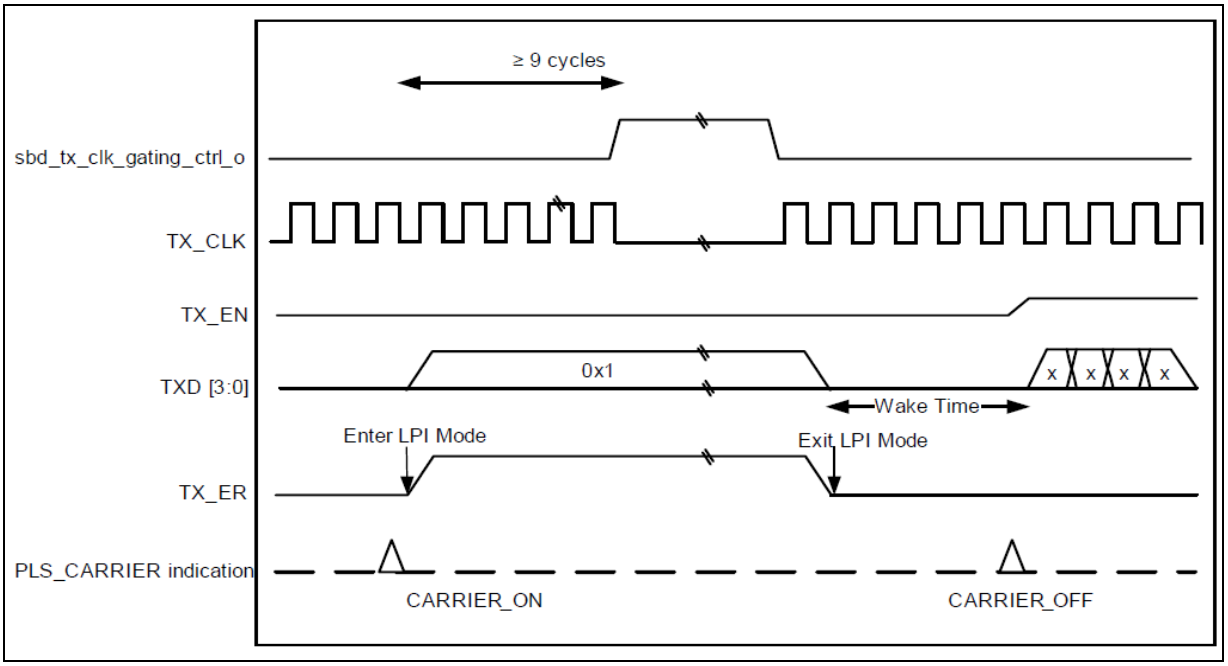


Figure 6.34-9 LPI Transitions (Transmit)

Receive Path Functions

In the receive path, when the PHY receives the signals from the link partner to enter into the LPI state, the PHY and MAC perform the following tasks:

1. The PHY asserts RX_ER.
2. The PHY sets RXD[7: 0] to 0x01.
3. The PHY de-asserts RX_DV.

   Note: The PHY maintains the same state of the RX_ER, RXD, and RX_DV signals for the entire duration during which it remains in the LPI state.
4. The MAC updates Bit 2 (RLPIEN) of Register 12 (LPI Control and Status Register) at offset 0x0030 and generates an interrupt immediately.

If the LPI pattern is detected for a very short duration (that is, less than 2 cycles of Rx clock), the MAC does not enter the Rx LPI mode.

If the duration between end of the current Rx LPI pattern and start of the next Rx LPI pattern, is very short (that is, less than 2 cycles of Rx clock), then the MAC exits and again enters the Rx LPI mode. The MAC does not give the Rx LPI Exit and Entry interrupts.

When the PHY receives signals from the link partner to exit the LPI state, the PHY and MAC perform the following tasks:

1. The PHY de-asserts RX_ER and returns to a normal inter-frame state.
2. The MAC updates Bit 3 (RLPIEX) of Register 12 (LPI Control and Status Register) at 0x0030 and generates an interrupt immediately. The sideband signal lpi_intr_o (synchronous to Rx clock) is also asserted.

Figure 6.34-10 shows the behavior of RX_ER, RX_DV, and RXD[3: 0] signals during the LPI mode transitions.

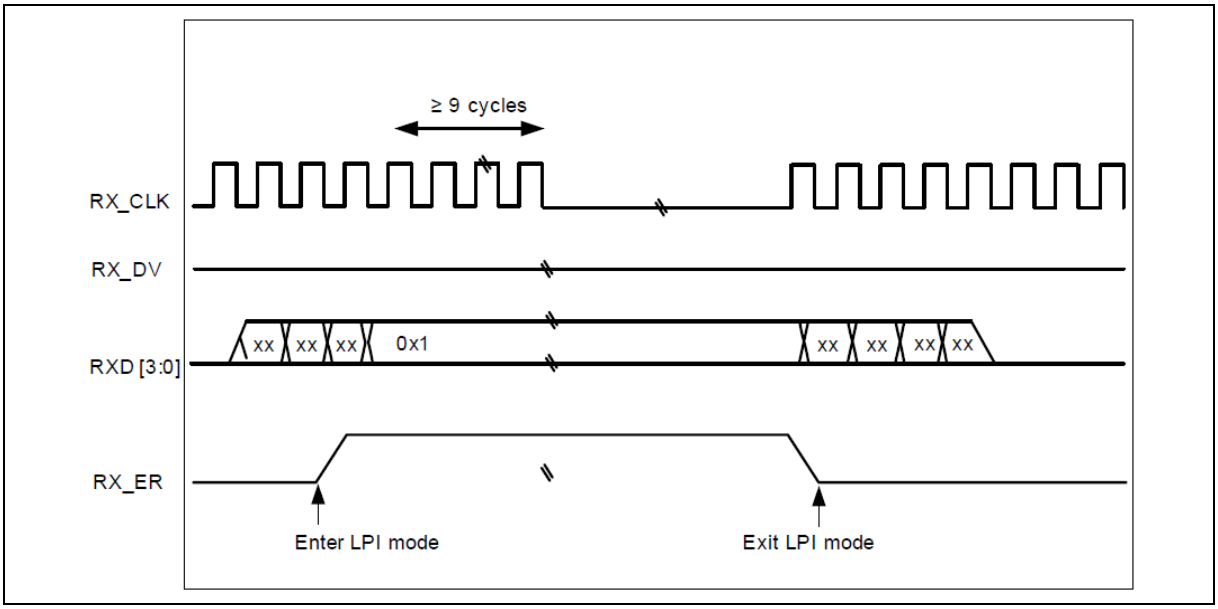


Figure 6.34-10 LPI Transitions (Receive)

If the RX_CLK_stoppable bit (in the PHY register written through MDIO) is asserted when the PHY is indicating LPI to the MAC, then the PHY may halt the RX_CLK at any time more than 9 clock cycles after the start of the LPI state as shown in Figure 6.34-10.

If the MAC is in the LPI mode and the host issues a soft reset or hard reset, the MAC receiver comes out of the LPI mode during reset. If the LPI pattern is still received after the reset is de-asserted, the MAC receiver again enters the LPI state.

If the RX clock is stopped in the RX LPI mode, the host should not write to the CSR registers that are being synchronized to the RX clock domain.

When the PHY sends the LPI pattern, the MAC automatically enters the LPI state. There is no software control to prevent the MAC from entering the LPI state.

LPI Timers

The transmitter maintains the following two timers that are loaded with the respective values (from Register 13 (LPI Timers Control Register) at offset 0x0034):

- LPI LS TIMER

  The LPI LS TIMER counts, in milliseconds, the time expired since the link status is up. You can enable the monitoring of Bit 3 (link status) of Register 54 (RGMII Control and Status Register) by setting Bit 18 (PLSEN) of Register 12 (LPI Control and Status Register).

  The link status is indicated to the MAC by Bit 3 (link status) of Register 54 (RGMII Control and Status Register) or the value programmed by the software in Bit 17 (PLS) of Register 12 (LPI Control and Status Register). If the link status is not available in Register 54 (RGMII Control and Status Register), the software should get the PHY link status by reading the PHY register and accordingly update Bit 17 (PLS).

  This timer is cleared every time the link goes down. It starts to increment when the link is up again and continues to increment until the value of the timer becomes equal to the terminal count. When the terminal count is reached, the timer remains at the same value as long as the link is up. The terminal count is the value programmed in Bits[25: 16], LST, of Register 13 (LPI Timers Control Register). The GMII interface does not assert the LPI pattern unless the terminal count is reached. This ensures a minimum time for which no LPI pattern is asserted after a link is established with the remote station. This period is

defined as 1 second in the IEEE standard 802.3-az, version D2.0. The LPI LS TIMER is 10-bit wide. Therefore, the software can program up to 1023 milliseconds.

- LPI TW TIMER

  The LPI TW TIMER counts, in microseconds, the time expired since the de-assertion of LPI. The terminal should be programmed in Bit[15: 0], TWT, of Register 13 (LPI Timers Control Register). The terminal count of the timer is the value of resolved Transmit TW that is the auto-negotiated time after which the MAC can resume the normal transmit operation. After exiting the LPI mode, the MAC resumes its normal operation after the TW timer reaches the terminal count.

  The MAC supports the LPI TW TIMER in units of microsecond. The LPI TW TIMER is 16-bit wide. Therefore, the software can program up to 65535 us.

Program Bit 17 (PLS) of Register 12 (LPI Control and Status Register) to 1'b0 before switching between the GMII and MII modes. This resets the internal timers. If the mode is changed after the LPI LS TIMER or LPI TW TIMER starts, the change in the TX clock frequency can result in incorrect timeout.

LPI Interrupt

The MAC generates the LPI interrupt when the Tx or Rx side enters or exits the LPI state. The LPI interrupt can be cleared by reading Register 12 (LPI Control and Status Register).

6.34.5.4 Source Address, VLAN, and CRC Insertion, Replacement, or Deletion

The GMAC supports the following functions for transmit frames:

- Source Address Insertion or Replacement
- VLAN Insertion, Replacement, or Deletion
- CRC Replacement

Source Address Insertion or Replacement

The software can use the SA insertion or replacement feature to instruct the MAC to do the following for transmit frames:

- Insert the content of the MAC Address Registers in the SA field.
- Replace the content of the SA field with the content of the MAC Address Registers.

The software can enable the SA insertion or replacement feature for all transmit frames or selective frames.

- To enable SA insertion or replacement feature for all frames, program Bits[30: 28] of Register 0 (MAC Configuration Register).
- Program the SA Insertion Control field (TDES1 Bits [31: 29]) in the first transmit descriptor of the frame. When Bit 31 of TDES1 is set, the SA Insertion Control field indicates insertion or replacement by MAC Address1 registers. When Bit 31 of TDES1 is reset, it indicates insertion or replacement by MAC Address 0 registers. If MAC Address1 Registers are not enabled, then MAC Address0 Registers are used for insertion or replacement irrespective of the value of the most significant bit of the SA Insertion Control field.

When SA insertion is enabled, the application should ensure that the frames that are sent to the MAC do not have the SA field. This is because the MAC does not check the presence of SA field in the transmit frame and just inserts the content of MAC Address Registers in the SA field. Similarly, when SA replacement is enabled, the application should ensure that the frames that are sent to the MAC have the SA field. The MAC just replaces the six bytes, following the Destination Address field in the transmit frame, with the content of the MAC Address Registers.

VLAN Insertion, Replacement, or Deletion

The software can use the VLAN insertion, replacement, or deletion feature to instruct the MAC to do the following for transmit frames:

- Insert or replace the VLAN Type field (C-VLAN or S-VLAN indicated by Bit 19 (CSVL) of Register 353 (VLAN Tag Inclusion or Replacement Register) and VLAN Tag field in the transmit frame with Bit [15: 0], VLT, of Register 353 (VLAN Tag Inclusion or Replacement Register).
- Delete the VLAN Type and VLAN Tag fields in transmit frame.

The software can enable the VLAN insertion, replacement, or deletion feature for all transmit frames or selective frames. To enable this function for all transmit frames, program Bits[17: 16] of Register 353 (VLAN Tag Inclusion or Replacement Register). To enable this function for selective frames, program the VLAN Insertion Control field (TDES0 Bits [19: 18]) in the first transmit descriptor of the frame.

When VLAN replacement or deletion is enabled, the MAC checks the presence of the VLAN Type field (0x8100 or 0x88a8), after the Destination address (DA) and SA fields, in the transmit frame. The replace or delete operation does not occur if the VLAN Type field is not detected in the two bytes following the DA and SA fields. However, when VLAN insertion is enabled, the MAC does not check the presence of VLAN Type field in the transmit frame and just inserts the VLAN Type and VLAN Tag fields.

CRC Replacement

The software can use the CRC replacement feature to instruct the MAC to replace the FCS field in the transmit frame with the CRC computed by the MAC. This feature works on per-frame basis. To enable the CRC replacement feature, program the CRC Replacement Control field (Bit 24 of Transmit Descriptor Word 0 (TDES0)) in the first transmit descriptor of the frame.

This feature is valid only when Disable CRC control (Bit 27 in TDES0) is enabled, that is, the FCS field is provided by the software in the transmit frame. If SA or VLAN insertion control is enabled, the MAC appends or replaces the FCS field with the computed CRC when Disable CRC Control is enabled or disabled, respectively.

Table 6.34-9 explains how CRC replacement is done based on the values of Bit 27 (DC) and Bit 24 (CRCR) of Transmit Descriptor Word 0 (TDES0).

Bit 27 (DC)	Bit 24 (CRCR)	Description
0	x	Append CRC When DC = 0, the MAC appends the computed CRC irrespective of the CRCR setting.
1	1	Replace CRC
1	0	No operation (User has appended the CRC)

Table 6.34-9 CRC Replacement Based on TDES0 Bit 27 and Bit 24

6.34.5.5 Checksum Offload Engine

Communication protocols such as TCP and UDP implement checksum fields, which help determine the integrity of data transmitted over a network. Because the most widespread use of Ethernet is to encapsulate TCP and UDP over IP datagrams, the MAC has an optional Checksum Offload Engine (COE) to support checksum calculation and insertion in the transmit path, and error detection in the receive path.

Transmit Checksum Offload Engine

The COE module supports two types of checksum calculation and insertion. The checksum engine can be controlled for each frame by setting the CIC bits (Bits 28: 27 of TDES1, described in “Transmit Descriptor 1 (TDES1)”).

The checksum for TCP, UDP, or ICMP is calculated over a complete frame, and then inserted into its corresponding header field. Because of this requirement, this function is enabled only when the Transmit

FIFO is configured for the store-and-forward mode (that is, when the TSF bit is set in DMA Register 6). If the GMAC is configured for Threshold (cut-through) mode, the Transmit COE is bypassed.

You must make sure that the Transmit FIFO is deep enough to store a complete frame before that frame is transferred to the MAC transmitter. The reason being that when space is not available to accept the programmed burst length of data, then the MTL Tx FIFO starts reading to avoid dead-lock. When reading starts, the COE fails and consequently all succeeding frames may get corrupted because of improper recovery. Therefore, you must enable the checksum insertion only in the frames that are less than the number of bytes in size (even in the store-and-forward mode).

See IETF specifications RFC 791, RFC 793, RFC 768, RFC 792, RFC 2460, and RFC 4443 for IPv4, TCP, UDP, ICMP, IPv6, and ICMPv6 packet header specifications, respectively.

IP Header Checksum Engine

In IPv4 datagrams, the integrity of the header fields is indicated by the 16-bit Header Checksum field (the eleventh and twelfth bytes of the IPv4 datagram). The COE detects an IPv4 datagram when the Ethernet frame's Type field has the value 0x0800 and the IP datagram's Version field has the value 0x4. The input frame's checksum field is ignored during calculation and replaced with the calculated value.

IPv6 headers do not have a checksum field. Therefore, the COE does not modify the IPv6 header fields. The result of this IP header checksum calculation is indicated by the IP Header Error status bit in the

Transmit status (Bit 16 in Table 6.34-15). This status bit is set whenever the values of the Ethernet Type field and the IP header's Version field are not consistent, or when the Ethernet frame does not have enough data, as indicated by the IP header Length field. In other words, this bit is set when an IP header error is asserted under the following circumstances:

- For IPv4 datagrams:
  - The received Ethernet type is 0x0800, but the IP header's Version field is not equal to 0x4.
  - The IPv4 Header Length field indicates a value less than 0x5 (20 bytes).
  - The total frame length is less than the value given in the IPv4 Header Length field.
- For IPv6 datagrams:
  - The Ethernet type is 0x86dd but the IP header Version field is not equal to 0x6.

If COE detects an IP header error, it still inserts an IPv4 header checksum if the Ethernet Type field indicates an IPv4 payload.

TCP/UDP/ICMP Checksum Engine

The TCP/UDP/ICMP Checksum Engine processes the IPv4 or IPv6 header (including extension headers) and determines whether the encapsulated payload is TCP, UDP, or ICMP.

The checksum is calculated for the TCP, UDP, or ICMP payload and inserted into its corresponding field in the header. This engine can work in the following two modes:

- The TCP, UDP, or ICMPv6 pseudo-header is not included in the checksum calculation and is assumed to be present in the Checksum field of the input frame. This engine includes the Checksum field in the checksum calculation, and then replaces the Checksum field with the final calculated checksum.
- The engine ignores the Checksum field, includes the TCP, UDP, or ICMPv6 pseudo-header data into the checksum calculation, and overwrites the checksum field with the final calculated value.

For ICMP-over-IPv4 packets, the Checksum field in the ICMP packet must always be 16'h0000 in both modes, because pseudo-headers are not defined for such packets. If it does not equal 16'h0000, an incorrect checksum may be inserted into the packet.

The result of this operation is indicated by the Payload Checksum Error status bit in the Transmit Status vector (Bit 12 in Table 6.34-15). This engine sets the Payload Checksum Error status bit when it detects

that the frame has been forwarded to the MAC Transmitter engine in the store-and-forward mode without the end of frame (EOF) being written to the FIFO, or when the packet ends before the number of bytes indicated by the Payload Length field in the IP Header is received. When the packet is longer than the indicated payload length, the COE ignores them as stuff bytes, and no error is reported. When this engine detects the first type of error, it does not modify the TCP, UDP, or ICMP header. For the second error type, it still inserts the calculated checksum into the corresponding header field.

Table 6.34-10 describes the functions supported by Transmit Checksum Offload engine based on the packet type. When the MAC does not insert the checksum, it is indicated as “No” in the table.

You should not enable checksum insertion for IPv4 or IPv6 packets that are greater than the frame size constraint specified in Transmit Checksum Offload Engine because it may result in incorrect checksum insertion or unexpected behavior.

Packet type	Hardware IP header checksum insertion	Hardware TCP/UDP checksum insertion
Non-IPv4 or IPv6	No	No
IPv4 packet is greater than 1,522 bytes (2,000 bytes when IEEE 802.3ad Support for 2K Packets is enabled in MAC) but less than or equal to the frame size constraint specified in “Transmit Checksum Offload Engine”.	Yes	Yes
IPv6 packet is greater than 1,522 bytes (2,000 bytes when IEEE 802.3ad Support for 2K Packets is enabled in MAC) but less than or equal to the frame size constraint specified in “Transmit Checksum Offload Engine”.	Not applicable	Yes
IPv4 with TCP, UDP, or ICMP	Yes	Yes
IPv4 packet has IP options (IP header is longer than 20 bytes)	Yes	Yes
Packet is an IPv4 fragment	Yes	No
IPv6 packet with the following next header fields in main or extension headers: ■Hop-by-hop options (in IPv6 main header) ■Hop-by-hop options (in IPv6 extension header) ■Destinations options ■Routing (with segment left 0) ■Routing (with segment left > 0) ■TCP, UDP, or ICMP ■Any other next header field in main or extension headers	 ■Not applicable ■Not applicable ■Not applicable ■Not applicable ■Not applicable ■Not applicable ■Not applicable	 ■Yes ■No ■Yes ■No ■No ■Yes ■No
IPv4 packet has TCP header with Options fields	Yes	Yes
IPv4 Tunnels: ■IPv4 packet in an IPv4 tunnel ■IPv6 packet in an IPv4 tunnel	 ■Yes (IPv4 tunnel header) ■Yes (IPv4 tunnel header)	 ■No ■No
IPv6 Tunnels: ■IPv4 packet in an IPv6 tunnel ■IPv6 packet in an IPv6 tunnel	 ■Not applicable ■Not applicable	 ■No ■No
IPv4 packet has 802.3ac tag (with C-VLAN Tag or S-VLAN Tag when enabled).	Yes	Yes
IPv6 packet has 802.3ac tag (with C-VLAN Tag or S-VLAN Tag when enabled).	Not applicable	Yes

IPv4 frames with security features (such as encapsulated security payload)	Yes	No
IPv6 frames with security features (such as encapsulated security payload)	Not applicable	No

Table 6.34-10 Transmit Checksum Offload Engine Functions for Different Packet Types

Receive Checksum Offload Engine

There are two types of Receive Checksum Offload engine available for configuration.

Type 1

The application can enable IP header checksum checking and TCP/UDP checksum offload by setting Bit 10 (IPC) of Register 0 (MAC Configuration Register). This module calculates the checksum by taking 16-bit one's complement of sum of the Ethernet frame's payload data's (DATA field) one's complement. The payload data is assumed to start from byte 15 (19 for a VLAN-tagged frame) of the received Ethernet frame. This module only processes IPv4 datagrams, bypassing and not processing all other types (such as IPv6). The receiver identifies an IPv4 frame by checking if the first nibble of the Ethernet payload (IP header) contains value 0x4.

This module also compares the calculated IP checksum with the received frame's IPv4 header checksum. Bytes 25 and 26 of the received Ethernet frame (29 and 30 for a VLAN-tagged frame) are taken as the IP header checksum. The header checksum is calculated against the header length field (20 bytes minimum). The result of the comparison (pass or fail) is given to the RFC, which sets the appropriate bit in the receive status word. If the Header Length field value is less than 5 or if the IP Version field does not equal 4, an error is indicated for the IP header checksum.

The one's complement sum of the IP datagram's 16-bit payload is also calculated. The start of the payload is considered to be the data after the IP header. If the data payload ends with a non-aligned half-word, then a pad byte is added for the sum calculation. The 16-bit one's complement of the resultant sum is forwarded to the RFC module, which inserts it into the data stream (towards the application) right after the FCS bytes (MS byte first) of the Ethernet payload. This 16-bit sum helps the software check the TCP/UDP header checksums faster. This 16-bit sum (which is always appended to the Ethernet frame in this mode) is invalid when the IP header checksum bit shows an Error status.

Type 2

In this mode, both IPv4 and IPv6 frames in the received Ethernet frames are detected and processed for data integrity. Like Type 1, you can enable this module by setting Bit 10 (IPC) of Register 0 (MAC Configuration Register). The MAC receiver identifies IPv4 or IPv6 frames by checking for value 0x0800 or 0x86DD, respectively, in the received Ethernet frames' Type field. This identification also applies to single VLAN- tagged frames.

The Receive Checksum Offload engine calculates IPv4 header checksums and checks that they match the received IPv4 header checksums. The result of this operation (pass or fail) is given to the RFC module for insertion into the receive status word. The IP Header Error bit is set for any mismatch between the indicated payload type (Ethernet Type field) and the IP header version, or when the received frame does not have enough bytes, as indicated by the Length field of the IPv4 header (or when fewer than 20 bytes are available in an IPv4 or IPv6 header).

This engine also identifies a TCP, UDP, or ICMP payload in the received IP datagrams (IPv4 or IPv6) and calculates the checksum of such payloads properly, as defined in the TCP, UDP, or ICMP specifications. This engine includes the TCP, UDP, or ICMPv6 pseudo-header bytes for checksum calculation and checks whether the received checksum field matches the calculated value. The result of this operation is given as a Payload Checksum Error bit in the receive status word. This status bit is also set if the length of the TCP, UDP, or ICMP payload is less than the expected payload length given in the IP header.

Table 6.34-11 describes the functions supported by Type 2 Receive Checksum Offload engine based on the packet type.

In Type 2 mode, the MAC does not append any payload checksum bytes to the received Ethernet frames (as the Type 1 engine does).

Packet type	Hardware IP header checksum checking	Hardware TCP/UDP/ICMP checksum checking
Non-IPv4 or IPv6	No	No
IPv4 packet is greater than 1522 bytes (2000 bytes when IEEE 802.3ad Support for 2K Packets is enabled in the MAC)	Yes	Yes
IPv6 packet is greater than 1522 bytes (2000 bytes when IEEE 802.3ad Support for 2K Packets is enabled in the MAC)	Not applicable	Yes
IPv4 with TCP, UDP, or ICMP	Yes	Yes
IPv4 header's protocol field contains a protocol other than TCP, UDP, or ICMP	Yes	No
IPv4 packet has IP options (IP header is longer than 20 bytes)	Yes	Yes
Packet is an IPv4 fragment	Yes	No
IPv6 packet with the following next header fields in main or extension headers: ■Hop-by-hop options (in IPv6 main header) ■Hop-by-hop options (in IPv6 extension header) ■Destinations options ■Routing (with segment left 0) ■Routing (with segment left > 0) ■Authentication ■TCP, UDP, or ICMP ■Any other next header field in main or extension headers	 ■Not applicable ■Not applicable ■Not applicable ■Not applicable ■Not applicable ■Not applicable ■Not applicable ■Not applicable	 ■Yes ■No ■Yes ■Yes ■No ■Yes ■Yes ■No
IPv4 packet has TCP header with Options fields	Yes	Yes
IPv4 Tunnels: ■IPv4 packet in an IPv4 tunnel ■IPv6 packet in an IPv4 tunnel	 ■Yes (IPv4 tunnel header) ■Yes (IPv4 tunnel header)	 ■No ■No
IPv6 Tunnels: ■IPv4 packet in an IPv6 tunnel ■IPv6 packet in an IPv6 tunnel	 ■Not applicable ■Not applicable	 ■No ■No
IPv4 packet has 802.3ac tag (with C-VLAN Tag or S-VLAN Tag when enabled).	Yes	Yes
IPv6 packet has 802.3ac tag (with C-VLAN Tag or S-VLAN Tag when enabled).	Not applicable	Yes
IPv4 frames with security features (such as encapsulated security payload)	Yes	No
IPv6 frames with security features (such as encapsulated security payload)	Not applicable	No

Table 6.34-11 Receive Checksum Offload Engine (Type 2) Functions for Different Packet Types

6.34.5.6 Power Management Block

The power management (PMT) block supports the reception of magic packet frames. The PMT block does not perform the clock gate function, but generates interrupts for magic packets that the MAC receives.

When you enable the power-down mode in the PMT block, the MAC drops all received frames and does not forward any frame to the MTL RxFIFO or the application. The MAC comes out of the power-down mode only when a magic packet frame is received and the corresponding detection is enabled.

The PMT block is available in the receive path of MAC. You can use Bit 1 of Register 11 (PMT Control and Status Register) to generate power management events. The application should program these bits. You can access the PMT registers in the similar manner as you access the MAC CSR registers. For mapping information, see Register 11 (PMT Control and Status Register).

Magic Packet Detection

The magic packet frame is based on a method that uses Advanced Micro Device's magic packet technology to power up the sleeping device on the network. The MAC receives a specific packet of information, called a magic packet, addressed to the node on the network.

The MAC checks only those magic packets that are addressed to the MAC or a broadcast address to determine whether these packets meet the wake-up requirements. The magic packets that pass the address filtering (unicast or broadcast) are checked to determine whether they meet the remote wake-up frame data format of 6 bytes of all ones followed by a MAC Address appearing 16 times.

The application enables the magic packet wake-up by writing 1 to Bit 1 (MGKPKTEN) of the PMT Control and Status Register. The PMT block constantly monitors each frame addressed to the node for a specific magic packet pattern. Each frame received is checked for a 48'hFF_FF_FF_FF_FF_FF pattern following the destination and source address field. The PMT block then checks the frame for 16 repetitions of the MAC address without any breaks or interruptions. In case of a break in the 16 repetitions of the address, the PMT block again scans the 48'hFF_FF_FF_FF_FF_FF pattern in the incoming frame. The 16 repetitions can be anywhere in the frame, but must be preceded by the synchronization stream (48'hFF_FF_FF_FF_FF_FF). The device can also accept a multicast frame, as long as the 16 duplications of the MAC address are detected.

If the MAC address of a node is 48'h00_11_22_33_44_55, then the MAC scans for the following data sequence:

Destination Address Source Address …….. FF FF FF FF FF FF

00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55

00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55

00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55

00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55

…CRC

The magic packet detection is updated in the PMT Control and Status Register for the received magic packet. A PMT interrupt to the Application triggers a read to the PMT Control and Status Register to determine whether a magic packet frame has been received.

PMT Interrupts

The PMT interrupt signal is asserted when a valid magic packet frame is received. When software resets the Bit 0 (PWRDWN) in PMT Control and Status Register, the MAC comes out of the power-down mode, but this event does not generate the PMT interrupt.

6.34.5.7 Station Management Agent

The Station Management Agent (SMA) module allows the Application to access any PHY registers through a 2-wire Station Management interface (MIM).

The application can select one of the PHYs and one of the 32 registers within any PHY and send control data or receive status information. Only one register in one PHY can be addressed at any given time. For more details about the communication from the Application to the PHYs, refer to the Reconciliation Sublayer and Media Independent Interface Specifications section of the IEEE 802.3z specification, 1000BASE Ethernet. The application sends the control data to the PHY and receives status information from the PHY through the SMA module.

Functions

The MAC initiates the Management Write or Read operation. The MDC clock is a divided clock from the CSR clock. The divide factor depends on the clock range setting in the GMII Address register. Clock range is set as follows:

Selection	CSR Clock	MDC Clock
0000	60–100 MHz	CSR clock/42
0001	100–150 MHz	CSR clock/62
0010	20–35 MHz	CSR clock/16
0011	35–60 MHz	CSR clock/26
0100	150–250 MHz	CSR clock/102
0101	250–300 MHz	CSR clock/124
0110, 0111	Reserved	

The following figure displays the structure of a frame on the MDIO line:

IDLE	PREAMBLE	START	OPCODE	PHY ADDR	REG ADDR	TA	DATA	IDLE

Table 6.34-12 provides information about the frame fields.

Field	Description
IDLE	The mdio line is three-state; there is no clock on gmii_mdc_o.
PREAMBLE	32 continuous bits of value 1
START	Start of frame is 2’01
OPCODE	2’b10 for Read and 2’b01 for Write
PHY ADDR	5-bit address select for one of 32 PHYs
REG ADDR	Register address in the selected PHY
TA	Turnaround is 2’bZ0 for Read and 2’b10 for Write
DATA	Any 16-bit value. In a Write operation, the MAC drives mdio. In a Read operation, PHY drives it.

Table 6.34-12 MDIO Frame Structure

GMII/MII Management Write Operation

When you set Bit 1 (GMII Write) and Bit 0 (GMII Busy) in Register 4 (GMII Address Register), the MAC CSR module transfers the PHY address, the register address in PHY, and the write data (GMII Data Register) to the SMA to initiate a Write operation into the PHY registers. At this point, the SMA module starts a Write operation on the GMII Management Interface using the Management Frame Format specified in the GMII specifications (as per IEEE Standard 802.3-2002, Section 22.2.4.5). The application should not change the content of the GMII Address register or GMII Data register while the transaction is going on. The MAC ignores the Write operations performed to the GMII Address register or the GMII Data Register during this period (the Busy bit is high), and the transaction is completed without any error on the MCI interface.

When the Write operation is complete, the SMA indicates this to the CSR. The CSR then resets the Busy bit. The SMA module divides the CSR (Application) clock with the clock divider programmed (CR bits of GMII Address Register) to generate the MDC clock for this interface. The MAC drives the MDIO line for the complete duration of the frame. The frame format for the Write operation is as follows:

IDLE	PREAMBLE	START	OPCODE	PHY ADDR	REG ADDR	TA	DATA	IDLE
Z	1111...11	01	01	AAAAA	RRRRR	10	DDD . . . .DDD	Z

GMII/MII Management Read Operation

When you set Bit 1 and Bit 0 to 0 in Register 4 (GMII Address Register), the MAC CSR module transfers the PHY address and the register address in PHY to the SMA to initiate a Read operation in the PHY registers. At this point, the SMA module starts a Read operation on the GMII Management Interface using the Management Frame Format specified in the GMII specifications (as per IEEE Standard 802.3-2002, Section 22.2.4.5). The application should not change the contents of the GMII Address register or GMII Data register while the transaction is going on. The MAC ignores the Write operations to the GMII Address register or GMII Data Register during this period (the Busy bit is high) and the transaction is completed without any error on the MCI interface.

When the Read operation is complete, the SMA indicates this to the CSR. The CSR resets the Busy bit and updates the GMII Data register with the data read from the PHY. The SMA module divides the CSR (Application) clock with the clock divider programmed (CR bits of GMII Address Register) to generate the MDC clock for this interface. The MAC drives the MDIO line for the complete duration of the frame except during the Data fields when the PHY is driving the MDIO line.

The frame format for the Read operation is as follows:

IDLE	PREAMBLE	START	OPCODE	PHY ADDR	REG ADDR	TA	DATA	IDLE
Z	1111...11	01	10	AAAAA	RRRRR	Z0	DDD . . . .DDD	Z

The MAC drives the MDO on negedge of MDC so that it can be sampled on posedge with sufficient setup or hold time (half cycle of MDC). Similarly, the PHY can drive MDI on negedge so that the MAC samples it on posedge, thus providing half cycle setup or hold time.

6.34.5.8 Reduced Media Independent Interface

The Reduced Media Independent Interface (RMII) specification reduces the pin count between Ethernet PHYs and Switch ASICs (only in 10/100 mode). According to the IEEE 802.3u standard, an MII contains 16 pins for data and control. In devices incorporating multiple MAC or PHY interfaces (such as switches), the number of pins adds significant cost with increase in port count. The RMII specification addresses this problem by reducing the pin count to 7 for each port — a 62.5% decrease in pin count.

The RMII module is instantiated between the MAC and the PHY. This helps in translating MAC's MII

into the RMII. The RMII block has the following characteristics:

- Supports 10-Mbps and 100-Mbps operating rates. It does not support the 1000-Mbps operation.
- Provides independent, 2-bit wide transmit and receive paths by sourcing two clock references externally.

Block Diagram

Figure 6.34-11 shows the position of the RMII block relative to the GMAC and RMII PHY. The RMII block is placed in front of the GMAC to translate the MII signals to RMII signals.

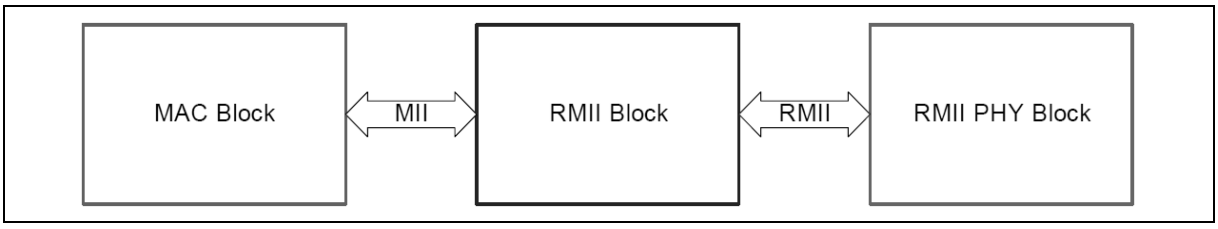


Figure 6.34-11 RMII Block Diagram

Transmit Bit Ordering

Each nibble from the MII must be transmitted on the RMII a di-bit at a time with the order of di-bit transmission shown in Figure 6.34-12. The lower order bits (D0 and D1) are transmitted first followed by higher order bits (D2 and D3).

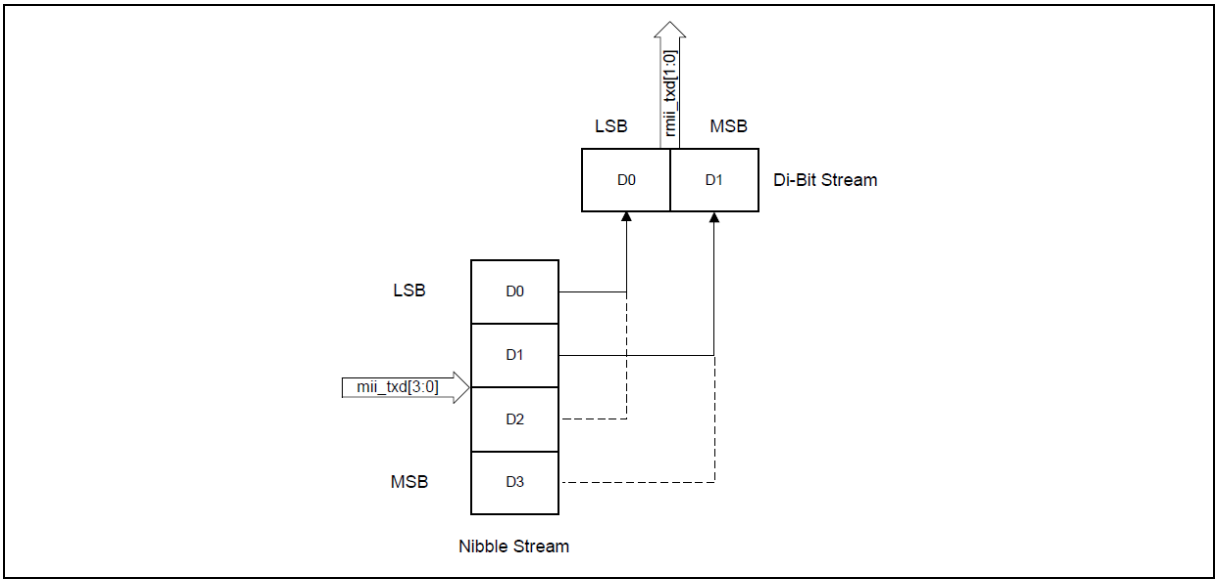


Figure 6.34-12 Transmission Bit Ordering

Receive Bit Ordering

Each nibble is transmitted to the MII from the di-bit received from the RMII in the nibble transmission order shown in Figure 6.34-13. The lower order bits (D0 and D1) are received first, followed by the higher order bits (D2 and D3).

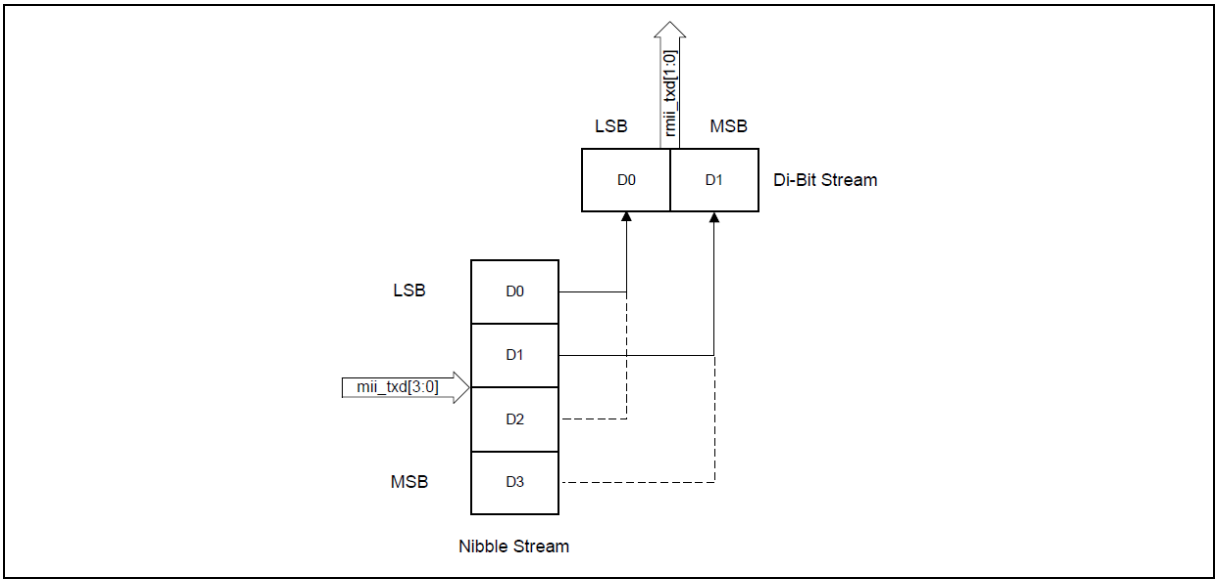


Figure 6.34-13 Receive Bit Ordering

6.34.5.9 Reduced Gigabit Media Independent Interface

The Reduced Gigabit Media Independent Interface (RGMII) specification reduces the pin count of the interconnection between the MAC and the PHY for GMII and MII interfaces. To achieve this, the data path and control signals are reduced and multiplexed together with both the edges of the transmit and receive clocks. For gigabit operation the clocks operate at 125 MHz; for 10/100 operation, the clock rates are 2.5 MHz/25 MHz.

The optional RGMII module is instantiated between the MAC's GMII and the PHY to translate the control and data signals between the GMII and RGMII protocols. The RGMII block supports 10-Mbps, 100-Mbps, and 1000-Mbps operation rates.

Block Diagram

Figure 6.34-14 shows the position of the RGMII block relative to the MAC and RGMII PHY. The RGMII block is placed between the GMII and the PHY to translate the GMII signals to RGMII signals.

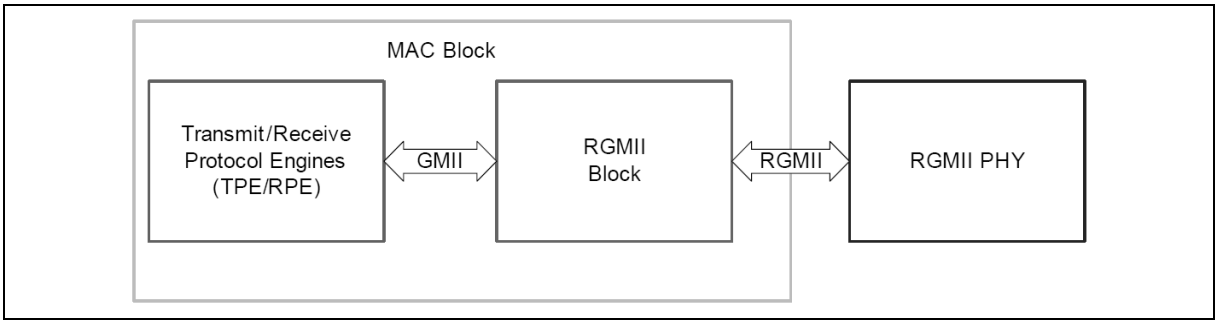


Figure 6.34-14 RGMII Block Diagram

RGMII Clocks

The RGMII operation uses both edges of the RGMII transmit and receive clocks (clk_tx_i and clk_rx_i), which run at 125 MHz for gigabit operation or 2.5 MHz/25 MHz for 10/100 operation. These clocks come from a separate PLL logic block. To simplify implementation of the MAC, the RGMII block provides two additional inputs to drive the logic that uses the falling edges of clk_tx_i and clk_rx_i:

- clk_tx_180_i must be 180 degrees out-of-phase with respect to clk_tx_i

- clk_rx_180_i must be 180 degrees out-of-phase with respect to clk_rx_i

Separate synchronous inputs are present to synchronously reset the logic driven by each of the four RGMII clocks.

Signal Conversion

Transmit Data Conversion

As defined in the RGMII specification, multiplexing of data and control in gigabit mode is accomplished by using both edges of the transmit clock. The RGMII block accepts the 8-bit GMII transmit data from the MAC GMII (gmii_txd[7: 0]) on the rising edge of clk_tx_i. For gigabit operation, the RGMII block then converts gmii_txd[7: 0] to two 4-bit nibbles and transmits the data on the RGMII transmit data port (rgmii_txd_o[3: 0]) as follows:

- At the rising edge of clk_tx_i, rgmii_txd_o[3: 0] contains gmii_txd[3: 0]
- At the falling edge of clk_tx_i (rising edge of clk_tx_180_i), rgmii_txd_o[3: 0] contains gmii_txd[7: 4]

For 10/100 mode, the transmit data to the PHY is 4 bits. So, for 10/100 mode, the RGMII block drives the gmii_txd[3: 0] data to the PHY on rgmii_txd_o[3: 0] at the rising edge of clk_tx_i.

Transmit Control Signal Conversion

The RGMII block accepts the GMII transmit control signals (gmii_txen and gmii_txer) from the MAC GMII on the rising edge of clk_tx_i, then multiplexes gmii_txen and gmii_txer onto the rgmii_tctl_o output as required by the RGMII specification:

- At the rising edge of clk_tx_i, rgmii_tctl_o is gmii_txen.
- At the falling edge of clk_tx_i (rising edge of clk_tx_180_i), rgmii_tctl_o is the logical XOR of gmii_txen and gmii_txer.

Table 6.34-13 shows the mapping of the PLS_DATA.request parameters from the MAC to the corresponding signaling on the GMII and RGMII control and data signals. This table shows the encoding of only gmii_txen and gmii_txer signals.

gmii_txen	gmii_txer	gmii_txd[a]	rgmii_tctl_o[b]	Description	PLS_DATA.request parameter
0	0	0x00–0xFF	0,0	Normal inter-frame	TRANSMIT_COMPLETE
0	1	0x00	0,1	Reserved	ZERO, ONE (8 bits)
0	1	0x01	0,1	LPI pattern	TRANSMIT_LPI in EEE mode
0	1	0x02–0x0E	0,1	Reserved	ZERO, ONE (8 bits)
0	1	0x0F	0,1	Carrier extend	EXTEND (8 bits)
0	1	0x10–0x1E	0,1	Reserved	
0	1	0x1F	0,1	Carrier extend error	EXTEND_ERROR (8 bits)
0	1	0x20–0xFF	0,1	Reserved	
1	0	0x00–0xFF	1,1	Normal data transmission	ZERO, ONE (8 bits)
1	1	0x00–0xFF	1,0	Transmit error propagation	No applicable parameter

a. When the RGMII interface is configured to transmit the configuration during the IFG, then rgmii_txd[3: 0] reflects the Duplex Mode, Port Select, Speed (encoded as 00 for 10 Mbps, 01 for 100 Mbps and 10 for 1000 Mbps), and Link Up/Down bits of the MAC Configuration Register, during this period.

b. Values at rising, falling edges of clk_tx_i.

Table 6.34-13 Encoding of gmii_txen, gmii_txer, rgmii_tctl_o, and rgmii_txd_o Signals

Receive Data Conversion

For Gigabit mode, the RGMII block registers the 4-bit data on rgmii_rxd_i[3: 0] at both edges of the receive clock (clk_rx_i) and transfers the resulting 8-bit data to the GMAC GMII on the rising edge of clk_rx_i, as follows:

- gmii_rxd[3: 0] is the value received on rgmii_rxd_i[3: 0] at the rising edge of clk_rx_i
- gmii_rxd[7: 4] is the value received on rgmii_rxd_i[3: 0] at the falling edge of clk_rx_i (rising edge of clk_rx_180_i)

For 10/100 mode, the RGMII block registers the 4-bit data on rgmii_rxd_i[3: 0] only at the rising edge of clk_rx_i and transfers the data to the GMAC GMII on the rising edge of clk_rx_i, as follows:

- gmii_rxd[3: 0] is the value received on rgmii_rxd_i[3:0] at the rising edge of clk_rx_i.
- gmii_rxd[7:4] is 0x0.

Receive Control Signal Conversion

The RGMII block samples the rgmii_rctl_i signal from the PHY at both edges of clk_rx_i and drives the resulting GMII receive control signals to the GMAC GMII as follows:

- gmii_rxdv is the value received on rgmii_rctl_i at the rising edge of clk_rx_i.
- gmii_rxer is the logical XOR of the following two signals:
  - The value received on rgmii_rctl_i at the rising edge of clk_rx_i (gmii_rxdv).
  - The value received on rgmii_rctl_i at the falling edge of clk_rx_i (rising edge of clk_rx_180_i).

Receive Status Signal Conversion

To generate the GMII link status signals, the RGMII block decodes the rgmii_rxd_i[3:0] value sampled at the rising edge of clk_rx_i when the value of rgmii_rctl_i is 0 for both the rising and falling edges of clk_rx_i (normal inter-frame value). The decoded status signals are:

- link_status is the value of rgmii_rxd_i[0].
- link_speed is the value of rgmii_rxd_i[2:1].
- link_mode is the value of rgmii_rxd_i[3].

Table 6.34-14 shows the mapping of the PLS_DATA.indicate parameters from the GMAC to the corresponding signaling on the GMII and RGMII control and data signals. In addition, the RGMII block asserts gmii_crs to the GMAC GMII when any of the following conditions is true:

- The gmii_rxdv is true.
- The gmii_rxdv is false, gmii_rxer is true, and the value of gmii_rxd is 0xFF.
- A carrier extend, carrier extend error, or false carrier occurs in gigabit mode (see Table 6.34-14).
- A false carrier occurs in 10/100 mode.
- The RGMII is transmitting data, carrier extend, or carrier extend error on the RGMII transmit outputs.

The RGMII block asserts gmii_col to the MAC GMII when both of the following conditions is true:

- The RGMII is transmitting data, carrier extend, or carrier extend error on the RGMII transmit outputs.
- The RGMII is asserting either gmii_crs or gmii_rxdv.

rgmii_rctl_i[a]	gmii_rxd[7:0] ([3:0] in 10/100 mode)	gmii_rxdv	gmii_rxer	Description	PLS_DATA.indicate or PHY_status parameter

0,0	XXXXXXX0 or XXXXXXX1	0	0	Normal inter-frame	Indicates link status: ■0 = down ■1 = up
0,0	XXXXX00X or XXXXX01X or XXXXX10X or XXXXX11X	0	0	Normal inter-frame	Indicates receive clock frequency: ■00 = 2.5 MHz ■01 = 25 MHz ■10 = 125 MHz ■11 = Reserved
0,0	XXXX1XXX or XXXX0XXX	0	0	Normal inter-frame	Indicates duplex status: ■0 = half-duplex ■1 = full-duplex
0,1	0x01 (or 0x1 in 10/100 mode)	0	1	Low Power Idle (LPI)	Remote MAC in LPI in the Energy Efficient Ethernet (EEE) mode
0,1	0x0E (or 0xE in 10/100 mode)	0	1	False carrier indication	False carrier present
0,1	0x0F	0	1	Carrier extend	EXTEND (8 bits)
0,1	0x1F	0	1	Carrier extend error	ZERO, ONE (8 bits)
0,1	0xFF (or 0xF in 10/100 mode)	0	1	Carrier sense	PLS_Carrier.Indicate
1,1	0x00–0xFF	1	0	Normal data reception	ZERO, ONE (8 bits)
1,0	0x00–0xFF	1	1	Data reception error	ZERO, ONE (8 bits)

a. Values at rising, falling edges of clk_rx_i..

Table 6.34-14 Decoding of rgmii_rctl_i and rgmii_rxd_i

RGMII Transmit Timing

Figure 6.34-15 shows the timing for RGMII transmission. The RGMII specification defines a transmit skew (TskewT) of ± 0.5 ns. As shown in Figure 6.34-15, the trace delay on the PC board for clk_tx_i going to a receiving PHY must satisfy the set-up and hold time requirements of the RGMII transmit signals at the receiving PHY.

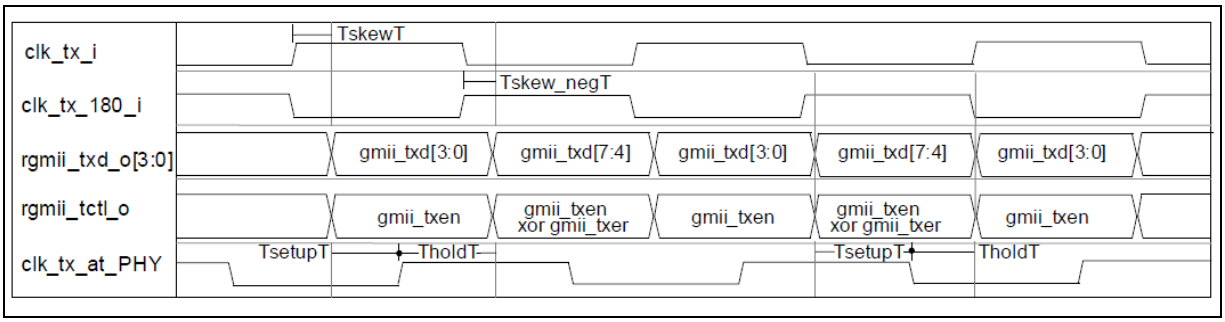


Figure 6.34-15 RGMII Transmit Timing

RGMII Receive Timing

Figure 6.34-16 shows the receive timing for RGMII. The TsetupR and TholdR values are required to be a minimum of 1 ns.

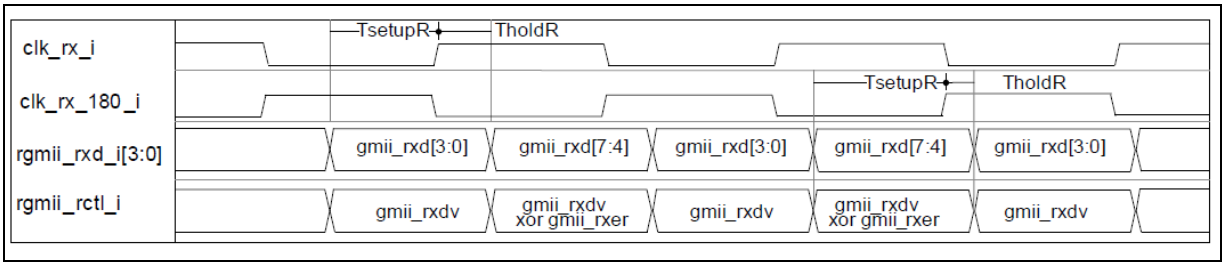


Figure 6.34-16 RGMII Receive Timing

6.34.5.10 Interrupts from the MAC

Interrupts can be generated from the MAC as a result of various events in the optional modules. These interrupt events are combined with the events in the DMA on the sbd_intr_o signal. The MAC interrupts are of level type, that is, the interrupt remains asserted (high) until it is cleared by the host or software.

The Interrupt Status register in chapter **1.1.7 Register Map** describes the events that can cause an interrupt from the MAC. The MAC interrupts are enabled by default. Each event can be prevented from asserting the interrupt on the mci_intr_o or sbd_intr_o signals by setting the corresponding mask bits in the Register 15 (Interrupt Mask Register).

The interrupt register bits only indicate the block from which the event is reported. You must read the corresponding status registers and other registers to clear the interrupt. For example, when set high, Bit 0 of the Register 14 (Interrupt Status Register) indicates that the link status on the RGMII interface has changed. You must read Register 54 (RGMII Control and Status Register) to clear this interrupt event.

The interrupts from the RGMII is given as the GLI bit in the Register 1005 (Status Register). The TTI, GPI, GLPII, and GMI signals in Figure 6.34-17 refer to the bits that can be read in the Register 1005 (Status Register).

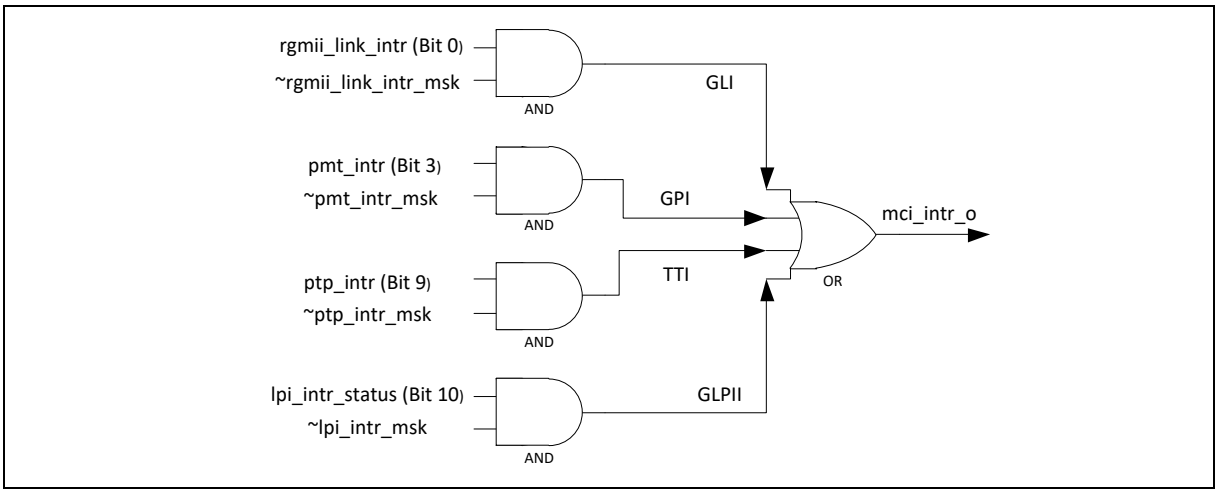


Figure 6.34-17 MAC Interrupt Masking Scheme

6.34.6 Descriptors

This chapter describes the GMAC descriptors. It contains the following sections:

- "Overview"
- "Descriptor Endianness"
- "Normal Descriptor"
- "Alternate or Enhanced Descriptors"

6.34.6.1 Overview

The DMA in the Ethernet subsystem transfers data based on a linked list of descriptors. The descriptors are created in the system memory. The GMAC supports the following two types of descriptors:

- Normal Descriptor

  This is the default descriptor format which can have 4 DWORDS. For information about this descriptor type, see 1.1.6.3 “Normal Descriptor”.

- Enhanced Descriptor

  This is the alternate descriptor format which can have 8 DWORDS (32-bytes).

Each descriptor contains two buffers, two byte-count buffers, and two address pointers, which enable the adapter port to be compatible with various types of memory management schemes. Once configured, you cannot change the descriptor type.

There is no limit for number of descriptors that can be used for a single frame.

6.34.6.2 Descriptor Endianness

The descriptor addresses must be aligned to the used bus width (Word, DWord, or LWord for 32-bit, 64-bit, or 128-bit bus respectively).

Figure 6.34-18 shows the normal Receive and Transmit descriptors for 64-bit data bus when the endian mode of data bus and descriptors is little-endian.

	63 55	47	39	31	23 15 7 0
DES1-DES0	Control Bits [9:0]	Byte Count Buffer2 [10:0]	Byte Count Buffer1[10:0]	OWN	Status [30:0]
DES3-DES2	Buffer2 Address [31:0] / Next Descriptor Address [31:0]			Buffer1 Address[31:0]	

Figure 6.34-18 Rx/Tx Descriptors in Same-Endian Mode for 64-Bit, Little-Endian Format Data Bus

6.34.6.3 Normal Descriptor

This is the default descriptor format which can have 4 DWORDS. You can use this descriptor format when advanced features such as IEEE 1588-2008 Advanced Timestamping, and SA, VLAN, and CRC Insertion are not enabled.

Transmit Descriptor

The DMA in GMAC requires at least one descriptor for a transmit frame. In addition to two buffers, two byte-count buffers, and two address pointers, the transmit descriptor has control fields which can be used to control the MAC operation on per-transmit frame basis.

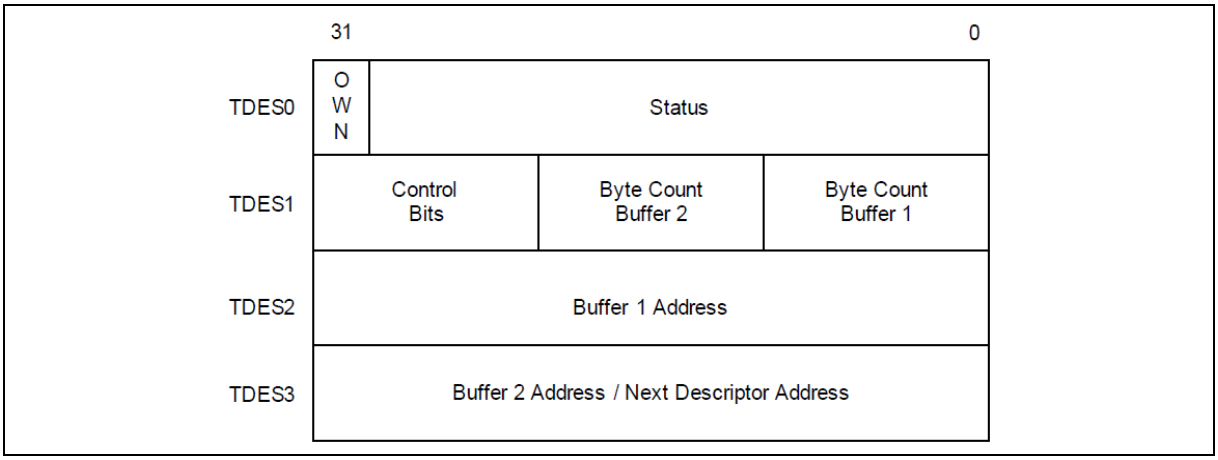


Figure 6.34-19 Transmit Descriptor Format in Little-Endian Mode With a 32-Bit Data Bus

Transmit Descriptor 0 (TDES0)

TDES0 contains the transmitted frame status and the descriptor ownership information. The descriptions in the Tables Table 6.34-15 through Table 6.34-18 are for the default mode of a little-endian 32-bit data bus with same-endian descriptors, or big-endian data bus with reverse-endian descriptors. If the DUT is configured for big-endian mode, the 32-bit data bus with same-endian descriptors or little-endian data bus with reverse-endian descriptors, then byte lanes 3 and 0 are swapped whereas byte lanes 1 and 2 are swapped, that is, Bits [31:24] are available on data bus [7:0], and vice-versa.

The status bits are valid when the LS bit is set. Therefore, when multiple descriptors are used for transmitting a single frame, the status bits of only last descriptor (that is, descriptor with LS bit set) are valid.

Bit	Description
31	OWN: Own Bit When set, this bit indicates that the descriptor is owned by the DMA. When this bit is reset, it indicates that the descriptor is owned by the Host. The DMA clears this bit either when it completes the frame transmission or when the buffers allocated in the descriptor are empty. The ownership bit of the First Descriptor of the frame should be set after all subsequent descriptors belonging to the same frame have been set. This avoids a possible race condition between fetching a descriptor and the driver setting an ownership bit.
30:18	Reserved
17	TTSS: Tx Timestamp Status This status bit indicates that a timestamp has been captured for the corresponding transmit frame. When this bit is set, TDES2 and TDES3 have timestamp values that were captured for the transmit frame. This field is valid only when the Last Segment control bit (TDES1[30]) in a descriptor is set. This bit is valid only when IEEE1588 timestamping feature is enabled; otherwise, it is reserved.
16	IHE: IP Header Error When set, this bit indicates that the Checksum Offload engine detected an IP header error. This bit is valid only when Tx Checksum Offload is enabled. Otherwise, it is reserved. If COE detects an IP header error, it still inserts an IPv4 header checksum if the Ethernet Type field indicates an IPv4 payload.
15	ES: Error Summary Indicates the logical OR of the following bits: TDES0[16] = IP Header Error TDES0[14] = Jabber Timeout TDES0[13] = Frame Flush TDES0[12] = Payload Checksum Error TDES0[11] = Loss of Carrier

	TDES0[10] = No Carrier TDES0[9] = Late Collision TDES0[8] = Excessive Collision TDES0[2] = Excessive Deferral TDES0[1] = Underflow Error
14	JT: Jabber Timeout When set, this bit indicates that the MAC transmitter has experienced a jabber timeout. This bit is only set when the JD bit of Register 0 (MAC Configuration Register) is not set.
13	FF: Frame Flushed When set, this bit indicates that the DMA or MTL flushed the frame because of a software flush command given by the CPU.
12	PCE: Payload Checksum Error This bit, when set, indicates that the Checksum Offload engine had a failure and did not insert any checksum into the encapsulated TCP, UDP, or ICMP payload. This failure can be either because of insufficient bytes, as indicated by the IP Header's Payload Length field, or the MTL starting to forward the frame to the MAC transmitter in the store-and-forward mode without the checksum having been calculated yet. This second error condition only occurs when the Transmit FIFO depth is less than the length of the Ethernet frame being transmitted to avoid deadlock, the MTL starts forwarding the frame when the FIFO is full, even in the store- and-forward mode.
11	LOC: Loss of Carrier When set, this bit indicates that Loss of Carrier occurred during frame transmission (that is, the phy_crs_i signal was inactive for one or more transmit clock periods during frame transmission). This is valid only for the frames transmitted without collision and when the MAC operates in the half-duplex mode.
10	NC: No Carrier When set, this bit indicates that the carrier sense signal from the PHY was not asserted during transmission. **Note:** When any optional PHY interface is enabled, the carrier errors (NC or LOC) cannot be generated. This is because the CRS signal is internally-generated based on the internal MII TX_EN and RX_DV derived from the optional PHY interface signals.
9	LC: Late Collision When set, this bit indicates that frame transmission was aborted due to a collision occurring after the collision window (64 byte times including Preamble in MII Mode and 512 byte times including Preamble and Carrier Extension in GMII Mode). Not valid if Underflow Error is set.
8	EC: Excessive Collision When set, this bit indicates that the transmission was aborted after 16 successive collisions while attempting to transmit the current frame. If Bit 9 (Disable Retry) in Register 0 (MAC Configuration Register) is set, this bit is set after the first collision and the transmission of the frame is aborted.
7	VF: VLAN Frame When set, this bit indicates that the transmitted frame was a VLAN-type frame.
6:3	CC: Collision Count This 4-bit counter value indicates the number of collisions occurring before the frame was transmitted. The count is not valid when the Excessive Collisions bit (TDES0[8]) is set.
2	ED: Excessive Deferral When set, this bit indicates that the transmission has ended because of excessive deferral of over 24,288 bit times (155,680 bits times in 1000-Mbps mode, or in Jumbo Frame enabled mode) if Bit 4 (Deferral Check) is set high in Register 0 (MAC Configuration Register).
1	UF: Underflow Error When set, this bit indicates that the MAC aborted the frame because data arrived late from the Host memory. Underflow Error indicates that the DMA encountered an empty Transmit Buffer while transmitting the frame. The transmission process enters the suspended state and sets both Transmit Underflow (Register 5[5]) and Transmit Interrupt (Register 5[0]).
0	DB: Deferred Bit When set, this bit indicates that the MAC defers before transmission because of the presence of carrier. This bit is

	valid only in the half-duplex mode..

Table 6.34-15 Transmit Descriptor 0

Transmit Descriptor 1 (TDES1)

TDES1 contains the buffer sizes and other bits which control the descriptor chain or ring and the frame being transferred.

Bit	Description
31	IC: Interrupt on Completion When set, this bit sets Transmit Interrupt (Register 5[0]) after the present frame has been transmitted. **Note:** This bit is valid only when the last segment bit (TDES1[30]) is set.
30	LS: Last Segment When set, this bit indicates that the buffer contains the last segment of the frame. When this bit is set, the TBS1 or TBS2 field should have a non-zero value.
29	FS: First Segment When set, this bit indicates that the buffer contains the first segment of a frame.
28:27	CIC: Checksum Insertion Control These bits control the insertion of checksums in Ethernet frames that encapsulate TCP, UDP, or ICMP over IPv4 or IPv6: ■2'b00 = Do nothing. Checksum Engine is bypassed ■2'b01 = Insert IPv4 header checksum. Use this value to insert IPv4 header checksum when the frame encapsulates an IPv4 datagram. ■2'b10 = Insert TCP/UDP/ICMP checksum. The checksum is calculated over the TCP, UDP, or ICMP segment only and the TCP, UDP, or ICMP pseudo-header checksum is assumed to be present in the corresponding input frame's Checksum field. An IPv4 header checksum is also inserted if the encapsulated datagram conforms to IPv4. ■2'b11 = Insert a TCP/UDP/ICMP checksum that is fully calculated in this engine. In other words, the TCP, UDP, or ICMP pseudo-header is included in the checksum calculation, and the input frame's corresponding Checksum field has an all-zero value. An IPv4 Header checksum is also inserted if the encapsulated datagram conforms to IPv4. The Checksum engine detects whether the TCP, UDP, or ICMP segment is encapsulated in IPv4 or IPv6 and processes its data accordingly. These bits are valid only when the first segment bit (TDES1[29]) is set.
26	DC: Disable CRC When set, the MAC does not append the Cyclic Redundancy Check (CRC) to the end of the transmitted frame. This bit is valid only when the first segment (TDES1[29]) is set.
25	TER: Transmit End of Ring When set, this bit indicates that the descriptor list reached its final descriptor. The returns to the base address of the list, creating a descriptor ring.
24	TCH: Second Address Chained When set, this bit indicates that the second address in the descriptor is the Next Descriptor address rather than the second buffer address. When TDES1[24] is set, TBS2 (TDES1[21–11]) are "don't care" values. TDES1[25] takes precedence over TDES1[24].
23	DP: Disable Padding When set, the MAC does not automatically add padding to a frame shorter than 64 bytes. When this bit is reset, the DMA automatically adds padding and CRC to a frame shorter than 64 bytes and the CRC field is added despite the state of the DC (TDES1[26]) bit. This is valid only when the first segment (TDES1[29]) is set.
22	TTSE: Transmit Timestamp Enable When set, this bit enables IEEE1588 hardware timestamping for the transmit frame referenced by the descriptor. This field is valid only when the First Segment control bit (TDES1[29]) is set.
21:11	TBS2: Transmit Buffer 2 Size

	These bits indicate the Second Data Buffer in bytes. This field is not valid if TDES1[24] is set.
10:0	TBS1: Transmit Buffer 1 Size These bits indicate the First Data Buffer byte size. If this field is 0, the DMA ignores this buffer and uses Buffer 2 or next descriptor depending on the value of TCH (Bit 24).

Table 6.34-16 Transmit Descriptor 1

Transmit Descriptor 2 (TDES2)

TDES2 contains the address pointer to the first buffer of the descriptor.

Bit	Description
31:0	Buffer 1 Address Pointer These bits indicate the physical address of Buffer 1. There is no limitation on the buffer address alignment. See “Host Data Buffer Alignment” for further detail on buffer address alignment.

Table 6.34-17 Transmit Descriptor 2

Transmit Descriptor 3 (TDES3)

TDES3 contains the address pointer either to the second buffer of the descriptor or the next descriptor.

Bit	Description
31:0	Buffer 2 Address Pointer (Next Descriptor Address) Indicates the physical address of Buffer 2 when a descriptor ring structure is used. If the Second Address Chained (TDES1[24]) bit is set, this address contains the pointer to the physical memory where the Next Descriptor is present. The buffer address pointer must be aligned to the bus width only when TDES1[24] is set. (LSBs are ignored internally.)

Table 6.34-18 Transmit Descriptor 3

Receive Descriptor

The DMA in GMAC requires at least two descriptors when receiving a frame. The Receive state machine of the DMA always attempts to acquire an extra descriptor in anticipation of an incoming frame. (The size of the incoming frame is unknown). Before the Rx DMA closes a descriptor, it attempts to acquire the next descriptor even if no frames are received.

In a single descriptor (receive) system, the subsystem generates a descriptor error if receive buffer is unable to accommodate the incoming frame and the next descriptor is not owned by the DMA. Thus, the Host is forced to increase either its descriptor pool or the buffer size. Otherwise, the subsystem starts dropping all incoming frames.

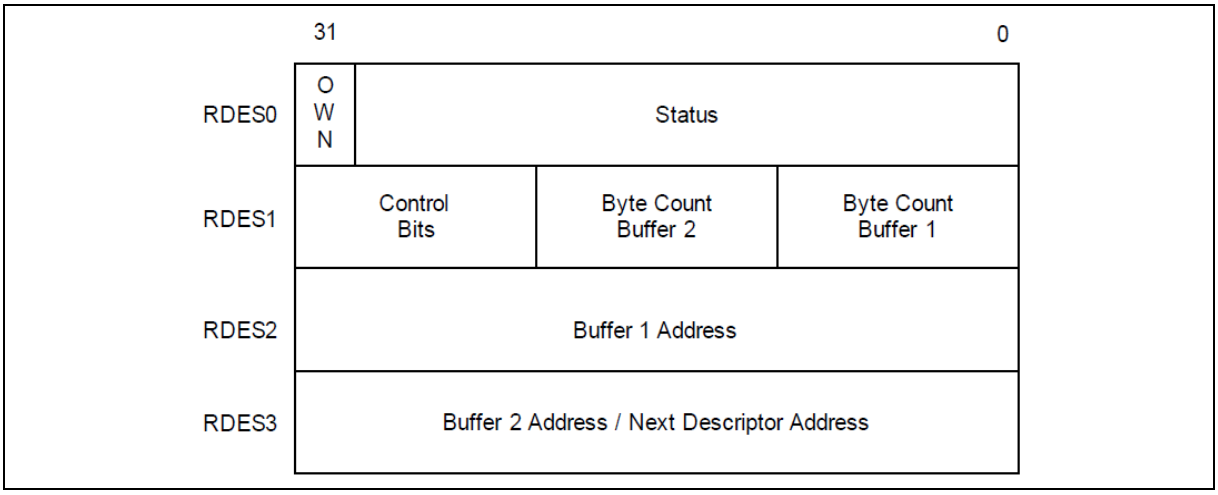


Figure 6.34-20 Receive Descriptor Format in Little-Endian Mode with a 32-Bit Data Bus

Receive Descriptor 0 (RDES0)

RDES0 contains the received frame status, the frame length, and the descriptor ownership information. The descriptions in Table 6.34-19 and Tables Table 6.34-21 through Table 6.34-23 are for the default mode of a little-endian 32-bit data bus with same-endian descriptors.

The status bits are valid when the LS bit is set. Therefore, when multiple descriptors are used for a single frame, status bits of only last descriptor (that is, descriptor with LS bit set) are valid.

Bit	Description
31	OWN: Own Bit When set, this bit indicates that the descriptor is owned by the DMA of the GMAC. When this bit is reset, this bit indicates that the descriptor is owned by the Host. The DMA clears this bit either when it completes the frame reception or when the buffers that are associated with this descriptor are full.
30	AFM: Destination Address Filter Fail When set, this bit indicates a frame that failed in the DA Filter in the MAC.
29:16	FL: Frame Length These bits indicate the byte length of the received frame that was transferred to host memory. This field is valid when Last Descriptor (RDES0[8]) is set and either the Descriptor Error (RDES0[14]) or Overflow Error bit is reset. The frame length also includes the two bytes appended to the Ethernet frame when IP checksum calculation (Type 1) is enabled and the received frame is not a MAC control frame. When the Last Descriptor bit is not set, this field indicates the accumulated number of bytes that have been transferred for the current frame. The inclusion of CRC length in the frame length depends on the settings of Bit 7 and Bit 25 in Register 0 (MAC Configuration Register).
15	ES: Error Summary Indicates the logical OR of the following bits: ■ RDES0[0] = Payload Checksum Error ■ RDES0[1] = CRC Error ■ RDES0[3] = Receive Error ■ RDES0[4] = Watchdog Timeout ■ RDES0[6] = Late Collision ■ RDES0[7] = IPC Checksum (Type 2) or Giant Frame ■ RDES0[11] = Overflow Error

	■ RDES0[14] = Descriptor Error This field is valid only when the Last Descriptor (RDES0[8]) is set.
14	DE: Descriptor Error When set, this bit indicates a frame truncation caused by a frame that does not fit within the current descriptor buffers, and that the DMA does not own the Next Descriptor. The frame is truncated. This field is valid only when the Last Descriptor (RDES0[8]) is set.
13	SAF: Source Address Filter Fail When set, this bit indicates that the SA field of frame failed the SA Filter in the MAC.
12	LE: Length Error When set, this bit indicates that the actual length of the frame received and the Length/ Type field do not match. This bit is valid only when the Frame Type (RDES0[5]) bit is reset. Length error status is not valid when CRC error is present.
11	OE: Overflow Error When set, this bit indicates that the received frame is damaged because of buffer overflow in Rx FIFO. **Note:** This bit is set only when the DMA transfers a partial frame to the application. This happens only when the Rx FIFO is operating in the threshold mode. In the store-and-forward mode, all partial frames are dropped completely in Rx FIFO.
10	VLAN: VLAN Tag When set, this bit indicates that the frame pointed to by this descriptor is a VLAN frame tagged by the MAC. The VLAN tagging depends on checking VLAN fields of the received frame based on the Register 7 (VLAN Tag Register) settings.
9	FS: First Descriptor When set, this bit indicates that this descriptor contains the first buffer of the frame. If the size of the first buffer is 0, the second buffer contains the beginning of the frame. If the size of the second buffer is also 0, the next Descriptor contains the beginning of the frame.
8	LS: Last Descriptor When set, this bit indicates that the buffers pointed to by this descriptor are the last buffers of the frame
7	IPC Checksum Error or Giant Frame When IP Checksum Engine (Type 1) is enabled, this bit, when set, indicates one of the following: ■ The 16-bit IPv4 Header checksum calculated by the MAC did not match the received checksum bytes. ■ The header checksum checking is bypassed for non-IPv4 frames. If this bit is set because of any of these conditions, then Bit 15 (ES) is NOT set. If this bit is set when Full Checksum Offload Engine (Type 2) is enabled, it indicates an error in the IPv4 or IPv6 header. This error can be because of inconsistent Ethernet Type field and IP header Version field values, a header checksum mismatch in IPv4, or an Ethernet frame lacking the expected number of IP header bytes. See Table 6.34-20 for more details.
6	LC: Late Collision When set, this bit indicates that a late collision has occurred while receiving the frame in half-duplex mode.
5	FT: Frame Type When set, this bit indicates that the Receive Frame is an Ethernet-type frame (the LT field is greater than or equal to 1,536). When this bit is reset, it indicates that the received frame is an IEEE802.3 frame. This bit is not valid for Runt frames which are less than 14 bytes. In addition, when Receive COE (Type 2) is selected and Bit 10 (IPC) of Register 0 (MAC Configuration Register) is set, this bit has value as indicated in Table 6.34-20.
4	RWT: Receive Watchdog Timeout When set, this bit indicates that the Receive Watchdog Timer has expired while receiving the current frame and the current frame is truncated after the Watchdog Timeout.
3	RE: Receive Error When set, this bit indicates that the gmii_rxer_i signal is asserted while gmii_rxdv_i is asserted during frame reception. This error also includes carrier extension error in GMII and half-duplex mode. Error can be of less or no extension, or error (rxd ≠ 0f) during extension.
2	DE: Dribble Bit Error When set, this bit indicates that the received frame has a non-integer multiple of bytes (odd nibbles). This bit is valid only

	in MII Mode.
1	CE: CRC Error When set, this bit indicates that a Cyclic Redundancy Check (CRC) Error occurred on the received frame. This field is valid only when the Last Descriptor (RDES0[8]) is set.
0	Rx MAC Address or Payload Checksum Error When set, this bit indicates that the Rx MAC Address registers value (1 to 15) matched the frame's DA field. When reset, this bit indicates that the Rx MAC Address Register 0 value matched the DA field. This bit is invalid when Bit 30 is set. If Full Checksum Offload Engine is enabled, this bit, when set, indicates the TCP, UDP, or ICMP checksum the core calculated does not match the received encapsulated TCP, UDP, or ICMP segment's Checksum field. This bit is also set when the received number of payload bytes does not match the value indicated in the Length field of the encapsulated IPv4 or IPv6 datagram in the received Ethernet frame. See Table 6.34-20 for more details.

Table 6.34-19 Receive Descriptor 0

When the Full Checksum Offload Engine (Type 2) is selected and it is enabled by setting Bit 10 (IPC) of Register 0 (MAC Configuration Register), the permutations of Bits 5, 7, and 0 reflect the conditions described in Table 6.34-20.

When the Full Checksum Offload Engine (Type 2) is selected and Bit 10 (IPC) of Register 0 (MAC Configuration Register) is reset, only the conditions shown in first two rows of Table 6.34-20 are valid. However, the frame status changes for the second condition as described in Table 6.34-20.

Bit 5: Frame Type	Bit 7: IPC Checksum Error	Bit 0: Payload Checksum Error	Frame Status
0	0	0	IEEE 802.3 Type frame (Length field value is less than 1,536) **Note:** This frame status is also valid when the Full Checksum Offload Engine (Type 2) is selected and Bit 10 (IPC) of Register 0 (MAC Configuration Register) is reset.
1	0	0	■ Frame status when the Full Checksum Offload Engine (Type 2) is selected and Bit 10 (IPC) is set: IPv4/IPv6 Type frame in which no checksum error is detected. ■Frame status when the Full Checksum Offload Engine (Type 2) is selected and Bit 10 (IPC) is reset: The frame is an IEEE 802.3 Type frame (Length field value is greater than or equal to 1,536).
1	0	1	IPv4/IPv6 Type frame with a payload checksum error (as described for PCE) detected
1	1	0	IPv4/IPv6 Type frame with an IP header checksum error (as described for IPC CE) detected
1	1	1	IPv4/IPv6 Type frame with both IP header and payload checksum errors detected
0	0	1	IPv4/IPv6 Type frame with no IP header checksum error and the payload check bypassed, due to an unsupported payload
0	1	1	A Type frame that is neither IPv4 or IPv6 (the Checksum Offload engine bypasses checksum completely.)
0	1	0	Reserved

Table 6.34-20 Receive Descriptor 0 when COE (Type 2) is Enabled

Receive Descriptor 1 (RDES1)

RDES1 contains the buffer sizes and other bits that control the descriptor chain or ring.

Bit	Description
31	DIC: Disable Interrupt on Completion When set, this bit prevents the setting of the bit 6 (RI) of the Register 1005 (Status Register) for the received frame that ends in the buffer pointed to by this descriptor. This, in turn, disables the assertion of the interrupt to Host due to RI for that frame. **Note:** This bit is valid only when the last descriptor bit (RDES0[8]) is set.
30:26	Reserved
25	RER: Receive End of Ring When set, this bit indicates that the descriptor list reached its final descriptor. The DMA returns to the base address of the list, creating a Descriptor Ring.
24	RCH: Second Address Chained When set, this bit indicates that the second address in the descriptor is the Next Descriptor address rather than the second buffer address. When RDES1[24] is set, RBS2 (RDES1[21-11]) is a “don’t care” value. RDES1[25] takes precedence over RDES1[24].
23:22	Reserved
21:11	RBS2: Receive Buffer 2 Size These bits indicate the second data buffer size in bytes. The buffer size must be a multiple of 4, 8, or 16 depending upon the bus widths (32, 64, or 128), even if the value of RDES3 (buffer2 address pointer) is not aligned to bus width. In the case where the buffer size is not a multiple of 4, 8, or 16, the resulting behavior is undefined. This field is not valid if RDES1[24] is set.
10:0	RBS1: Receive Buffer 1 Size Indicates the first data buffer size in bytes. The buffer size must be a multiple of 4, 8, or 16 depending upon the bus widths (32, 64, or 128), even if the value of RDES2 (buffer1 address pointer) is not aligned. In the case where the buffer size is not a multiple of 4, 8, or 16, the resulting behavior is undefined. If this field is 0, the DMA ignores this buffer and uses Buffer 2 or next descriptor depending on the value of RCH (Bit 24).

Table 6.34-21 Receive Descriptor 1

Receive Descriptor 2 (RDES2)

RDES2 contains the address pointer to the first data buffer in the descriptor.

Bit	Description
31:0	Buffer 1 Address Pointer These bits indicate the physical address of Buffer 1. There are no limitations on the buffer address alignment except for the following condition: The DMA uses the configured value for its address generation when the RDES2 value is used to store the start of frame. The DMA performs a write operation with the RDES2[3:0, 2:0, or 1:0] bits as 0 during the transfer of the start of frame but the frame data is shifted as per the actual Buffer address pointer. The DMA ignores RDES2[3:0, 2:0, or 1:0], corresponding to bus width of 128, 64, or 32, if the address pointer is to a buffer where the middle or last part of the frame is stored.

Table 6.34-22 Receive Descriptor 2

Receive Descriptor 3 (RDES3)

RDES3 contains the address pointer either to the second data buffer in the descriptor or to the next descriptor.

Bit	Description

31:0	Buffer 2 Address Pointer (Next Descriptor Address) These bits indicate the physical address of Buffer 2 when a descriptor ring structure is used. If the Second Address Chained (RDES1[24]) bit is set, this address contains the pointer to the physical memory where the Next Descriptor is present. If RDES1[24] is set, the buffer (Next Descriptor) address pointer must be bus width-aligned (RDES3[3, 2, or 1:0] = 0, corresponding to a bus width of 128, 64, or 32. LSBs are ignored internally.) However, when RDES1[24] is reset, there are no limitations on the RDES3 value, except for the following condition: The DMA uses the configured value for its buffer address generation when the RDES3 value is used to store the start of frame. The DMA ignores RDES3[3:0, 2:0, or 1:0], corresponding to a bus width of 128, 64, or 32, if the address pointer is to a buffer where the middle or last part of the frame is stored.

Table 6.34-23 Receive Descriptor 3

Descriptor Format With IEEE 1588-2005 Timestamps Enabled

The default descriptor format (as described in “Transmit Descriptor” and “Receive Descriptor”), and field descriptions remain unchanged when created by software (Own bit is set in DES0). However, if the software has enabled IEEE 1588-2005 functionality, the DES2 and DES3 descriptor fields (see Figure 6.34-21) take on a different meaning when the DMA closes the descriptor (Own bit in DES0 is cleared).

The DMA updates the DES2 and DES3 with the timestamp value before clearing the Own bit in DES0.

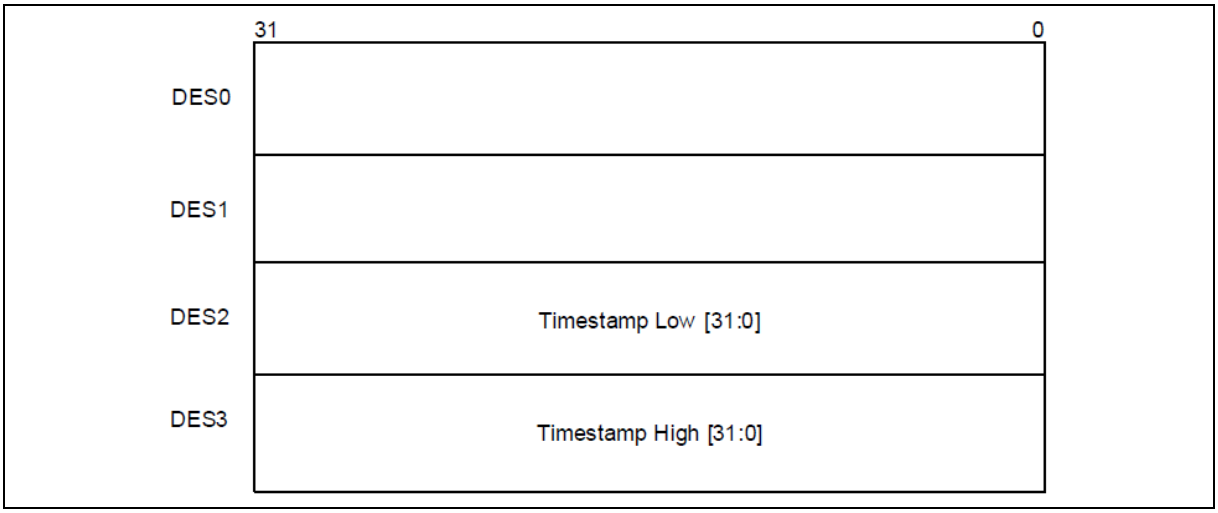


Figure 6.34-21 Descriptor Fields when DMA Clears the Own Bit

DES2 is updated with TSL and DES3 is updated with TSH. This ensures that the 64-bit timestamp can be processed as it is without any swapping between the two 32-bit words.

The following sections describe the details specific to receive and transmit descriptors in this mode.

Transmit Descriptor

In addition to the changes described in “Descriptor Format With IEEE 1588-2005 Timestamps Enabled”, the Transmit descriptor has additional control and status bits (TTSE and TTSS, respectively) for timestamping, as shown in Figure 6.34-22. Software sets the TTSE bit (when the Own bit is set), instructing the core to generate a timestamp for the corresponding Ethernet frame being transmitted. The DMA sets the TTSS bit if the timestamp has been updated in the TDES2 and TDES3 fields when the descriptor is closed (Own bit is cleared).

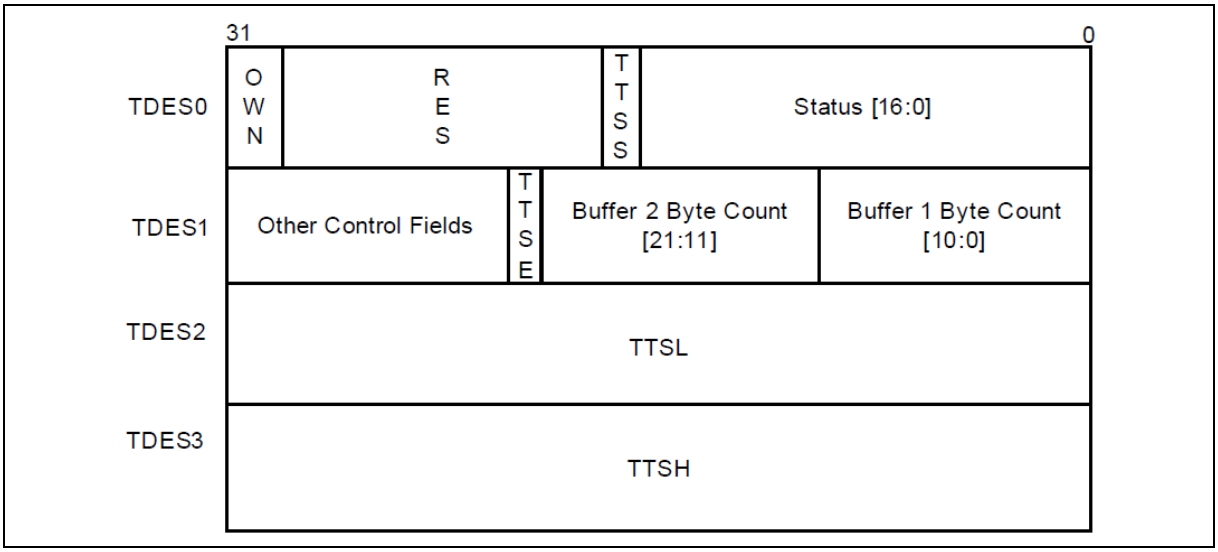


Figure 6.34-22 Descriptor Fields when DMA Clears the Own Bit

Transmit Descriptor: Transmit Timestamp Control and Status Fields

The position of these fields is different for normal transmit descriptor and enhanced format transmit descriptor. The value of this field in both the cases shall be preserved by the DMA at the time of closing the descriptor.

Updates to TDES0 and TDES1 for the default (normal) descriptor format are described in following tables.

Bit	Description
17	TTSS: Transmit Timestamp Status This field is a status bit indicating that a timestamp was captured for the corresponding transmit frame. When this bit is set, both TDES2 and TDES3 have a timestamp value that was captured for the transmit frame. This field is valid only when the Last Segment control bit (TDES1[30] in the descriptor) is set.

Table 6.34-24 Transmit Timestamp Status – Normal Descriptor Format Case (TDES0)

Bit	Description
22	TTSE: Transmit Timestamp Enable When set, this field enables IEEE1588 hardware timestamping for the transmit frame described by the descriptor. This field is valid only when the First Segment control bit (TDES1[29] in the descriptor) is set.

Table 6.34-25 Transmit Timestamp Control – Normal Descriptor Format Case (TDES1)

Transmit Descriptor: Transmit Timestamp Field

The transmit descriptor format and field descriptions remain unchanged when they are created by software (when the Own bit is set). However, when the DMA closes the last descriptor (marked, in the alternative descriptor format, by the LS bit in TDES1 or TDES0) and IEEE 1588 functionality is enabled (the Own bit is cleared), the TDES2 and TDES3 descriptor fields are updated with the timestamp, if taken, for that frame.

The fields in TDES2 and TDES3 are swapped when the core operates in 64-bit or 128-bit and with the descriptor in reverse-endian mode

Tables Table 6.34-26 and Table 6.34-27 describe the fields that have different meaning when the descriptor is closed.

Bit	Description
31:0	TTSL: Transmit Frame Timestamp Low This field is updated by DMA with the least significant 32 bits of the timestamp captured for the corresponding transmit frame. This field has the timestamp only if the Last Segment control bit (LS) in the descriptor is set.

Table 6.34-26 Transmit Descriptor Fields (TDES2)

Bit	Description
31:0	TTSH: Transmit Frame Timestamp High This field is updated by DMA with the most significant 32 bits of the timestamp captured for the corresponding transmit frame. This field has the timestamp only if the Last Segment control bit (LS) in the descriptor is set.

Table 6.34-27 Transmit Descriptor Fields (TDES3)

Receive Descriptor

Receive Descriptor: Receive Timestamp

Figure 6.34-23 shows the format of receive descriptor when timestamping is enabled.

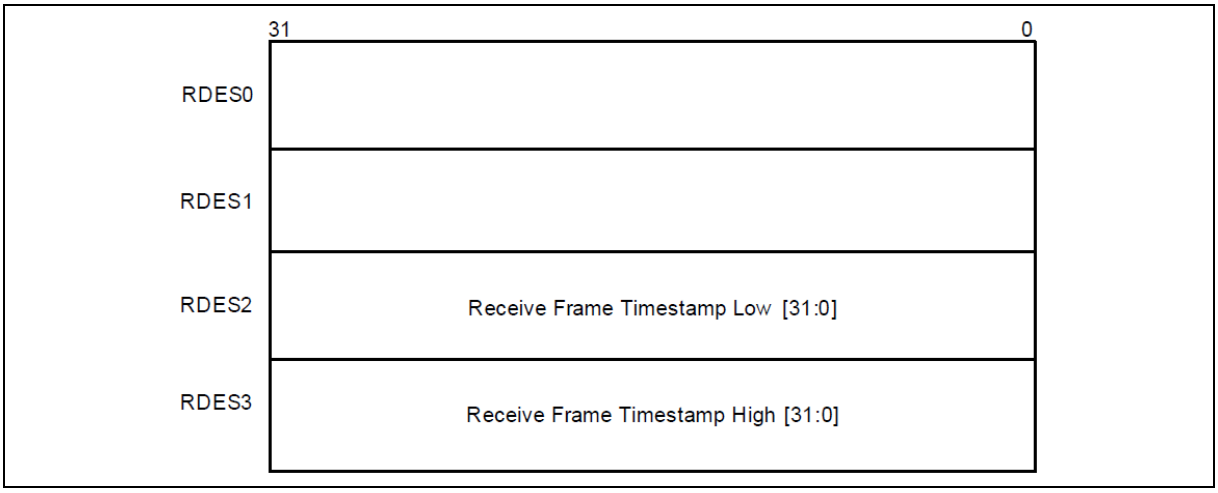


Figure 6.34-23 Receive Timestamp Fields

Table 6.34-28 and Table 6.34-29 describe the fields that have different meaning for RDES2 and RDES3 when the receive descriptor is closed and timestamping is enabled.

When software disables the Timestamp feature (the TSENA bit in Register 448 is low), the DMA does *not* update the RDES2 or RDES3 fields of descriptor before closing the RDES0.

Bit	Description
31:0	RTSL: Receive Frame Timestamp Low The DMA updates this field with the least significant 32 bits of the timestamp captured for the corresponding receive frame. The DMA updates this field only for the last descriptor of the receive frame indicated by Last Descriptor status bit (RDES0[8]). When this field and the RTSH field in RDES3 show an all-ones value, the timestamp must be treated as corrupt.

Table 6.34-28 Receive Descriptor Fields (RDES2)

Bit	Description

31:0	RTSH: Receive Frame Timestamp High The DMA updates this field with the most significant 32 bits of the timestamp captured for the corresponding receive frame. The DMA updates this field only for the last descriptor of the receive frame indicated by Last Descriptor status bit (RDES0[8]). When this field and RDES2's RTSL field show all-ones values, the timestamp must be treated as corrupt.

Table 6.34-29 Receive Descriptor Fields (RDES3)

6.34.6.4 Alternate or Enhanced Descriptors

The alternate (or enhanced) descriptor structure can have 8 DWORDS (32-bytes) instead of the 4 DWORDS as in the case of normal descriptor format. The alternate (or enhanced) descriptor structure is the only supported format when the IEEE 1588-2008 Advanced Timestamp feature, or SA, VLAN, and CRC Insertion feature is enabled. The features of the alternate descriptor structure are:

- The normal descriptor structure allows data buffers of up to 2,048 bytes. The alternative descriptor structure is implemented to support buffers of up to 8 Kbytes (useful for Jumbo frames).
- There is a re-assignment of control and status bits in TDES0, TDES1, RDES0 (Advanced timestamp or IPC full offload configuration), and RDES1.
- The transmit descriptor stores the timestamp in TDES6 and TDES7 when you select the Advanced Timestamp.
- This receive descriptor structure is also used for storing the extended status (RDES4) and timestamp (RDES6 and RDES7) when Advanced Timestamp or IPC Full Checksum Offload Engine feature is selected.
- You can select one of the following options for descriptor structure:
  - If timestamping is enabled in Register 448 (Timestamp Control Register) or Checksum Offload is enabled in Register 0 (MAC Configuration Register), the software needs to allocate 32-bytes (8 DWORDS) of memory for every descriptor. For this, the software should set Bit 7 (Alternate Descriptor Size) of Register 1000 (Bus Mode Register).
  - If timestamping or Checksum Offload is not enabled, the extended descriptors (DES4 to DES7) are not required. Therefore, the software can use alternate descriptors with the default size of 16 bytes (4 DWORDS). For this, the software should reset Bit 7 (Alternate Descriptor Size) of Register 1000 (Bus Mode Register) to 0.
- When alternate descriptor is selected without Timestamp or Receive IPC Full Checksum Offload Engine (Type 2) feature, the descriptor size is always 4 DWORDs (DES0-DES3). Therefore, the software can use alternate descriptors with the default size of 16 bytes.

Transmit Descriptor

The transmit descriptor structure is shown in Figure 6.34-24. The application software must program the control bits TDES0[31:18] during descriptor initialization. When the DMA updates the descriptor, it write backs all the control bits except the OWN bit (which it clears) and updates the status bits[7:0]. The contents of the transmitter descriptor word 0 (TDES0) through word 3 (TDES3) are given in Tables Table 6.34-30 through Table 6.34-33, respectively.

With the advance timestamp support, the snapshot of the timestamp to be taken can be enabled for a given frame by setting Bit 25 (TTSE) of TDES0. When the descriptor is closed (that is, when the OWN bit is cleared), the timestamp is written into TDES6 and TDES7. This is indicated by the status Bit 17 (TTSS) of TDES0 shown in Figure 6.34-24. The contents of TDES6 and TDES7 are mentioned in Table 6.34-34 and Table 6.34-35.

When Advanced Timestamp feature is enabled, the software should set Bit 7 of Register 1000 (Bus Mode Register), so that the DMA operates with extended descriptor size. When this control bit is reset, the TDES4-TDES7 descriptor space is not valid.

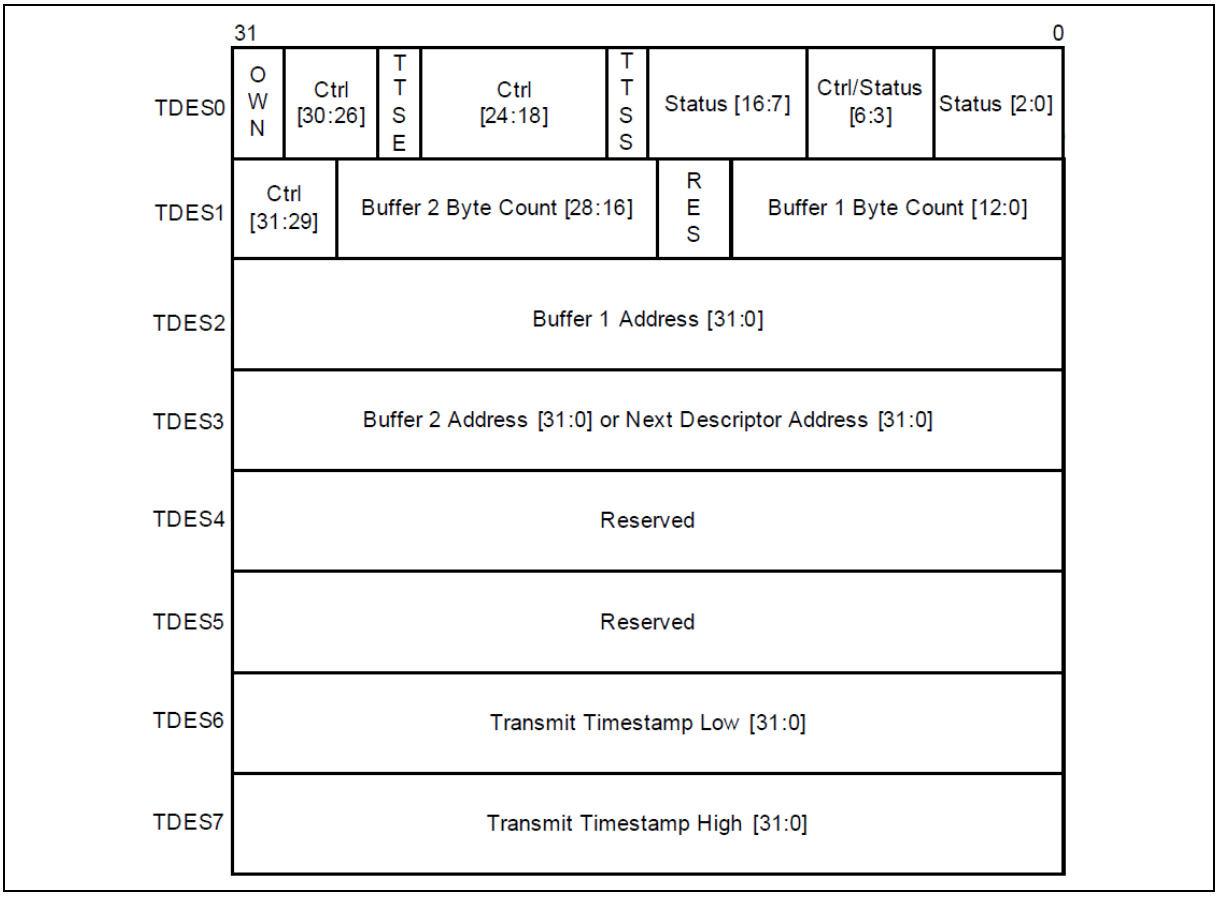


Figure 6.34-24 Transmit Descriptor Fields - Alternate (Enhanced) Format

The DMA always reads or fetches four DWORDS of the descriptor from system memory to obtain the buffer and control information as shown in Figure 6.34-25.

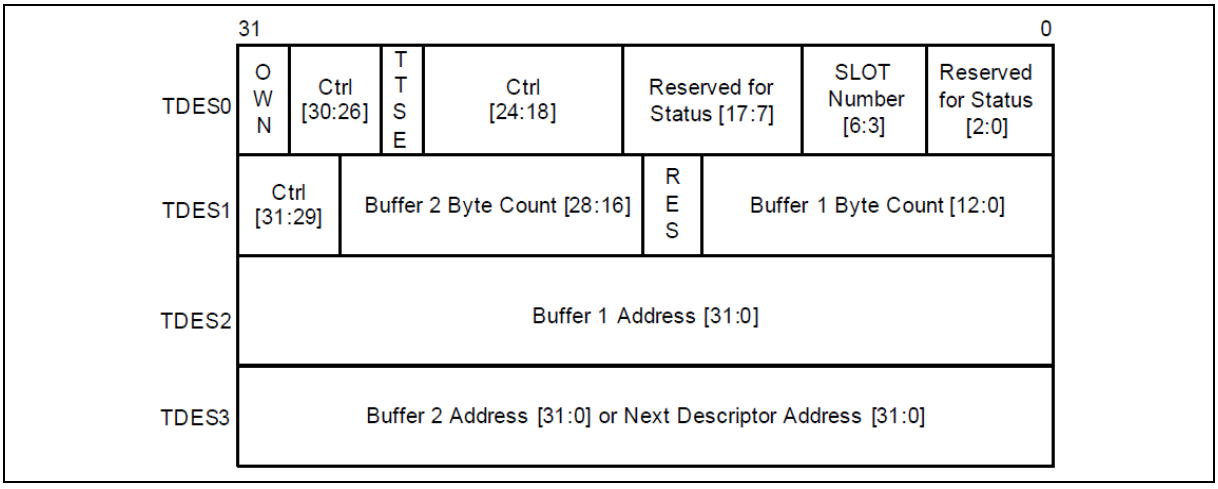


Figure 6.34-25 Transmit Descriptor Fetch (Read) for Alternate (Enhanced) Format

Bit	Description
31	OWN: Own Bit When set, this bit indicates that the descriptor is owned by the DMA. When this bit is reset, it indicates that the descriptor

	is owned by the Host. The DMA clears this bit either when it completes the frame transmission or when the buffers allocated in the descriptor are read completely. The ownership bit of the frame's first descriptor must be set after all subsequent descriptors belonging to the same frame have been set. This avoids a possible race condition between fetching a descriptor and the driver setting an ownership bit.
30	IC: Interrupt on Completion When set, this bit sets the Transmit Interrupt (Register 5[0]) after the present frame has been transmitted. This bit is valid only when the last segment bit (TDES0[29]) is set.
29	LS: Last Segment When set, this bit indicates that the buffer contains the last segment of the frame. When this bit is set, the TBS1 or TBS2 field in TDES1 should have a non-zero value.
28	FS: First Segment When set, this bit indicates that the buffer contains the first segment of a frame.
27	DC: Disable CRC When this bit is set, the MAC does not append a cyclic redundancy check (CRC) to the end of the transmitted frame. This is valid only when the first segment (TDES0[28]) is set.
26	DP: Disable Pad When set, the MAC does not automatically add padding to a frame shorter than 64 bytes. When this bit is reset, the DMA automatically adds padding and CRC to a frame shorter than 64 bytes, and the CRC field is added despite the state of the DC (TDES0[27]) bit. This is valid only when the first segment (TDES0[28]) is set.
25	TTSE: Transmit Timestamp Enable When set, this bit enables IEEE1588 hardware timestamping for the transmit frame referenced by the descriptor. This field is valid only when the First Segment control bit (TDES0[28]) is set.
24	CRCR: CRC Replacement Control When set, the MAC replaces the last four bytes of the transmitted packet with recalculated CRC bytes. The host should ensure that the CRC bytes are present in the frame being transferred from the Transmit Buffer. This bit is valid when the First Segment control bit (TDES0[28]) is set. In addition, CRC replacement is done only when Bit 27 (DC) is set to 1.
23:22	CIC: Checksum Insertion Control These bits control the checksum calculation and insertion. The following list describes the bit encoding: 2'b00 = Checksum Insertion Disabled. 2'b01 = Only IP header checksum calculation and insertion are enabled. 2'b10 = IP header checksum and payload checksum calculation and insertion are enabled, but pseudo- header checksum is not calculated in hardware. 2'b11 = IP Header checksum and payload checksum calculation and insertion are enabled, and pseudo- header checksum is calculated in hardware. This field is valid when the First Segment control bit (TDES0[28]) is set.
21	TER: Transmit End of Ring When set, this bit indicates that the descriptor list reached its final descriptor. The DMA returns to the base address of the list, creating a descriptor ring.
20	TCH: Second Address Chained When set, this bit indicates that the second address in the descriptor is the Next Descriptor address rather than the second buffer address. When TDES0[20] is set, TBS2 (TDES1[28:16]) is a "don't care" value. TDES0[21] takes precedence over TDES0[20].
19:18	VLIC: VLAN Insertion Control When set, these bits request the MAC to perform VLAN tagging or untagging before transmitting the frames. If the frame is modified for VLAN tags, the MAC automatically recalculates and replaces the CRC bytes. The following list describes the values of these bits: 2'b00 = Do not add a VLAN tag. 2'b01 = Remove the VLAN tag from the frames before transmission. This option should be used only with the VLAN frames. 2'b10 = Insert a VLAN tag with the tag value programmed in Register 353 (VLAN Tag Inclusion or Replacement

	Register). 2'b11 = Replace the VLAN tag in frames with the Tag value programmed in Register 353 (VLAN Tag Inclusion or Replacement Register). This option should be used only with the VLAN frames. These bits are valid when the First Segment control bit (TDES0[28]) is set.
17	TTSS: Transmit Timestamp Status This field is used as a status bit to indicate that a timestamp was captured for the described transmit frame. When this bit is set, TDES2 and TDES3 have a timestamp value captured for the transmit frame. This field is only valid when the descriptor's Last Segment control bit (TDES0[29]) is set.
16	IHE: IP Header Error When set, this bit indicates that the MAC transmitter detected an error in the IP datagram header. The transmitter checks the header length in the IPv4 packet against the number of header bytes received from the application and indicates an error status if there is a mismatch. For IPv6 frames, a header error is reported if the main header length is not 40 bytes. Furthermore, the Ethernet Length/Type field value for an IPv4 or IPv6 frame must match the IP header version received with the packet. For IPv4 frames, an error status is also indicated if the Header Length field has a value less than 0x5. This bit is valid only when the Tx Checksum Offload is enabled. Otherwise, it is reserved. If COE detects an IP header error, it still inserts an IPv4 header checksum if the Ethernet Type field indicates an IPv4 payload.
15	ES: Error Summary Indicates the logical OR of the following bits: TDES0[14] = Jabber Timeout TDES0[13] = Frame Flush TDES0[11] = Loss of Carrier TDES0[10] = No Carrier TDES0[9] = Late Collision TDES0[8] = Excessive Collision TDES0[2] = Excessive Deferral TDES0[1] = Underflow Error TDES0[16] = IP Header Error TDES0[12] = IP Payload Error
14	JT: Jabber Timeout When set, this bit indicates the MAC transmitter has experienced a jabber timeout. This bit is only set when Bit 22 (Jabber Disable) of Register 0 (MAC Configuration Register) is not set.
13	FF: Frame Flushed When set, this bit indicates that the DMA or MTL flushed the frame because of a software Flush command given by the CPU.
12	IPE: IP Payload Error When set, this bit indicates that MAC transmitter detected an error in the TCP, UDP, or ICMP IP datagram payload. The transmitter checks the payload length received in the IPv4 or IPv6 header against the actual number of TCP, UDP, or ICMP packet bytes received from the application and issues an error status in case of a mismatch.
11	LOC: Loss of Carrier When set, this bit indicates that a loss of carrier occurred during frame transmission (that is, the phy_crs_i signal was inactive for one or more transmit clock periods during frame transmission). This is valid only for the frames transmitted without collision when the MAC operates in the half-duplex mode.
10	NC: No Carrier When set, this bit indicates that the Carrier Sense signal from the PHY was not asserted during transmission. **Note:** When any optional PHY interface is enabled, the carrier errors (NC or LOC) cannot be generated. This is because the CRS signal is internally-generated based on the internal MII TX_EN and RX_DV derived from the optional PHY interface signals.
9	LC: Late Collision When set, this bit indicates that frame transmission is aborted because of a collision occurring after the collision window (64 byte-times, including preamble, in MII mode and 512 byte-times, including preamble and carrier extension, in GMII mode). This bit is not valid if the Underflow Error bit is set.

8	EC: Excessive Collision When set, this bit indicates that the transmission was aborted after 16 successive collisions while attempting to transmit the current frame. If Bit 9 (Disable Retry) bit in the Register 0 (MAC Configuration Register) is set, this bit is set after the first collision, and the transmission of the frame is aborted.
7	VF: VLAN Frame When set, this bit indicates that the transmitted frame is a VLAN-type frame.
6:3	CC: Collision Count (Status field) These status bits indicate the number of collisions that occurred before the frame was transmitted. This count is not valid when the Excessive Collisions bit (TDES0[8]) is set. The core updates this status field only in the half-duplex mode.
2	ED: Excessive Deferral When set, this bit indicates that the transmission has ended because of excessive deferral of over 24,288 bit times (155,680 bits times in 1,000-Mbps mode or if Jumbo Frame is enabled) if Bit 4 (Deferral Check) bit in Register 0 (MAC Configuration Register) is set high.
1	UF: Underflow Error When set, this bit indicates that the MAC aborted the frame because the data arrived late from the Host memory. Underflow Error indicates that the DMA encountered an empty transmit buffer while transmitting the frame. The transmission process enters the Suspended state and sets both Transmit Underflow (Register 5[5]) and Transmit Interrupt (Register 5[0]).
0	DB: Deferred Bit When set, this bit indicates that the MAC defers before transmission because of the presence of carrier. This bit is valid only in the half-duplex mode.

Table 6.34-30 Transmit Descriptor Word 0 (TDES0)

Bit	Description
31:29	SAIC: SA Insertion Control These bits request the MAC to add or replace the Source Address field in the Ethernet frame with the value given in the MAC Address 0 register. If the Source Address field is modified in a frame, the MAC automatically recalculates and replaces the CRC bytes. The Bit 31 specifies the MAC Address Register (1 or 0) value that is used for Source Address insertion or replacement. The following list describes the values of Bits[30:29]: 2'b00 = Do not include the source address. 2'b01 = Include or insert the source address. For reliable transmission, the application must provide frames without source addresses. 2'b10 = Replace the source address. For reliable transmission, the application must provide frames with source addresses. 2'b11 = Reserved These bits are valid when the First Segment control bit (TDES0[28]) is set.
28:16	TBS2: Transmit Buffer 2 Size This field indicates the second data buffer size in bytes. This field is not valid if TDES0[20] is set.
15:13	Reserved
12:0	TBS1: Transmit Buffer 1 Size These bits indicate the first data buffer byte size, in bytes. If this field is 0, the DMA ignores this buffer and uses Buffer 2 or the next descriptor, depending on the value of TCH (TDES0[20]).

Table 6.34-31 Transmit Descriptor Word 1 (TDES1)

Bit	Description

31:0	Buffer 1 Address Pointer These bits indicate the physical address of Buffer 1. There is no limitation on the buffer address alignment. For more information about buffer address alignment, see “Host Data Buffer Alignment”.

Table 6.34-32 Transmit Descriptor 2 (TDES2)

Bit	Description
31:0	Buffer 2 Address Pointer (Next Descriptor Address) Indicates the physical address of Buffer 2 when a descriptor ring structure is used. If the Second Address Chained (TDES1[24]) bit is set, this address contains the pointer to the physical memory where the Next Descriptor is present. The buffer address pointer must be aligned to the bus width only when TDES1[24] is set. (LSBs are ignored internally.)

Table 6.34-33 Transmit Descriptor 3 (TDES3)

Bit	Description
31:0	TTSL: Transmit Frame Timestamp Low This field is updated by DMA with the least significant 32 bits of the timestamp captured for the corresponding transmit frame. This field has the timestamp only if the Last Segment bit (LS) in the descriptor is set and Timestamp status (TTSS) bit is set.

Table 6.34-34 Transmit Descriptor 6 (TDES6)

Bit	Description
31:0	TTSH: Transmit Frame Timestamp High This field is updated by DMA with the most significant 32 bits of the timestamp captured for the corresponding receive frame. This field has the timestamp only if the Last Segment bit (LS) in the descriptor is set and Timestamp status (TTSS) bit is set.

Table 6.34-35 Transmit Descriptor 7 (TDES7)

Receive Descriptor

The structure of the received descriptor is shown in Figure 6.34-26. This can have 32 bytes of descriptor data (8 DWORDs) when Advanced Timestamp or IPC Full Offload feature is selected.

When either of these features is enabled, the Software should set Bit 7 of Register 1000 (Bus Mode Register) so that the DMA operates with extended descriptor size. When this control bit is reset, the RDES0[0] is always cleared and the RDES4-RDES7 descriptor space is not valid.

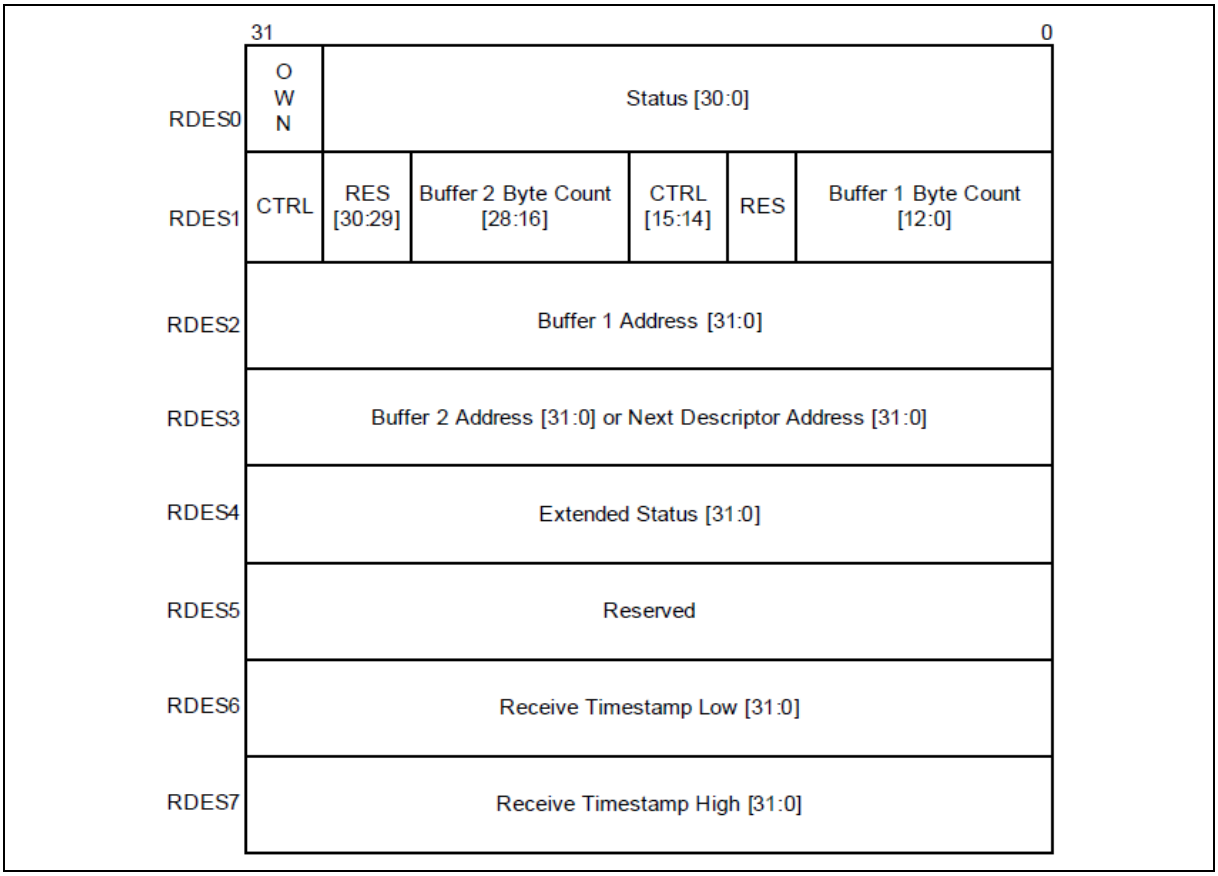


Figure 6.34-26 Receive Descriptor Fields - Alternate (Enhanced) Format

The contents of RDES0 are identified in Table 6.34-36. The contents of RDES1 through RDES3 are identified in Tables Table 6.34-37 through Table 6.34-39, respectively.

The status bits are valid when the LS bit is set. Therefore, when multiple descriptors are used for a single frame, status bits of only last descriptor (that is, descriptor with LS bit set) are valid.

Bit	Description
31	OWN: Own Bit When set, this bit indicates that the descriptor is owned by the DMA of the GMAC. When this bit is reset, this bit indicates that the descriptor is owned by the Host. The DMA clears this bit either when it completes the frame reception or when the buffers that are associated with this descriptor are full.
30	AFM: Destination Address Filter Fail When set, this bit indicates a frame that failed in the DA Filter in the MAC.
29:16	FL: Frame Length These bits indicate the byte length of the received frame that was transferred to host memory. This field is valid when Last Descriptor (RDES0[8]) is set and either the Descriptor Error (RDES0[14]) or Overflow Error bits are reset. The frame length also includes the two bytes appended to the Ethernet frame when IP checksum calculation (Type 1) is enabled and the received frame is not a MAC control frame. This field is valid when Last Descriptor (RDES0[8]) is set. When the Last Descriptor and Error Summary bits are not set, this field indicates the accumulated number of bytes that have been transferred for the current frame. The inclusion of CRC length in the frame length depends on the settings of Bit 7 and Bit 25 in Register 0 (MAC Configuration Register).
15	ES: Error Summary Indicates the logical OR of the following bits: RDES0[1] = CRC Error

	RDES0[3] = Receive Error RDES0[4] = Watchdog Timeout RDES0[6] = Late Collision RDES0[7] = Giant Frame RDES4[4:3] = IP Header or Payload Error RDES0[11] = Overflow Error RDES0[14] = Descriptor Error This field is valid only when the Last Descriptor (RDES0[8]) is set.
14	DE: Descriptor Error When set, this bit indicates a frame truncation caused by a frame that does not fit within the current descriptor buffers, and that the DMA does not own the Next Descriptor. The frame is truncated. This field is valid only when the Last Descriptor (RDES0[8]) is set.
13	SAF: Source Address Filter Fail When set, this bit indicates that the SA field of frame failed the SA Filter in the MAC.
12	LE: Length Error When set, this bit indicates that the actual length of the frame received and that the Length/ Type field does not match. This bit is valid only when the Frame Type (RDES0[5]) bit is reset.
11	OE: Overflow Error When set, this bit indicates that the received frame was damaged because of buffer overflow in MTL. **Note:** This bit is set only when the DMA transfers a partial frame to the application. This happens only when the Rx FIFO is operating in the threshold mode. In the store-and-forward mode, all partial frames are dropped completely in Rx FIFO.
10	VLAN: VLAN Tag When set, this bit indicates that the frame to which this descriptor is pointing is a VLAN frame tagged by the MAC. The VLAN tagging depends on checking the VLAN fields of received frame based on the Register 7 (VLAN Tag Register) setting.
9	FS: First Descriptor When set, this bit indicates that this descriptor contains the first buffer of the frame. If the size of the first buffer is 0, the second buffer contains the beginning of the frame. If the size of the second buffer is also 0, the next Descriptor contains the beginning of the frame.
8	LS: Last Descriptor When set, this bit indicates that the buffers pointed to by this descriptor are the last buffers of the frame
7	Timestamp Available, IP Checksum Error (Type1), or Giant Frame When Advanced Timestamp feature is present, when set, this bit indicates that a snapshot of the Timestamp is written in descriptor words 6 (RDES6) and 7 (RDES7). This is valid only when the Last Descriptor bit (RDES0[8]) is set. When IP Checksum Engine (Type 1) is selected, this bit, when set, indicates one of the following: ■The 16-bit IPv4 header checksum calculated by the core did not match the received checksum bytes. ■The header checksum checking is bypassed for non-IPv4 frames. Otherwise, this bit, when set, indicates the Giant Frame Status. Giant frames are larger than 1,518-byte (or 1,522-byte for VLAN or 2,000-byte when Bit 27 of MAC Configuration register is set) normal frames and larger than 9,018-byte (9,022-byte for VLAN) frame when Jumbo Frame processing is enabled.
6	LC: Late Collision When set, this bit indicates that a late collision has occurred while receiving the frame in the half-duplex mode.
5	FT: Frame Type When set, this bit indicates that the Receive Frame is an Ethernet-type frame (the LT field is greater than or equal to 1,536). When this bit is reset, it indicates that the received frame is an IEEE 802.3 frame. This bit is not valid for Runt frames less than 14 bytes.
4	RWT: Receive Watchdog Timeout When set, this bit indicates that the Receive Watchdog Timer has expired while receiving the current frame and the current frame is truncated after the Watchdog Timeout.

3	RE: Receive Error When set, this bit indicates that the gmii_rxer_i signal is asserted while gmii_rxdv_i is asserted during frame reception. This error also includes carrier extension error in the GMII and half-duplex mode. Error can be of less or no extension, or error (rxd ≠ 0f) during extension.
2	DE: Dribble Bit Error When set, this bit indicates that the received frame has a non-integer multiple of bytes (odd nibbles). This bit is valid only in the MII Mode.
1	CE: CRC Error When set, this bit indicates that a Cyclic Redundancy Check (CRC) Error occurred on the received frame. This field is valid only when the Last Descriptor (RDES0[8]) is set.
0	Extended Status Available/Rx MAC Address When either Advanced Timestamp or IP Checksum Offload (Type 2) is present, this bit, when set, indicates that the extended status is available in descriptor word 4 (RDES4). This is valid only when the Last Descriptor bit (RDES0[8]) is set. This bit is invalid when Bit 30 is set. When IP Checksum Offload (Type 2) is present, this bit is set even when IP Checksum Offload engine bypasses the processing of received frame. The bypassing may be because of non-IP frame or IP frame with non-TCP/UDP/ICMP payload. When Advance Timestamp Feature or IPC Full Offload is not selected, this bit indicates Rx MAC Address status. When set, this bit indicates that the Rx MAC Address registers value (1 to 15) matched the frame's DA field. When reset, this bit indicates that the Rx MAC Address Register 0 value matched the DA field.

Table 6.34-36 Receive Descriptor Fields 0 (RDES0)

Bit	Description
31	DIC: Disable Interrupt on Completion When set, this bit prevents setting the Status Register's RI bit (CSR5[6]) for the received frame ending in the buffer indicated by this descriptor. This, in turn, disables the assertion of the interrupt to Host because of RI for that frame. **Note:** This bit is valid only when the last descriptor bit (RDES0[8]) is set.
30:29	Reserved
28:16	RBS2: Receive Buffer 2 Size These bits indicate the second data buffer size, in bytes. The buffer size must be a multiple of 4, 8, or 16, depending on the bus widths (32, 64, or 128, respectively), even if the value of RDES3 (buffer2 address pointer) is not aligned to bus width. If the buffer size is not an appropriate multiple of 4, 8, or 16, the resulting behavior is undefined. This field is not valid if RDES1[14] is set.
15	RER: Receive End of Ring When set, this bit indicates that the descriptor list reached its final descriptor. The DMA returns to the base address of the list, creating a descriptor ring.
14	RCH: Second Address Chained When set, this bit indicates that the second address in the descriptor is the Next Descriptor address rather than the second buffer address. When this bit is set, RBS2 (RDES1[28:16]) is a "don't care" value. RDES1[15] takes precedence over RDES1[14].
13	Reserved
12:0	RBS1: Receive Buffer 1 Size Indicates the first data buffer size in bytes. The buffer size must be a multiple of 4, 8, or 16, depending upon the bus widths (32, 64, or 128), even if the value of RDES2 (buffer1 address pointer) is not aligned. When the buffer size is not a multiple of 4, 8, or 16, the resulting behavior is undefined. If this field is 0, the DMA ignores this buffer and uses Buffer 2 or next descriptor depending on the value of RCH (Bit 14).

Table 6.34-37 Receive Descriptor Fields 1 (RDES1)

Bit	Description

31:0	Buffer 1 Address Pointer These bits indicate the physical address of Buffer 1. There are no limitations on the buffer address alignment except for the following condition: The DMA uses the configured value for its address generation when the RDES2 value is used to store the start of frame. The DMA performs a write operation with the RDES2[3:0, 2:0, or 1:0] bits as 0 during the transfer of the start of frame but the frame data is shifted as per the actual Buffer address pointer. The DMA ignores RDES2[3:0, 2:0, or 1:0] (corresponding to bus width of 128, 64, or 32) if the address pointer is to a buffer where the middle or last part of the frame is stored.

Table 6.34-38 Receive Descriptor Fields 2 (RDES2)

Bit	Description
31:0	Buffer 2 Address Pointer (Next Descriptor Address) These bits indicate the physical address of Buffer 2 when a descriptor ring structure is used. If the Second Address Chained (RDES1[24]) bit is set, this address contains the pointer to the physical memory where the Next Descriptor is present. If RDES1[24] is set, the buffer (Next Descriptor) address pointer must be bus width-aligned (RDES3[3, 2, or 1:0]::0, corresponding to a bus width of 128, 64, or 32. LSBs are ignored internally.) However, when RDES1[24] is reset, there are no limitations on the RDES3 value, except for the following condition: The DMA uses the configured value for its buffer address generation when the RDES3 value is used to store the start of frame. The DMA ignores RDES3 [3, 2, or 1:0] (corresponding to a bus width of 128, 64, or 32) if the address pointer is to a buffer where the middle or last part of the frame is stored.

Table 6.34-39 Receive Descriptor Fields 3 (RDES3)

The extended status written is as shown in Table 6.34-40. The extended status is written only when there is status related to IPC or timestamp available. The availability of extended status is indicated by Bit 0 of RDES0. This status is available only when the Advance Timestamp or IPC Full Offload feature is selected.

Bit	Description
31:28	Reserved
27:26	Layer 3 and Layer 4 Filter Number Matched These bits indicate the number of the Layer 3 and Layer 4 Filter that matched the received frame. 00 = Filter 0 01 = Filter 1 10 = Filter 2 11 = Filter 3 This field is valid only when Bit 24 or Bit 25 is set high. When more than one filter matches, these bits give only the lowest filter number.
25	Layer 4 Filter Match When set, this bit indicates that the received frame matches one of the enabled Layer 4 Port Number fields. This status is given only when one of the following conditions is true: ■Layer 3 fields are not enabled and all enabled Layer 4 fields match. ■All enabled Layer 3 and Layer 4 filter fields match. When more than one filter matches, this bit gives the layer 4 filter status of filter indicated by Bits [27:26].
24	Layer 3 Filter Match When set, this bit indicates that the received frame matches one of the enabled Layer 3 IP Address fields. This status is given only when one of the following conditions is true: ■All enabled Layer 3 fields match and all enabled Layer 4 fields are bypassed. ■All enabled filter fields match. When more than one filter matches, this bit gives the layer 3 filter status of filter indicated by Bits [27:26].

23:21	Reserved
15	Reserved
14	Timestamp Dropped When set, this bit indicates that the timestamp was captured for this frame but got dropped in the MTL Rx FIFO because of overflow. This bit is available only when you select the Advanced Timestamp feature. Otherwise, this bit is reserved.
13	PTP Version When set, this bit indicates that the received PTP message is having the IEEE 1588 version 2 format. When reset, it has the version 1 format. This bit is available only when you select the Advanced Timestamp feature. Otherwise, this bit is reserved.
12	PTP Frame Type When set, this bit indicates that the PTP message is sent directly over Ethernet. When this bit is not set and the message type is non-zero, it indicates that the PTP message is sent over UDP-IPv4 or UDP-IPv6. The information about IPv4 or IPv6 can be obtained from Bits 6 and 7. This bit is available only when you select the Advanced Timestamp feature.
11:8	Message Type These bits are encoded to give the type of the message received. 0000 = No PTP message received 0001 = SYNC (all clock types) 0010 = Follow_Up (all clock types) 0011 = Delay_Req (all clock types) 0100 = Delay_Resp (all clock types) 0101 = Pdelay_Req (in peer-to-peer transparent clock) 0110 = Pdelay_Resp (in peer-to-peer transparent clock) 0111 = Pdelay_Resp_Follow_Up (in peer-to-peer transparent clock) 1000 = Announce 1001 = Management 1010 = Signaling 1011-1110 = Reserved 1111 = PTP packet with Reserved message type These bits are available only when you select the Advance Timestamp feature.
7	IPv6 Packet Received When set, this bit indicates that the received packet is an IPv6 packet. This bit is updated only when Bit 10 (IPC) of Register 0 (MAC Configuration Register) is set. This bit is available when you select the Enable Receive Full TCP/IP Checksum (Type 2) feature.
6	IPv4 Packet Received When set, this bit indicates that the received packet is an IPv4 packet. This bit is updated only when Bit 10 (IPC) of Register 0 (MAC Configuration Register) is set. This bit is available when you select the Enable Receive Full TCP/IP Checksum (Type 2) feature.
5	IP Checksum Bypassed When set, this bit indicates that the checksum offload engine is bypassed. This bit is available when you select the Enable Receive Full TCP/IP Checksum (Type 2) feature.
4	IP Payload Error When set, this bit indicates that the 16-bit IP payload checksum (that is, the TCP, UDP, or ICMP checksum) that the core calculated does not match the corresponding checksum field in the received segment. It is also set when the TCP, UDP, or ICMP segment length does not match the payload length value in the IP Header field. This bit is valid when either Bit 7 or Bit 6 is set. This bit is available when you select the Enable Receive Full TCP/IP Checksum (Type 2) feature.

3	IP Header Error When set, this bit indicates that either the 16-bit IPv4 header checksum calculated by the core does not match the received checksum bytes, or the IP datagram version is not consistent with the Ethernet Type value. This bit is valid when either Bit 7 or Bit 6 is set. This bit is available when you select the Enable Receive Full TCP/IP Checksum (Type 2) feature.
2:0	IP Payload Type These bits indicate the type of payload encapsulated in the IP datagram processed by the Receive Checksum Offload Engine (COE). The COE also sets these bits to 2'b00 if it does not process the IP datagram's payload due to an IP header error or fragmented IP. 3'b000 = Unknown or did not process IP payload 3'b001 = UDP 3'b010 = TCP 3'b011 = ICMP 3'b1xx = Reserved This bit is valid when either Bit 7 or Bit 6 is set. This bit is available when you select the Enable Receive Full TCP/IP Checksum (Type 2) feature.

Table 6.34-40 Receive Descriptor Fields 4 (RDES4)

RDES6 and RDES7 contain the snapshot of the timestamp. The availability of the snapshot of the timestamp in RDES6 and RDES7 is indicated by Bit 7 in the RDES0 descriptor. The contents of RDES6 and RDES7 are identified in Table 6.34-41 and Table 6.34-42, respectively.

Bit	Description
31:0	RTSL: Receive Frame Timestamp Low This field is updated by DMA with the least significant 32 bits of the timestamp captured for the corresponding receive frame. This field is updated by DMA only for the last descriptor of the receive frame which is indicated by Last Descriptor status bit (RDES0[8]).

Table 6.34-41 Receive Descriptor Fields 6 (RDES6)

Bit	Description
31:0	RTSH: Receive Frame Timestamp High This field is updated by DMA with the most significant 32 bits of the timestamp captured for the corresponding receive frame. This field is updated by DMA only for the last descriptor of the receive frame which is indicated by Last Descriptor status bit (RDES0[8]).

Table 6.34-42 Receive Descriptor Fields 7 (RDES7)

6.34.7 Register Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
GMAC Base Address: **GMACx_BA = 0x4012_0000 + (0x10000 * x)** **x=0, 1**				
MAC_Configuration	GMACx_BA+0x0000	R/W	Register 0 (MAC Configuration Register)	0x00000000
MAC_Frame_Filter	GMACx_BA+0x0004	R/W	Register 1 (MAC Frame Filter)	0x00000000
GMII_Address	GMACx_BA+0x0010	R/W	Register 4 (GMII Address Register)	0x00000000
GMII_Data	GMACx_BA+0x0014	R/W	Register 5 (GMII Data Register)	0x00000000
Flow_Control	GMACx_BA+0x0018	R/W	Register 6 (Flow Control Register)	0x00000000
VLAN_Tag	GMACx_BA+0x001c	R/W	Register 7 (VLAN Tag Register)	0x00000000
Version	GMACx_BA+0x0020	R	Register 8 (Version Register)	0x00001037
Debug	GMACx_BA+0x0024	R	Register 9 (Debug Register)	0x00000000
PMT_Control_Status	GMACx_BA+0x002c	R/W	Register 11 (PMT Control and Status Register)	0x00000000
LPI_Control_Status	GMACx_BA+0x0030	R/W	Register 12 (LPI Control and Status Register)	0x00000000
LPI_Timers_Control	GMACx_BA+0x0034	R/W	Register 13 (LPI Timers Control Register)	0x03E80000
Interrupt_Status	GMACx_BA+0x0038	R	Register 14 (Interrupt Register)	0x00000000
Interrupt_Mask	GMACx_BA+0x003c	R/W	Register 15 (Interrupt Mask Register)	0x00000000
MAC_Address0_High	GMACx_BA+0x0040	R/W	Register 16 (MAC Address0 High Register)	0x8000FFFF
MAC_Address0_Low	GMACx_BA+0x0044	R/W	Register 17 (MAC Address0 Low Register)	0xFFFFFFFF
MAC_Address1_High	GMACx_BA+0x0048	R/W	Register 18 (MAC Address1 High Register)	0x0000FFFF
MAC_Address1_Low	GMACx_BA+0x004c	R/W	Register 19 (MAC Address1 Low Register)	0xFFFFFFFF
MAC_Address2_High	GMACx_BA+0x0050	R/W	Register 20 (MAC Address2 High Register)	0x0000FFFF
MAC_Address2_Low	GMACx_BA+0x0054	R/W	Register 21 (MAC Address2 Low Register)	0xFFFFFFFF
MAC_Address3_High	GMACx_BA+0x0058	R/W	Register 22 (MAC Address3 High Register)	0x0000FFFF
MAC_Address3_Low	GMACx_BA+0x005c	R/W	Register 23 (MAC Address3 Low Register)	0xFFFFFFFF
MAC_Address4_High	GMACx_BA+0x0060	R/W	Register 24 (MAC Address4 High Register)	0x0000FFFF

MAC_Address4_Low	GMACx_BA+0x0064	R/W	Register 25 (MAC Address4 Low Register)	0xFFFFFFFF
MAC_Address5_High	GMACx_BA+0x0068	R/W	Register 26 (MAC Address5 High Register)	0x0000FFFF
MAC_Address5_Low	GMACx_BA+0x006c	R/W	Register 27 (MAC Address5 Low Register)	0xFFFFFFFF
MAC_Address6_High	GMACx_BA+0x0070	R/W	Register 28 (MAC Address6 High Register)	0x0000FFFF
MAC_Address6_Low	GMACx_BA+0x0074	R/W	Register 29 (MAC Address6 Low Register)	0xFFFFFFFF
MAC_Address7_High	GMACx_BA+0x0078	R/W	Register 30 (MAC Address7 High Register)	0x0000FFFF
MAC_Address7_Low	GMACx_BA+0x007c	R/W	Register 31 (MAC Address7 Low Register)	0xFFFFFFFF
MAC_Address8_High	GMACx_BA+0x0080	R/W	Register 32 (MAC Address8 High Register)	0x0000FFFF
MAC_Address8_Low	GMACx_BA+0x0084	R/W	Register 33 (MAC Address8 Low Register)	0xFFFFFFFF
RGMII_Control_Status	GMACx_BA+0x00d8	R	Register 54 (RGMII Status Register)	0x00000000
WDog_Timeout	GMACx_BA+0x00dc	R/W	Register 55 (Watchdog Timeout Register) This register controls the watchdog timeout for received frames	0x00000000
VLAN_Incl_Reg	GMACx_BA+0x0584	R/W	Register 353 (VLAN Tag Inclusion or Replacement Register)	0x00000000
Timestamp_Control	GMACx_BA+0x0700	R/W	Register 448 (Timestamp Control Register)	0x00002000
Sub_Second_Increment	GMACx_BA+0x0704	R/W	Register 449 (Sub-Second Increment Register)	0x00000000
System_Time_Seconds	GMACx_BA+0x0708	R	Register 450 (System Time - Seconds Register)	0x00000000
System_Time_Nanoseconds	GMACx_BA+0x070c	R	Register 451 (System Time - Nanoseconds Register)	0x00000000
System_Time_Seconds_Update	GMACx_BA+0x0710	R/W	Register 452 (System Time - Seconds Update Register)	0x00000000
System_Time_Nanoseconds_Update	GMACx_BA+0x0714	R/W	Register 453 (System Time - Nanoseconds Update Register)	0x00000000
Timestamp_Addend	GMACx_BA+0x0718	R/W	Register 454 (Timestamp Addend Register)	0x00000000
Target_Time_Seconds	GMACx_BA+0x071c	R/W	Register 455 (Target Time Seconds Register)	0x00000000
Target_Time_Nanoseconds	GMACx_BA+0x0720	R/W	Register 456 (Target Time Nanoseconds Register)	0x00000000
System_Time_Higher_Word_Seconds	GMACx_BA+0x0724	R/W	Register 457 (System Time - Higher Word Seconds Register)	0x00000000
Timestamp_Status	GMACx_BA+0x0728	R	Register 458 (Timestamp Status Register)	0x00000000

PPS_Control	GMACx_BA+0x072c	R/W	Register 459 (PPS Control Register)	0x00000000
PPS0_Interval	GMACx_BA+0x0760	R/W	Register 472 (PPS0 Interval Register)	0x00000000
PPS0_Width	GMACx_BA+0x0764	R/W	Register 473 (PPS0 Width Register)	0x00000000
Bus_Mode	GMACx_BA+0x1000	R/W	Register 1000 (Bus Mode Register)	0x00020101
Transmit_Poll_Demand	GMACx_BA+0x1004	R/W	Register 1001 (Transmit Poll Demand Register)	0x00000000
Receive_Poll_Demand	GMACx_BA+0x1008	R/W	Register 1002 (Receive Poll Demand Register)	0x00000000
Receive_Descriptor_List_Address	GMACx_BA+0x100c	R/W	Register 1003 (Receive Descriptor List Address Register)	0x00000000
Transmit_Descriptor_List_Address	GMACx_BA+0x1010	R/W	Register 1004 (Transmit Descriptor List Address Register)	0x00000000
Status	GMACx_BA+0x1014	R/W	Register 1005 (Status Register)	0x00000000
Operation_Mode	GMACx_BA+0x1018	R/W	Register 1006 (Operation Mode Register)	0x00000000
Interrupt_Enable	GMACx_BA+0x101c	R/W	Register 1007 (Interrupt Enable Register)	0x00000000
Missed_Frame_And_Buffer_Overflow_Counter	GMACx_BA+0x1020	R	Register 1008 (Missed Frame and Buffer Overflow Counter Register)	0x00000000
Receive_Interrupt_Watchdog_Timer	GMACx_BA+0x1024	R/W	Register 1009 (Receive Interrupt Watchdog Timer Register)	0x00000000
AXI_Bus_Mode	GMACx_BA+0x1028	R/W	Register 1010 (AXI Bus Mode Register)	0x00110001
AHB_or_AXI_Status	GMACx_BA+0x102c	R	Register 1011 (AHB or AXI Status Register)	0x00000000
Current_Host_Transmit_Descriptor	GMACx_BA+0x1048	R	Register 1018 (Current Host Transmit Descriptor Register)	0x00000000
Current_Host_Receive_Descriptor	GMACx_BA+0x104c	R	Register 1019 (Current Host Receive Descriptor Register)	0x00000000
Current_Host_Transmit_Buffer_Address	GMACx_BA+0x1050	R	Register 1020 (Current Host Transmit Buffer Address Register)	0x00000000
Current_Host_Receive_Buffer_Address	GMACx_BA+0x1054	R	Register 1021 (Current Host Receive Buffer Address Register)	0x00000000
HW_Feature	GMACx_BA+0x1058	R	Register 1022 (HW Feature Register)	0x1F0D6527

6.34.8 Register Description

GMAC Register 0 MAC Configuration Register (MAC_Configuration)

Register	Offset	R/W	Description	Reset Value
MAC_Configuration	GMACx_BA+0x0000	R/W	Register 0 (MAC Configuration Register)	0x00000000

31	30	29	28	27	26	25	24
Reserved	SARC			TWOKPE	Reserved	CST	TC
23	22	21	20	19	18	17	16
WD	JD	BE	JE	IFG			DCRS
15	14	13	12	11	10	9	8
PS	FES	DO	LM	DM	IPC	DR	LUD
7	6	5	4	3	2	1	0
ACS	BL		DC	TE	RE	PRELEN	

Bits	Description	
[31]	**Reserved**	Reserved.
[30:28]	**SARC**	**Source Address Insertion or Replacement Control** This field controls the source address insertion or replacement for all transmitted frames. Bit 30 specifies which MAC Address register (0 or 1) is used for source address insertion or replacement based on the values of Bits [29:28]: * 2'b0x = The input signals mti_sa_ctrl_i and ati_sa_ctrl_i control the SA field generation. * 2'b10 = ● If Bit 30 is set to 0, the MAC inserts the content of the MAC Address 0 registers (registers 16 and 17) in the SA field of all transmitted frames. ● If Bit 30 is set to 1, the MAC inserts the content of the MAC Address 1 registers (registers 18 and 19) in the SA field of all transmitted frames. * 2'b11 = ● If Bit 30 is set to 0, the MAC replaces the content of the MAC Address 0 registers (registers 16 and 17) in the SA field of all transmitted frames. ● If Bit 30 is set to 1, the MAC replaces the content of the MAC Address 1 registers (registers 18 and 19) in the SA field of all transmitted frames. **Note:** ● Changes to this field take effect only on the start of a frame. If you write this register field when a frame is being transmitted, only the subsequent frame can use the updated value, that is, the current frame does not use the updated value.
[27]	**TWOKPE**	**IEEE 802.3as Support for 2K Packets** When set, the MAC considers all frames, with up to 2,000 bytes length, as normal packets. When Bit 20 (JE) is not set, the MAC considers all received frames of size more than 2 Kbytes as Giant frames. When this bit is reset and Bit 20 (JE) is not set, the MAC considers all received frames of size more than 1,518 bytes (1,522 bytes for tagged) as Giant frames. When Bit 20 is set, setting this bit has no effect on Giant Frame status.
[26]	**Reserved**	Reserved.

[25]	**CST**	**CRC Stripping for Type Frames** When this bit is set, the last 4 bytes (FCS) of all frames of Ether type (Length/Type field greater than or equal to 1,536) are stripped and dropped before forwarding the frame to the application. This function is not valid when the IP Checksum Engine (Type 1) is enabled in the MAC receiver. This function is valid when Type 2 Checksum Offload Engine is enabled.
[24]	**TC**	**Transmit Configuration in RGMII** When set, this bit enables the transmission of duplex mode, link speed, and link up or down information to the PHY in the RGMII port. When this bit is reset, no such information is driven to the PHY.
[23]	**WD**	**Watchdog Disable** When this bit is set, the MAC disables the watchdog timer on the receiver. The MAC can receive frames of up to 16,384 bytes. When this bit is reset, the MAC does not allow a receive frame which more than 2,048 bytes (10,240 if JE is set high) or the value programmed in Register 55 (Watchdog Timeout Register). The MAC cuts off any bytes received after the watchdog limit number of bytes.
[22]	**JD**	**Jabber Disable** When this bit is set, the MAC disables the jabber timer on the transmitter. The MAC can transfer frames of up to 16,384 bytes. When this bit is reset, the MAC cuts off the transmitter if the application sends out more than 2,048 bytes of data (10,240 if JE is set high) during transmission.
[21]	**BE**	**Frame Burst Enable** When this bit is set, the MAC allows frame bursting during transmission in the GMII half-duplex mode.
[20]	**JE**	**Jumbo Frame Enable** When this bit is set, the MAC allows Jumbo frames of 9,018 bytes (9,022 bytes for VLAN tagged frames) without reporting a giant frame error in the receive frame status.
[19:17]	**IFG**	**Inter-Frame Gap** These bits control the minimum IFG between frames during transmission. 000 =96 bit times 001 =88 bit times 010 = 80 bit times ... 111 = 40 bit times In the half-duplex mode, the minimum IFG can be configured only for 64 bit times (IFG = 100). Lower values are not considered. In the 1000-Mbps mode, the minimum IFG supported is 80 bit times (and above).
[16]	**DCRS**	**Disable Carrier Sense During Transmission** When set high, this bit makes the MAC transmitter ignore the (G)MII CRS signal during frame transmission in the half-duplex mode. This request results in no errors generated because of Loss of Carrier or No Carrier during such transmission. When this bit is low, the MAC transmitter generates such errors because of Carrier Sense and can even abort the transmissions.
[15]	**PS**	**Port Select** This bit selects the Ethernet line speed: 0 = For 1000 Mbps operations 1 = For 10 or 100 Mbps operations In 10 or 100 Mbps operations, this bit, along with FES bit, selects the exact line speed. The mac_portselect_o signal reflects the value of this bit.
[14]	**FES**	**Speed** This bit selects the speed in the RMII or RGMII interface 0 = 10 Mbps 1 = 100 Mbps
[13]	**DO**	**Disable Receive Own** When this bit is set, the MAC disables the reception of frames when the phy_txen_o is asserted in the half-duplex mode. When this bit is reset, the MAC receives all packets that are given by the PHY while transmitting. This bit is not applicable if the MAC is operating in the full-duplex mode.

[12]	**LM**	**Loopback Mode** When this bit is set, the MAC operates in the loopback mode at GMII or MII. The (G)MII Receive clock input (clk_rx_i) is required for the loopback to work properly, because the Transmit clock is not looped-back internally.
[11]	**DM**	**Duplex Mode** When this bit is set, the MAC operates in the full-duplex mode where it can transmit and receive simultaneously.
[10]	**IPC**	**Checksum Offload** When this bit is set, the MAC calculates the 16-bit one's complement of the one's complement sum of all received Ethernet frame payloads. It also checks whether the IPv4 Header checksum (assumed to be bytes 2526 or 2930 (VLAN-tagged) of the received Ethernet frame) is correct for the received frame and gives the status in the receive status word. The MAC also appends the 16-bit checksum calculated for the IP header datagram payload (bytes after the IPv4 header) and appends it to the Ethernet frame transferred to the application (when Type 2 COE is deselected). When this bit is reset, this function is disabled. When Type 2 COE is selected, this bit, when set, enables the IPv4 header checksum checking and IPv4 or IPv6 TCP, UDP, or ICMP payload checksum checking. When this bit is reset, the COE function in the receiver is disabled and the corresponding PCE and IP HCE status bits are always cleared.
[9]	**DR**	**Disable Retry** When this bit is set, the MAC attempts only one transmission. When a collision occurs on the GMII or MII interface, the MAC ignores the current frame transmission and reports a Frame Abort with excessive collision error in the transmit frame status. When this bit is reset, the MAC attempts retries based on the settings of the BL field (Bits [6:5]). This bit is applicable only in the half-duplex mode.
[8]	**LUD**	**Link Up or Down** This bit indicates whether the link is up or down during the transmission of configuration in the RGMII interface: 0 = Link Down 1 = Link Up
[7]	**ACS**	**Automatic Pad or CRC Stripping** When this bit is set, the MAC strips the Pad or FCS field on the incoming frames only if the value of the length field is less than 1,536 bytes. All received frames with length field greater than or equal to 1,536 bytes are passed to the application without stripping the Pad or FCS field. When this bit is reset, the MAC passes all incoming frames, without modifying them, to the Host.
[6:5]	**BL**	**Back-Off Limit** The Back-Off limit determines the random integer number (r) of slot time delays (4,096 bit times for 1000 Mbps and 512 bit times for 10/100 Mbps) for which the MAC waits before rescheduling a transmission attempt during retries after a collision. This bit is applicable only in the half-duplex mode. 00 = k is the min of (n, 10). 01 = k is the min of (n, 8). 10 = k is the min of (n, 4). 11 = k is the min of (n, 1). where *n* = retransmission attempt. The random integer *r* takes the value in the range 0 <= r < kth power of 2.

[4]	**DC**	**Deferral Check** When this bit is set, the deferral check function is enabled in the MAC. The MAC issues a Frame Abort status, along with the excessive deferral error bit set in the transmit frame status, when the transmit state machine is deferred for more than 24,288 bit times in the 10 or 100 Mbps mode. If the MAC is configured for 1000 Mbps operation or if the Jumbo frame mode is enabled in the 10 or 100 Mbps mode, the threshold for deferral is 155,680 bits times. Deferral begins when the transmitter is ready to transmit, but it is prevented because of an active carrier sense signal (CRS) on GMII or MII. The defer time is not cumulative. For example, if the transmitter defers for 10,000 bit times because the CRS signal is active and then the CRS signal becomes inactive, the transmitter transmits and collision happens. Because of collision, the transmitter needs to back off and then defer again after back off completion. In such a scenario, the deferral timer is reset to 0 and it is restarted. When this bit is reset, the deferral check function is disabled and the MAC defers until the CRS signal goes inactive. This bit is applicable only in the half-duplex mode.
[3]	**TE**	**Transmitter Enable** When this bit is set, the transmit state machine of the MAC is enabled for transmission on the GMII or MII. When this bit is reset, the MAC transmit state machine is disabled after the completion of the transmission of the current frame, and does not transmit any further frames.
[2]	**RE**	**Receiver Enable** When this bit is set, the receiver state machine of the MAC is enabled for receiving frames from the GMII or MII. When this bit is reset, the MAC receive state machine is disabled after the completion of the reception of the current frame, and does not receive any further frames from the GMII or MII.
[1:0]	**PRELEN**	**Preamble Length for Transmit Frames** These bits control the number of preamble bytes that are added to the beginning of every Transmit frame. The preamble reduction occurs only when the MAC is operating in the full-duplex mode. 2'b00 = 7 bytes of preamble 2'b01 = 5 byte of preamble 2'b10 = 3 bytes of preamble 2'b11 = reserved

GMAC Register 1 MAC Frame Filter)Register (MAC_Frame_Filter)

Register	Offset	R/W	Description	Reset Value
MAC_Frame_Filter	GMACx_BA+0x0004	R/W	Register 1 (MAC Frame Filter)	0x00000000

31	30	29	28	27	26	25	24
RA	Reserved						
23	22	21	20	19	18	17	16
Reserved							VTFE
15	14	13	12	11	10	9	8
Reserved						SAF	SAIF
7	6	5	4	3	2	1	0
PCF		DBF	PM	DAIF	Reserved		PR

Bits	Description	
[31]	**RA**	**Receive All** When this bit is set, the MAC Receiver module passes all received frames, irrespective of whether they pass the address filter or not, to the Application. The result of the SA or DA filtering is updated (pass or fail) in the corresponding bits in the Receive Status Word. When this bit is reset, the Receiver module passes only those frames to the Application that pass the SA or DA address filter.
[30:17]	**Reserved**	Reserved.
[16]	**VTFE**	**VLAN Tag Filter Enable** When set, this bit enables the MAC to drop VLAN tagged frames that do not match the VLAN Tag comparison. When reset, the MAC forwards all frames irrespective of the match status of the VLAN Tag.
[15:10]	**Reserved**	Reserved.
[9]	**SAF**	**Source Address Filter Enable** When this bit is set, the MAC compares the SA field of the received frames with the values programmed in the enabled SA registers. If the comparison fails, the MAC drops the frame. When this bit is reset, the MAC forwards the received frame to the application with updated SAF bit of the Rx Status depending on the SA address comparison. **Note:** According to the IEEE specification, Bit 47 of the SA is reserved and set to 0. However, in GMAC, the MAC compares all 48 bits. The software driver should take this into consideration while programming the MAC address registers for SA.
[8]	**SAIF**	**SA Inverse Filtering** When this bit is set, the Address Check block operates in inverse filtering mode for the SA address comparison. The frames whose SA matches the SA registers are marked as failing the SA Address filter. When this bit is reset, frames whose SA does not match the SA registers are marked as failing the SA Address filter.

[7:6]	**PCF**	**Pass Control Frames** These bits control the forwarding of all control frames (including unicast and multicast PAUSE frames). 00 = MAC filters all control frames from reaching the application. 01 = MAC forwards all control frames except PAUSE control frames to application even if they fail the Address filter. 10 = MAC forwards all control frames to application even if they fail the Address Filter. 11 = MAC forwards control frames that pass the Address Filter. The following conditions should be true for the PAUSE control frames processing: • Condition 1: The MAC is in the full-duplex mode and flow control is enabled by setting Bit 2 (RFE) of Register 6 (Flow Control Register) to 1. • Condition 2: The destination address (DA) of the received frame matches the special multicast address or the MAC Address 0 when Bit 3 (UP) of the Register 6 (Flow Control Register) is set. • Condition 3: The Type field of the received frame is 0x8808 and the OPCODE field is 0x0001. **Note:** This field should be set to 01 only when the Condition 1 is true, that is, the MAC is programmed to operate in the full-duplex mode and the RFE bit is enabled. Otherwise, the PAUSE frame filtering may be inconsistent. When Condition 1 is false, the PAUSE frames are considered as generic control frames. Therefore, to pass all control frames (including PAUSE control frames) when the full-duplex mode and flow control is not enabled, you should set the PCF field to 10 or 11 (as required by the application).
[5]	**DBF**	**Disable Broadcast Frames** When this bit is set, the AFM module filters all incoming broadcast frames. In addition, it overrides all other filter settings. When this bit is reset, the AFM module passes all received broadcast frames.
[4]	**PM**	**Pass All Multicast** When set, this bit indicates that all received frames with a multicast destination address (first bit in the destination address field is '1') are passed.
[3]	**DAIF**	**DA Inverse Filtering** When this bit is set, the Address Check block operates in inverse filtering mode for the DA address comparison for both unicast and multicast frames. When reset, normal filtering of frames is performed.
[2:1]	**Reserved**	Reserved.
[0]	**PR**	**Promiscuous Mode** When this bit is set, the Address Filter module passes all incoming frames regardless of its destination or source address. The SA or DA Filter Fails status bits of the Receive Status Word are always cleared when PR is set.

GMAC Register 4 GMII Address Register (GMII_Address)

Register	Offset	R/W	Description	Reset Value
GMII_Address	GMACx_BA+0x0010	R/W	Register 4 (GMII Address Register)	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PA					GR		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GR		CR				GW	GB

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:11]	**PA**	**Physical Layer Address** This field indicates which of the 32 possible PHY devices are being accessed.
[10:6]	**GR**	**GMII Register** These bits select the desired GMII register in the selected PHY device.
[5:2]	**CR**	**CSR Clock Range** The CSR Clock Range selection determines the frequency of the MDC clock according to the CSR clock frequency used in your design. The suggested range of CSR clock frequency applicable for each value (when Bit[5] = 0) ensures that the MDC clock is approximately between the frequency range 1.0 MHz - 2.5 MHz. 0000 = The frequency of the CSR clock is 60-100 MHz and the MDC clock is CSR clock/42. 0001 = The frequency of the CSR clock is 100-150 MHz and the MDC clock is CSR clock/62. 0010 = The frequency of the CSR clock is 20-35 MHz and the MDC clock is CSR clock/16. 0011 = The frequency of the CSR clock is 35-60 MHz and the MDC clock is CSR clock/26. 0100 = The frequency of the CSR clock is 150-250 MHz and the MDC clock is CSR clock/102. 0100 = The frequency of the CSR clock is 250-300 MHz and the MDC clock is CSR clock/124. 0110 and 0111 = Reserved When Bit 5 is set, you can achieve MDC clock of frequency higher than the IEEE 802.3 specified frequency limit of 2.5 MHz and program a clock divider of lower value. For example, when CSR clock is of 100 MHz frequency and you program these bits as 1010, then the resultant MDC clock is of 12.5 MHz which is outside the limit of IEEE 802.3 specified range. Program the following values only if the interfacing chips support faster MDC clocks: 1000 = CSR clock/4 1001 = CSR clock/6 1010 = CSR clock/8 1011 = CSR clock/10 1100 = CSR clock/12 1101 = CSR clock/14 1110 = CSR clock/16 1111 = CSR clock/18

[1]	**GW**	**GMII Write** When set, this bit indicates to the PHY that this is a Write operation using the GMII Data register. If this bit is not set, it indicates that this is a Read operation, that is, placing the data in the GMII Data register.
[0]	**GB**	**GMII Busy** This bit should read logic 0 before writing to Register 4 and Register 5. During a PHY register access, the software sets this bit to 1'b1 to indicate that a Read or Write access is in progress. Register 5 is invalid until this bit is cleared by the MAC. Therefore, Register 5 (GMII Data) should be kept valid until the MAC clears this bit during a PHY Write operation. Similarly for a read operation, the contents of Register 5 are not valid until this bit is cleared. The subsequent read or write operation should happen only after the previous operation is complete. Because there is no acknowledgment from the PHY to MAC after a read or write operation is completed, there is no change in the functionality of this bit even when the PHY is not present.

GMAC Register 5 GMII Data Register (GMII_Data)

Register	Offset	R/W	Description	Reset Value
GMII_Data	GMACx_BA+0x0014	R/W	Register 5 (GMII Data Register)	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
GD							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GD							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**GD**	**GMII Data** This field contains the 16-bit data value read from the PHY after a Management Read operation or the 16-bit data value to be written to the PHY before a Management Write operation.

GMAC Register 6 Flow Control Register (Flow_Control)

Register	Offset	R/W	Description	Reset Value
Flow_Control	GMACx_BA+0x0018	R/W	Register 6 (Flow Control Register)	0x00000000

31	30	29	28	27	26	25	24
PT							
23	22	21	20	19	18	17	16
PT							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
DZPQ	Reserved	PLT		UP	RFE	TFE	FCA_BPA

Bits	Description	
[31:16]	**PT**	**Pause Time** This field holds the value to be used in the Pause Time field in the transmit control frame. Consecutive writes to this register should be performed only after at least four clock cycles in the destination clock domain.
[15:8]	**Reserved**	Reserved.
[7]	**DZPQ**	**Disable Zero-Quanta Pause** When this bit is set, it disables the automatic generation of the Zero-Quanta Pause Control frames on the de-assertion of the flow-control signal from the FIFO layer (MTL or external sideband flow control signal sbd_flowctrl_i/mti_flowctrl_i). When this bit is reset, normal operation with automatic Zero-Quanta Pause Control frame generation is enabled.
[6]	**Reserved**	Reserved.
[5:4]	**PLT**	**Pause Low Threshold** This field configures the threshold of the PAUSE timer at which the input flow control signal mti_flowctrl_i (or sbd_flowctrl_i) is checked for automatic retransmission of PAUSE Frame. The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100H (256 slot-times), and PLT = 01, then a second PAUSE frame is automatically transmitted if the mti_flowctrl_i signal is asserted at 228 (256 - 28) slot times after the first PAUSE frame is transmitted. The following list provides the threshold values for different values: 00 = The threshold is Pause time minus 4 slot times (PT - 4 slot times). 01 = The threshold is Pause time minus 28 slot times (PT - 28 slot times). 10 = The threshold is Pause time minus 144 slot times (PT - 144 slot times). 11 = The threshold is Pause time minus 256 slot times (PT - 256 slot times). The slot time is defined as the time taken to transmit 512 bits (64 bytes) on the GMII or MII interface.
[3]	**UP**	**Unicast Pause Frame Detect** A pause frame is processed when it has the unique multicast address specified in the IEEE Std 802.3. When this bit is set, the MAC can also detect Pause frames with unicast address of the station. This unicast address should be as specified in the MAC Address0 High Register and MAC Address0 Low Register. When this bit is reset, the MAC only detects Pause frames with unique multicast address. **Note:** The MAC does not process a Pause frame if the multicast address of received frame is different from the unique multicast address.

[2]	**RFE**	**Receive Flow Control Enable** When this bit is set, the MAC decodes the received Pause frame and disables its transmitter for a specified (Pause) time. When this bit is reset, the decode function of the Pause frame is disabled.
[1]	**TFE**	**Transmit Flow Control Enable** In the full-duplex mode, when this bit is set, the MAC enables the flow control operation to transmit Pause frames. When this bit is reset, the flow control operation in the MAC is disabled, and the MAC does not transmit any Pause frames. In half-duplex mode, when this bit is set, the MAC enables the back-pressure operation. When this bit is reset, the back-pressure feature is disabled.
[0]	**FCA_BPA**	**Flow Control Busy or Backpressure Activate** This bit initiates a Pause Control frame in the full-duplex mode and activates the backpressure function in the half-duplex mode if the TFE bit is set. In the full-duplex mode, this bit should be read as 1'b0 before writing to the Flow Control register. To initiate a Pause control frame, the Application must set this bit to 1'b1. During a transfer of the Control Frame, this bit continues to be set to signify that a frame transmission is in progress. After the completion of Pause control frame transmission, the MAC resets this bit to 1'b0. The Flow Control register should not be written to until this bit is cleared. In the half-duplex mode, when this bit is set (and TFE is set), then backpressure is asserted by the MAC. During backpressure, when the MAC receives a new frame, the transmitter starts sending a JAM pattern resulting in a collision. This control register bit is logically ORed with the mti_flowctrl_i input signal for the backpressure function. When the MAC is configured for the full-duplex mode, the BPA is automatically disabled.

GMAC Register 7 VLAN Tag Register (VLAN_Tag)

Register	Offset	R/W	Description	Reset Value
VLAN_Tag	GMACx_BA+0x001c	R/W	Register 7 (VLAN Tag Register)	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved					ESVL	VTIM	ETV
15	14	13	12	11	10	9	8
VL							
7	6	5	4	3	2	1	0
VL							

Bits	Description	
[31:19]	**Reserved**	Reserved.
[18]	**ESVL**	**Enable S-VLAN** When this bit is set, the MAC transmitter and receiver also consider the S-VLAN (Type = 0x88A8) frames as valid VLAN tagged frames.
[17]	**VTIM**	**VLAN Tag Inverse Match Enable** When set, this bit enables the VLAN Tag inverse matching. The frames that do not have matching VLAN Tag are marked as matched. When reset, this bit enables the VLAN Tag perfect matching. The frames with matched VLAN Tag are marked as matched.
[16]	**ETV**	**Enable 12-Bit VLAN Tag Comparison** When this bit is set, a 12-bit VLAN identifier is used for comparing and filtering instead of the complete 16-bit VLAN tag. Bits [11:0] of VLAN tag are compared with the corresponding field in the received VLAN-tagged frame. Similarly, when enabled, only 12 bits of the VLAN tag in the received frame are used for hash-based VLAN filtering. When this bit is reset, all 16 bits of the 15th and 16th bytes of the received VLAN frame are used for comparison and VLAN hash filtering.
[15:0]	**VL**	**VLAN Tag Identifier for Receive Frames** This field contains the 802.1Q VLAN tag to identify the VLAN frames and is compared to the 15th and 16th bytes of the frames being received for VLAN frames. The following list describes the bits of this field: Bits [15:13] = User Priority Bit 12:Canonical Format Indicator (CFI) or Drop Eligible Indicator (DEI) Bits[11:0] = VLAN tag's VLAN Identifier (VID) field When the ETV bit is set, only the VID (Bits[11:0]) is used for comparison. If VL (VL[11:0] if ETV is set) is all zeros, the MAC does not check the fifteenth and 16th bytes for VLAN tag comparison, and declares all frames with a Type field value of 0x8100 or 0x88a8 as VLAN frames.

GMAC Register 9 Debug Register (Debug)

Register	Offset	R/W	Description	Reset Value
Debug	GMACx_BA+0x0024	R	Register 9 (Debug Register)	0x00000000

31	30	29	28	27	26	25	24
Reserved						TXSTSFSTS	TXFSTS
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved	TWCSTS	TRCSTS		TXPAUSED	TFCSTS		TPESTS
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						RXFSTS	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	RRCSTS		RWCSTS	Reserved	RFCFCSTS		RPESTS

Bits	Description	
[31:26]	**Reserved**	Reserved.
[25]	**TXSTSFSTS**	**MTL TxStatus FIFO Full Status** When high, this bit indicates that the MTL TxStatus FIFO is full. Therefore, the MTL cannot accept any more frames for transmission.
[24]	**TXFSTS**	**MTL Tx FIFO Not Empty Status** When high, this bit indicates that the MTL Tx FIFO is not empty and some data is left for transmission.
[23]	**Reserved**	Reserved.
[22]	**TWCSTS**	**MTL Tx FIFO Write Controller Active Status** When high, this bit indicates that the MTL Tx FIFO Write Controller is active and transferring data to the Tx FIFO.
[21:20]	**TRCSTS**	**MTL Tx FIFO Read Controller Status** This field indicates the state of the Tx FIFO Read Controller: 00 = IDLE state 01 = READ state (transferring data to MAC transmitter) 10 = Waiting for TxStatus from MAC transmitter 11 = Writing the received TxStatus or flushing the Tx FIFO
[19]	**TXPAUSED**	**MAC transmitter in PAUSE** When high, this bit indicates that the MAC transmitter is in the PAUSE condition (in the full-duplex only mode) and hence does not schedule any frame for transmission.
[18:17]	**TFCSTS**	**MAC Transmit Frame Controller Status** This field indicates the state of the MAC Transmit Frame Controller module: 00 = IDLE state 01 = Waiting for Status of previous frame or IFG or backoff period to be over 10 = Generating and transmitting a PAUSE control frame (in the full-duplex mode) 11 = Transferring input frame for transmission

[16]	**TPESTS**	**MAC GMII or MII Transmit Protocol Engine Status** When high, this bit indicates that the MAC GMII or MII transmit protocol engine is actively transmitting data and is not in the IDLE state.
[15:10]	**Reserved**	Reserved.
[9:8]	**RXFSTS**	**MTL Rx FIFO Fill-level Status** This field gives the status of the fill-level of the Rx FIFO: 00 = Rx FIFO Empty 01 = Rx FIFO fill level is below the flow-control deactivate threshold 10 = Rx FIFO fill level is above the flow-control activate threshold 11 = Rx FIFO Full
[7]	**Reserved**	Reserved.
[6:5]	**RRCSTS**	**MTL Rx FIFO Read Controller State** This field gives the state of the Rx FIFO read Controller: 00 = IDLE state 01 = Reading frame data 10 = Reading frame status (or timestamp) 11 = Flushing the frame data and status
[4]	**RWCSTS**	**MTL Rx FIFO Write Controller Active Status** When high, this bit indicates that the MTL Rx FIFO Write Controller is active and is transferring a received frame to the FIFO.
[3]	**Reserved**	Reserved.
[2:1]	**RFCFCSTS**	**MAC Receive Frame Controller FIFO Status** When high, this field indicates the active state of the small FIFO Read and Write controllers of the MAC Receive Frame Controller Module.
[0]	**RPESTS**	**MAC GMII or MII Receive Protocol Engine Status** When high, this bit indicates that the MAC GMII or MII receive protocol engine is actively receiving data and not in IDLE state.

GMAC Register 11 PMT Control and Status Register (PMT_Control_Status)

Register	Offset	R/W	Description	Reset Value
PMT_Control_Status	GMACx_BA+0x002c	R/W	Register 11 (PMT Control and Status Register)	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		MGKPRCVD	Reserved			MGKPKTEN	PWRDWN

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**MGKPRCVD**	**Magic Packet Received (read only)** When set, this bit indicates that the power management event is generated because of the reception of a magic packet. This bit is cleared by a Read into this register.
[4:2]	**Reserved**	Reserved.
[1]	**MGKPKTEN**	**Magic Packet Enable** When set, enables generation of a power management event because of magic packet reception.
[0]	**PWRDWN**	**Power Down** When set, the MAC receiver drops all received frames until it receives the expected magic packet or wake-up frame. This bit is then self-cleared and the power-down mode is disabled. The Software can also clear this bit before the expected magic packet or wake-up frame is received. The frames, received by the MAC after this bit is cleared, are forwarded to the application. This bit must only be set when the Magic Packet Enable, Global Unicast, or Wake-Up Frame Enable bit is set high. **Note:** You can gate-off the CSR clock during the power-down mode. However, when the CSR clock is gated-off, you cannot perform any read or write operations on this register. Therefore, the Software cannot clear this bit.

GMAC Register 12 LPI Control and Status Register (LPI_Control_Status)

Register	Offset	R/W	Description	Reset Value
LPI_Control_Status	GMACx_BA+0x0030	R/W	Register 12 (LPI Control and Status Register)	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				LPITXA	PLSEN	PLS	LPIEN
15	14	13	12	11	10	9	8
Reserved						RLPIST	TLPIST
7	6	5	4	3	2	1	0
Reserved				RLPIEX	RLPIEN	TLPIEX	TLPIEN

Bits	Description	
[31:20]	**Reserved**	Reserved.
[19]	**LPITXA**	**LPI TX Automate** This bit controls the behavior of the MAC when it is entering or coming out of the LPI mode on the transmit side. If the LPITXA and LPIEN bits are set to 1, the MAC enters the LPI mode only after all outstanding frames (in the core) and pending frames (in the application interface) have been transmitted. The MAC comes out of the LPI mode when the application sends any frame for transmission or the application issues a TX FIFO Flush command. In addition, the MAC automatically clears the LPIEN bit when it exits the LPI state. If TX FIFO Flush is set, in Bit 20 of Register 1006 (Operation Mode Register), when the MAC is in the LPI mode, the MAC exits the LPI mode. When this bit is 0, the LPIEN bit directly controls behavior of the MAC when it is entering or coming out of the LPI mode.
[18]	**PLSEN**	**PHY Link Status Enable** This bit enables the link status received on the RGMII receive paths to be used for activating the LPI LS TIMER. When set, the MAC uses the link-status bits of Register 54 (RGMII Status Register) and Bit 17 (PLS) for the LPI LS Timer trigger. When cleared, the MAC ignores the link-status bits of Register 54 and takes only the PLS bit. This bit is RO and reserved if you have not selected the RGMII PHY interface.
[17]	**PLS**	**PHY Link Status** This bit indicates the link status of the PHY. The MAC Transmitter asserts the LPI pattern only when the link status is up (okay) at least for the time indicated by the LPI LS TIMER. When set, the link is considered to be okay (up) and when reset, the link is considered to be down.
[16]	**LPIEN**	**LPI Enable** When set, this bit instructs the MAC Transmitter to enter the LPI state. When reset, this bit instructs the MAC to exit the LPI state and resume normal transmission. This bit is cleared when the LPITXA bit is set and the MAC exits the LPI state because of the arrival of a new packet for transmission.
[15:10]	**Reserved**	Reserved.
[9]	**RLPIST**	**Receive LPI State (read only)** When set, this bit indicates that the MAC is receiving the LPI pattern on the GMII or MII interface.

[8]	**TLPIST**	**Transmit LPI State (read only)** When set, this bit indicates that the MAC is transmitting the LPI pattern on the GMII or MII interface.
[7:4]	**Reserved**	Reserved.
[3]	**RLPIEX**	**Receive LPI Exit (read only)** When set, this bit indicates that the MAC Receiver has stopped receiving the LPI pattern on the GMII or MII interface, exited the LPI state, and resumed the normal reception. This bit is cleared by a read into this register. **Note:** This bit may not get set if the MAC stops receiving the LPI pattern for a very short duration, such as, less than 3 clock cycles of CSR clock.
[2]	**RLPIEN**	**Receive LPI Entry (read only)** When set, this bit indicates that the MAC Receiver has received an LPI pattern and entered the LPI state. This bit is cleared by a read into this register. **Note:** This bit may not get set if the MAC stops receiving the LPI pattern for a very short duration, such as, less than 3 clock cycles of CSR clock.
[1]	**TLPIEX**	**Transmit LPI Exit (read only)** When set, this bit indicates that the MAC transmitter has exited the LPI state after the user has cleared the LPIEN bit and the LPI TW Timer has expired. This bit is cleared by a read into this register.
[0]	**TLPIEN**	**Transmit LPI Entry (read only)** When set, this bit indicates that the MAC Transmitter has entered the LPI state because of the setting of the LPIEN bit. This bit is cleared by a read into this register.

GMAC Register 13 LPI Timers Control Register (LPI_Timers_Control)

Register	Offset	R/W	Description	Reset Value
LPI_Timers_Control	GMACx_BA+0x0034	R/W	Register 13 (LPI Timers Control Register)	0x03E80000

31	30	29	28	27	26	25	24
Reserved						LST	
23	22	21	20	19	18	17	16
LST							
15	14	13	12	11	10	9	8
TWT							
7	6	5	4	3	2	1	0
TWT							

Bits	Description	
[31:26]	**Reserved**	Reserved.
[25:16]	**LST**	**LPI LS Timer** This field specifies the minimum time (in milliseconds) for which the link status from the PHY should be up (OKAY) before the LPI pattern can be transmitted to the PHY. The MAC does not transmit the LPI pattern even when the LPIEN bit is set unless the LPI LS Timer reaches the programmed terminal count. The default value of the LPI LS Timer is 1000 (1 sec) as defined in the IEEE standard.
[15:0]	**TWT**	**LPI TW Timer** This field specifies the minimum time (in microseconds) for which the MAC waits after it stops transmitting the LPI pattern to the PHY and before it resumes the normal transmission. The TLPIEX status bit is set after the expiry of this timer.

GMAC Register 14 Interrupt Register (Interrupt_Status)

Register	Offset	R/W	Description	Reset Value
Interrupt_Status	GMACx_BA+0x0038	R	Register 14 (Interrupt Register)	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved					LPIIS	TSIS	Reserved
7	6	5	4	3	2	1	0
Reserved				PMTIS	Reserved		RGSMIIIS

Bits	Description	
[31:11]	**Reserved**	Reserved.
[10]	**LPIIS**	**LPI Interrupt Status** When the Energy Efficient Ethernet feature is enabled, this bit is set for any LPI state entry or exit in the MAC Transmitter or Receiver. This bit is cleared on reading Bit 0 of Register 12 (LPI Control and Status Register). In all other modes, this bit is reserved.
[9]	**TSIS**	**Timestamp Interrupt Status** When the Advanced Timestamp feature is enabled, this bit is set when any of the following conditions is true: ● The system time value equals or exceeds the value specified in the Target Time High and Low registers. ● There is an overflow in the seconds register. ● The Auxiliary snapshot trigger is asserted. This bit is cleared on reading Bit 0 of the Register 458 (Timestamp Status Register). If default Timestamping is enabled, when set, this bit indicates that the system time value is equal to or exceeds the value specified in the Target Time registers. In this mode, this bit is cleared after the completion of the read of this bit. In all other modes, this bit is reserved.
[8:4]	**Reserved**	Reserved.
[3]	**PMTIS**	**PMT Interrupt Status** This bit is set when a Magic packet or Wake-on-LAN frame is received in the power-down mode (see Bits 5 in the PMT Control and Status Register). This bit is cleared when Bit 5 is cleared because of a read operation to the PMT Control and Status register.
[2:1]	**Reserved**	Reserved.
[0]	**RGSMIIIS**	**RGMII Interrupt Status** This bit is set because of any change in value of the Link Status of RGMII interface (Bit 3 in Register 54 (RGMII Status Register)). This bit is cleared when you perform a read operation on the RGMII Status Register.

GMAC Register 15 Interrupt Mask Register (Interrupt_Mask)

Register	Offset	R/W	Description	Reset Value
Interrupt_Mask	GMACx_BA+0x003c	R/W	Register 15 (Interrupt Mask Register)	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved					LPIIM	TSIM	Reserved
7	6	5	4	3	2	1	0
Reserved				PMTIM	PCSANCIM	PCSLCHGIM	RGSMIIIM

Bits	Description	
[31:11]	**Reserved**	Reserved.
[10]	**LPIIM**	**LPI Interrupt Mask** When set, this bit disables the assertion of the interrupt signal because of the setting of the LPI Interrupt Status bit in Register 14 (Interrupt Status Register).
[9]	**TSIM**	**Timestamp Interrupt Mask** When set, this bit disables the assertion of the interrupt signal because of the setting of Timestamp Interrupt Status bit in Register 14 (Interrupt Status Register). This bit is valid only when IEEE1588 timestamping is enabled. In all other modes, this bit is reserved.
[8:4]	**Reserved**	Reserved.
[3]	**PMTIM**	**PMT Interrupt Mask** When set, this bit disables the assertion of the interrupt signal because of the setting of PMT Interrupt Status bit in Register 14 (Interrupt Status Register).
[2]	**PCSANCIM**	**PCS AN Completion Interrupt Mask (read only)** When set, this bit disables the assertion of the interrupt signal because of the setting of PCS Auto-negotiation complete bit in Register 14 (Interrupt Status Register)
[1]	**PCSLCHGIM**	**PCS Link Status Interrupt Mask (read only)** When set, this bit disables the assertion of the interrupt signal because of the setting of the PCS Link-status changed bit in Register 14 (Interrupt Status Register).
[0]	**RGSMIIIM**	**RGMII Interrupt Mask** When set, this bit disables the assertion of the interrupt signal because of the setting of the RGMII Interrupt Status bit in Register 14 (Interrupt Status Register).

GMAC Register 16 MAC Address0 High Register (MAC_Address0_High)

Register	Offset	R/W	Description	Reset Value
MAC_Address0_High	GMACx_BA+0x0040	R/W	Register 16 (MAC Address0 High Register)	0x8000FFFF

31	30	29	28	27	26	25	24
AE	**Reserved**						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ADDRHI							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADDRHI							

Bits	Description	
[31]	**AE**	**Address Enable (read only)** This bit is always set to 1.
[30:16]	**Reserved**	Reserved.
[15:0]	**ADDRHI**	**MAC Address0 [47:32]** This field contains the upper 16 bits (47:32) of the first 6-byte MAC address. The MAC uses this field for filtering the received frames and inserting the MAC address in the Transmit Flow Control (PAUSE) Frames.

GMAC Register 17 MAC Address0 Low Register (MAC_Address0_Low)

Register	Offset	R/W	Description	Reset Value
MAC_Address0_Low	GMACx_BA+0x0044	R/W	Register 17 (MAC Address0 Low Register)	0xFFFFFFFF

31	30	29	28	27	26	25	24
ADDRLO							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ADDRLO							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ADDRLO							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADDRLO							

Bits	Description	
[31:0]	**ADDRLO**	**MAC Address0 [31:0]** This field contains the lower 32 bits of the first 6-byte MAC address. This is used by the MAC for filtering the received frames and inserting the MAC address in the Transmit Flow Control (PAUSE) Frames.

GMAC Register 18 MAC Address1 High Register (MAC_Address1_High)

Register	Offset	R/W	Description	Reset Value
MAC_Address1_High	GMACx_BA+0x0048	R/W	Register 18 (MAC Address1 High Register)	0x0000FFFF

31	30	29	28	27	26	25	24
AE	**SA**	**MBC**					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ADDRHI							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADDRHI							

Bits	Description	
[31]	**AE**	**Address Enable** When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering.
[30]	**SA**	**Source Address** When this bit is set, the MAC Address1[47:0] is used to compare with the SA fields of the received frame. When this bit is reset, the MAC Address1[47:0] is used to compare with the DA fields of the received frame.
[29:24]	**MBC**	**Mask Byte Control** These bits are mask control bits for comparison of each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC Address1 registers. Each bit controls the masking of the bytes as follows: Bit 29 = Register 18[15:8] Bit 28 = Register 18[7:0] Bit 27 = Register 19[31:24] ... Bit 24 = Register 19[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
[23:16]	**Reserved**	Reserved.
[15:0]	**ADDRHI**	**MAC Address1 [47:32]** This field contains the upper 16 bits (47:32) of the second 6-byte MAC address.

GMAC Register 19 MAC Address1 Low Register (MAC_Address1_Low)

Register	Offset	R/W	Description	Reset Value
MAC_Address1_Low	GMACx_BA+0x004c	R/W	Register 19 (MAC Address1 Low Register)	0xFFFFFFFF

31	30	29	28	27	26	25	24
ADDRLO							
23	22	21	20	19	18	17	16
ADDRLO							
15	14	13	12	11	10	9	8
ADDRLO							
7	6	5	4	3	2	1	0
ADDRLO							

Bits	Description	
[31:0]	**ADDRLO**	**MAC Address1 [31:0]** This field contains the lower 32 bits of the second 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

GMAC Register 20 MAC Address2 High Register (MAC_Address2_High)

Register	Offset	R/W	Description	Reset Value
MAC_Address2_High	GMACx_BA+0x0050	R/W	Register 20 (MAC Address2 High Register)	0x0000FFFF

31	30	29	28	27	26	25	24
AE	SA	MBC					
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
ADDRHI							
7	6	5	4	3	2	1	0
ADDRHI							

Bits	Description	
[31]	**AE**	**Address Enable** When this bit is set, the address filter module uses the third MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering.
[30]	**SA**	**Source Address** When this bit is set, the MAC Address2[47:0] is used to compare with the SA fields of the received frame. When this bit is reset, the MAC Address2[47:0] is used to compare with the DA fields of the received frame.
[29:24]	**MBC**	**Mask Byte Control** These bits are mask control bits for comparison of each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC Address2 registers. Each bit controls the masking of the bytes as follows: Bit 29 = Register 20[15:8] Bit 28 = Register 20[7:0] Bit 27 = Register 21[31:24] ... Bit 24 = Register 21[7:0]
[23:16]	**Reserved**	Reserved.
[15:0]	**ADDRHI**	**MAC Address2 [47:32]** This field contains the upper 16 bits (47:32) of the third 6-byte MAC address.

GMAC Register 21 MAC Address2 Low Register (MAC_Address2_Low)

Register	Offset	R/W	Description	Reset Value
MAC_Address2_Low	GMACx_BA+0x0054	R/W	Register 21 (MAC Address2 Low Register)	0xFFFFFFFF

31	30	29	28	27	26	25	24
ADDRLO							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ADDRLO							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ADDRLO							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADDRLO							

Bits	Description	
[31:0]	**ADDRLO**	**MAC Address2 [31:0]** This field contains the lower 32 bits of the third 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

GMAC Register 22 MAC Address3 High Register (MAC_Address3_High)

Register	Offset	R/W	Description	Reset Value
MAC_Address3_High	GMACx_BA+0x0058	R/W	Register 22 (MAC Address3 High Register)	0x0000FFFF

31	30	29	28	27	26	25	24
AE	SA	MBC					
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
ADDRHI							
7	6	5	4	3	2	1	0
ADDRHI							

Bits	Description	
[31]	**AE**	**Address Enable** When this bit is set, the address filter module uses the fourth MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering.
[30]	**SA**	**Source Address** When this bit is set, the MAC Address3[47:0] is used to compare with the SA fields of the received frame. When this bit is reset, the MAC Address3[47:0] is used to compare with the DA fields of the received frame.
[29:24]	**MBC**	**Mask Byte Control** These bits are mask control bits for comparison of each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC Address3 registers. Each bit controls the masking of the bytes as follows: Bit 29 = Register 22[15:8] Bit 28 = Register 22[7:0] Bit 27 = Register 23[31:24] ... Bit 24 = Register 23[7:0]
[23:16]	**Reserved**	Reserved.
[15:0]	**ADDRHI**	**MAC Address3 [47:32]** This field contains the upper 16 bits (47:32) of the fourth 6-byte MAC address.

GMAC Register 23 MAC Address3 Low Register (MAC_Address3_Low)

Register	Offset	R/W	Description	Reset Value
MAC_Address3_Low	GMACx_BA+0x005c	R/W	Register 23 (MAC Address3 Low Register)	0xFFFFFFFF

31	30	29	28	27	26	25	24
ADDRLO							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ADDRLO							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ADDRLO							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADDRLO							

Bits	Description	
[31:0]	**ADDRLO**	**MAC Address3 [31:0]** This field contains the lower 32 bits of the fourth 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

GMAC Register 24 MAC Address4 High Register (MAC_Address4_High)

Register	Offset	R/W	Description	Reset Value
MAC_Address4_High	GMACx_BA+0x0060	R/W	Register 24 (MAC Address4 High Register)	0x0000FFFF

31	30	29	28	27	26	25	24
AE	SA	MBC					
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
ADDRHI							
7	6	5	4	3	2	1	0
ADDRHI							

Bits	Description	
[31]	**AE**	**Address Enable** When this bit is set, the address filter module uses the fifth MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering.
[30]	**SA**	**Source Address** When this bit is set, the MAC Address4[47:0] is used to compare with the SA fields of the received frame. When this bit is reset, the MAC Address4[47:0] is used to compare with the DA fields of the received frame.
[29:24]	**MBC**	**Mask Byte Control** These bits are mask control bits for comparison of each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC Address4 registers. Each bit controls the masking of the bytes as follows: Bit 29 = Register 24[15:8] Bit 28 = Register 24[7:0] Bit 27 = Register 25[31:24] ... Bit 24 = Register 25[7:0]
[23:16]	**Reserved**	Reserved.
[15:0]	**ADDRHI**	**MAC Address4 [47:32]** This field contains the upper 16 bits (47:32) of the fifth 6-byte MAC address.

GMAC Register 25 MAC Address4 Low Register (MAC_Address4_Low)

Register	Offset	R/W	Description	Reset Value
MAC_Address4_Low	GMACx_BA+0x0064	R/W	Register 25 (MAC Address4 Low Register)	0xFFFFFFFF

31	30	29	28	27	26	25	24
ADDRLO							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ADDRLO							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ADDRLO							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADDRLO							

Bits	Description	
[31:0]	**ADDRLO**	**MAC Address4 [31:0]** This field contains the lower 32 bits of the fifth 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

GMAC Register 26 MAC Address5 High Register (MAC_Address5_High)

Register	Offset	R/W	Description	Reset Value
MAC_Address5_High	GMACx_BA+0x0068	R/W	Register 26 (MAC Address5 High Register)	0x0000FFFF

31	30	29	28	27	26	25	24
AE	SA	MBC					
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
ADDRHI							
7	6	5	4	3	2	1	0
ADDRHI							

Bits	Description	
[31]	**AE**	**Address Enable** When this bit is set, the address filter module uses the sixth MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering.
[30]	**SA**	**Source Address** When this bit is set, the MAC Address5[47:0] is used to compare with the SA fields of the received frame. When this bit is reset, the MAC Address5[47:0] is used to compare with the DA fields of the received frame.
[29:24]	**MBC**	**Mask Byte Control** These bits are mask control bits for comparison of each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC Address5 registers. Each bit controls the masking of the bytes as follows: Bit 29 = Register 26[15:8] Bit 28 = Register 26[7:0] Bit 27 = Register 27[31:24] ... Bit 24 = Register 27[7:0]
[23:16]	**Reserved**	Reserved.
[15:0]	**ADDRHI**	**MAC Address5 [47:32]** This field contains the upper 16 bits (47:32) of the sixth 6-byte MAC address.

GMAC Register 27 MAC Address5 Low Register (MAC_Address5_Low)

Register	Offset	R/W	Description	Reset Value
MAC_Address5_Low	GMACx_BA+0x006c	R/W	Register 27 (MAC Address5 Low Register)	0xFFFFFFFF

31	30	29	28	27	26	25	24
ADDRLO							
23	22	21	20	19	18	17	16
ADDRLO							
15	14	13	12	11	10	9	8
ADDRLO							
7	6	5	4	3	2	1	0
ADDRLO							

Bits	Description	
[31:0]	**ADDRLO**	**MAC Address5 [31:0]** This field contains the lower 32 bits of the sixth 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

GMAC Register 28 MAC Address6 High Register (MAC_Address6_High)

Register	Offset	R/W	Description	Reset Value
MAC_Address6_High	GMACx_BA+0x0070	R/W	Register 28 (MAC Address6 High Register)	0x0000FFFF

31	30	29	28	27	26	25	24
AE	SA	MBC					
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
ADDRHI							
7	6	5	4	3	2	1	0
ADDRHI							

Bits	Description	
[31]	**AE**	**Address Enable** When this bit is set, the address filter module uses the seventh MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering.
[30]	**SA**	**Source Address** When this bit is set, the MAC Address6[47:0] is used to compare with the SA fields of the received frame. When this bit is reset, the MAC Address6[47:0] is used to compare with the DA fields of the received frame.
[29:24]	**MBC**	**Mask Byte Control** These bits are mask control bits for comparison of each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC Address6 registers. Each bit controls the masking of the bytes as follows: Bit 29 = Register 28[15:8] Bit 28 = Register 28[7:0] Bit 27 = Register 29[31:24] ... Bit 24 = Register 29[7:0]
[23:16]	**Reserved**	Reserved.
[15:0]	**ADDRHI**	**MAC Address6 [47:32]** This field contains the upper 16 bits (47:32) of the seventh 6-byte MAC address.

GMAC Register 29 MAC Address6 Low Register (MAC_Address6_Low)

Register	Offset	R/W	Description	Reset Value
MAC_Address6_Low	GMACx_BA+0x0074	R/W	Register 29 (MAC Address6 Low Register)	0xFFFFFFFF

31	30	29	28	27	26	25	24
ADDRLO							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ADDRLO							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ADDRLO							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADDRLO							

Bits	Description	
[31:0]	**ADDRLO**	**MAC Address6 [31:0]** This field contains the lower 32 bits of the seventh 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

GMAC Register 30 MAC Address7 High Register (MAC_Address7_High)

Register	Offset	R/W	Description	Reset Value
MAC_Address7_High	GMACx_BA+0x0078	R/W	Register 30 (MAC Address7 High Register)	0x0000FFFF

31	30	29	28	27	26	25	24
AE	**SA**	**MBC**					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ADDRHI							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADDRHI							

Bits	Description	
[31]	**AE**	**Address Enable** When this bit is set, the address filter module uses the eighth MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering.
[30]	**SA**	**Source Address** When this bit is set, the MAC Address7[47:0] is used to compare with the SA fields of the received frame. When this bit is reset, the MAC Address7[47:0] is used to compare with the DA fields of the received frame.
[29:24]	**MBC**	**Mask Byte Control** These bits are mask control bits for comparison of each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC Address6 registers. Each bit controls the masking of the bytes as follows: Bit 29 = Register 30[15:8] Bit 28 = Register 30[7:0] Bit 27 = Register 31[31:24] ... Bit 24 = Register 31[7:0]
[23:16]	**Reserved**	Reserved.
[15:0]	**ADDRHI**	**MAC Address7 [47:32]** This field contains the upper 16 bits (47:32) of the eighth 6-byte MAC address.

GMAC Register 31 MAC Address7 Low Register (MAC_Address7_Low)

Register	Offset	R/W	Description	Reset Value
MAC_Address7_Low	GMACx_BA+0x007c	R/W	Register 31 (MAC Address7 Low Register)	0xFFFFFFFF

31	30	29	28	27	26	25	24
ADDRLO							
23	22	21	20	19	18	17	16
ADDRLO							
15	14	13	12	11	10	9	8
ADDRLO							
7	6	5	4	3	2	1	0
ADDRLO							

Bits	Description	
[31:0]	**ADDRLO**	**MAC Address7 [31:0]** This field contains the lower 32 bits of the eighth 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

GMAC Register 32 MAC Address8 High Register (MAC_Address8_High)

Register	Offset	R/W	Description	Reset Value
MAC_Address8_High	GMACx_BA+0x0080	R/W	Register 32 (MAC Address8 High Register)	0x0000FFFF

31	30	29	28	27	26	25	24
AE	**SA**	**MBC**					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ADDRHI							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADDRHI							

Bits	Description	
[31]	**AE**	**Address Enable** When this bit is set, the address filter module uses the nineth MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering.
[30]	**SA**	**Source Address** When this bit is set, the MAC Address8[47:0] is used to compare with the SA fields of the received frame. When this bit is reset, the MAC Address8[47:0] is used to compare with the DA fields of the received frame.
[29:24]	**MBC**	**Mask Byte Control** These bits are mask control bits for comparison of each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC Address8 registers. Each bit controls the masking of the bytes as follows: Bit 29 = Register 32[15:8] Bit 28 = Register 32[7:0] Bit 27 = Register 33[31:24] ... Bit 24 = Register 33[7:0]
[23:16]	**Reserved**	Reserved.
[15:0]	**ADDRHI**	**MAC Address8 [47:32]** This field contains the upper 16 bits (47:32) of the nineth 6-byte MAC address.

GMAC Register 33 MAC Address8 Low Register (MAC_Address8_Low)

Register	Offset	R/W	Description	Reset Value
MAC_Address8_Low	GMACx_BA+0x0084	R/W	Register 33 (MAC Address8 Low Register)	0xFFFFFFFF

31	30	29	28	27	26	25	24
ADDRLO							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ADDRLO							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ADDRLO							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADDRLO							

Bits	Description	
[31:0]	**ADDRLO**	**MAC Address8 [31:0]** This field contains the lower 32 bits of the nineth 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

GMAC Register 54 RGMII Status Register (RGMII_Control_Status)

Register	Offset	R/W	Description	Reset Value
RGMII_Control_Status	GMACx_BA+0x00d8	R	Register 54 (RGMII Status Register)	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				LNKSTS	LNKSPEED		LNKMOD

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3]	**LNKSTS**	**Link Status** When set, this bit indicates that the link is up between the local PHY and the remote PHY. When cleared, this bit indicates that the link is down between the local PHY and the remote PHY.
[2:1]	**LNKSPEED**	**Link Speed** This bit indicates the current speed of the link: 00 = 2.5 MHz 01 = 25 MHz 10 = 125 MHz
[0]	**LNKMOD**	**Link Mode** This bit indicates the current mode of operation of the link: 1'b0 = Half-duplex mode 1'b1 = Full-duplex mode

GMAC Register 55 Watchdog Timeout Register (WDog_Timeout)

Register	Offset	R/W	Description	Reset Value
WDog_Timeout	GMACx_BA+0x00dc	R/W	Register 55 (Watchdog Timeout Register) This register controls the watchdog timeout for received frames	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							**PWE**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		**WTO**					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WTO							

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**PWE**	**Programmable Watchdog Enable** When this bit is set and Bit 23 (WD) of Register 0 (MAC Configuration Register) is reset, the WTO field (Bits[13:0]) is used as watchdog timeout for a received frame. When this bit is cleared, the watchdog timeout for a received frame is controlled by the setting of Bit 23 (WD) and Bit 20 (JE) in Register 0 (MAC Configuration Register).
[15:14]	**Reserved**	Reserved.
[13:0]	**WTO**	**Watchdog Timeout** When Bit 16 (PWE) is set and Bit 23 (WD) of Register 0 (MAC Configuration Register) is reset, this field is used as watchdog timeout for a received frame. If the length of a received frame exceeds the value of this field, such frame is terminated and declared as an error frame. **Note:** When Bit 16 (PWE) is set, the value in this field should be more than 1,522 (0x05F2). Otherwise, the IEEE Std 802.3-specified valid tagged frames are declared as error frames and are dropped.

GMAC Register 353 VLAN Tag Inclusion or Replacement Register (VLAN_Incl_Reg)

Register	Offset	R/W	Description	Reset Value
VLAN_Incl_Reg	GMACx_BA+0x0584	R/W	Register 353 (VLAN Tag Inclusion or Replacement Register)	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				CSVL	VLP	VLC	
15	14	13	12	11	10	9	8
VLT							
7	6	5	4	3	2	1	0
VLT							

Bits	Description	
[31:20]	**Reserved**	Reserved.
[19]	**CSVL**	**C-VLAN or S-VLAN** When this bit is set, S-VLAN type (0x88A8) is inserted or replaced in the 13th and 14th bytes of transmitted frames. When this bit is reset, C-VLAN type (0x8100) is inserted or replaced in the transmitted frames.
[18]	**VLP**	**VLAN Priority Control** When this bit is set, the control Bits [17:16] are used for VLAN deletion, insertion, or replacement. When this bit is reset, the mti_vlan_ctrl_i control input is used, and Bits [17:16] are ignored.
[17:16]	**VLC**	**VLAN Tag Control in Transmit Frames** 2'b00 = No VLAN tag deletion, insertion, or replacement 2'b01 = VLAN tag deletion The MAC removes the VLAN type (bytes 13 and 14) and VLAN tag (bytes 15 and 16) of all transmitted frames with VLAN tags. 2'b10 = VLAN tag insertion The MAC inserts VLT in bytes 15 and 16 of the frame after inserting the Type value (0x8100/0x88a8) in bytes 13 and 14. This operation is performed on all transmitted frames, irrespective of whether they already have a VLAN tag. 2'b11 = VLAN tag replacement The MAC replaces VLT in bytes 15 and 16 of all VLAN-type transmitted frames (Bytes 13 and 14 are 0x8100/0x88a8). **Note:** Changes to this field take effect only on the start of a frame. If you write this register field when a frame is being transmitted, only the subsequent frame can use the updated value, that is, the current frame does not use the updated value
[15:0]	**VLT**	**VLAN Tag for Transmit Frames** This field contains the value of the VLAN tag to be inserted or replaced. The value must only be changed when the transmit lines are inactive or during the initialization phase. Bits[15:13] are the User Priority, Bit 12 is the CFI/DEI, and Bits[11:0] are the VLAN tag's VID field.

GMAC Register 448 Timestamp Control Register (Timestamp_Control)

Register	Offset	R/W	Description	Reset Value
Timestamp_Control	GMACx_BA+0x0700	R/W	Register 448 (Timestamp Control Register)	0x00002000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved					TSENMACADDR	SNAPTYPSEL	
15	14	13	12	11	10	9	8
TSMSTRENA	TSEVNTENA	TSIPV4ENA	TSIPV6ENA	TSIPENA	TSVER2ENA	TSCTRLSSR	TSENALL
7	6	5	4	3	2	1	0
Reserved		TSADDREG	TSTRIG	TSUPDT	TSINIT	TSCFUPDT	TSENA

Bits	Description	
[31:19]	**Reserved**	Reserved.
[18]	**TSENMACADDR**	**Enable MAC address for PTP Frame Filtering** When set, the DA MAC address (that matches any MAC Address register) is used to filter the PTP frames when PTP is directly sent over Ethernet.
[17:16]	**SNAPTYPSEL**	**Select PTP packets for Taking Snapshots** These bits along with Bits 15 and 14 decide the set of PTP packet types for which snapshot needs to be taken.
[15]	**TSMSTRENA**	**Enable Snapshot for Messages Relevant to Master** When set, the snapshot is taken only for the messages relevant to the master node. Otherwise, the snapshot is taken for the messages relevant to the slave node.
[14]	**TSEVNTENA**	**Enable Timestamp Snapshot for Event Messages** When set, the timestamp snapshot is taken only for event messages (SYNC, Delay_Req, Pdelay_Req, or Pdelay_Resp). When reset, the snapshot is taken for all messages except Announce, Management, and Signaling.
[13]	**TSIPV4ENA**	**Enable Processing of PTP Frames Sent over IPv4-UDP** When set, the MAC receiver processes the PTP packets encapsulated in UDP over IPv4 packets. When this bit is clear, the MAC ignores the PTP transported over UDP-IPv4 packets. This bit is set by default.
[12]	**TSIPV6ENA**	**Enable Processing of PTP Frames Sent Over IPv6-UDP** When set, the MAC receiver processes PTP packets encapsulated in UDP over IPv6 packets. When this bit is clear, the MAC ignores the PTP transported over UDP-IPv6 packets.
[11]	**TSIPENA**	**Enable Processing of PTP over Ethernet Frames** When set, the MAC receiver processes the PTP packets encapsulated directly in the Ethernet frames. When this bit is clear, the MAC ignores the PTP over Ethernet packets.
[10]	**TSVER2ENA**	**Enable PTP packet Processing for Version 2 Format** When set, the PTP packets are processed using the 1588 version 2 format. Otherwise, the PTP packets are processed using the version 1 format.

[9]	**TSCTRLSSR**	**Timestamp Digital or Binary Rollover Control** When set, the Timestamp Low register rolls over after 0x3B9A_C9FF value (that is, 1 nanosecond accuracy) and increments the timestamp (High) seconds. When reset, the rollover value of sub-second register is 0x7FFF_FFFF. The sub-second increment has to be programmed correctly depending on the PTP reference clock frequency and the value of this bit.
[8]	**TSENALL**	**Enable Timestamp for All Frames** When set, the timestamp snapshot is enabled for all frames received by the MAC.
[7:6]	**Reserved**	Reserved.
[5]	**TSADDREG**	**Addend Reg Update** When set, the content of the Timestamp Addend register is updated in the PTP block for fine correction. This is cleared when the update is completed. This register bit should be zero before setting it.
[4]	**TSTRIG**	**Timestamp Interrupt Trigger Enable** When set, the timestamp interrupt is generated when the System Time becomes greater than the value written in the Target Time register. This bit is reset after the generation of the Timestamp Trigger Interrupt.
[3]	**TSUPDT**	**Timestamp Update** When set, the system time is updated (added or subtracted) with the value specified in Register 452 (System Time - Seconds Update Register) and Register 453 (System Time - Nanoseconds Update Register). This bit should be read zero before updating it. This bit is reset when the update is completed in hardware. The Timestamp Higher Word register is not updated.
[2]	**TSINIT**	**Timestamp Initialize** When set, the system time is initialized (overwritten) with the value specified in the Register 452 (System Time - Seconds Update Register) and Register 453 (System Time - Nanoseconds Update Register). This bit should be read zero before updating it. This bit is reset when the initialization is complete. The Timestamp Higher Word register can only be initialized.
[1]	**TSCFUPDT**	**Timestamp Fine or Coarse Update** When set, this bit indicates that the system times update should be done using the fine update method. When reset, it indicates the system timestamp update should be done using the Coarse method.
[0]	**TSENA**	**Timestamp Enable** When set, the timestamp is added for the transmit and receive frames. When disabled, timestamp is not added for the transmit and receive frames and the Timestamp Generator is also suspended. You need to initialize the Timestamp (system time) after enabling this mode. On the receive side, the MAC processes the 1588 frames only if this bit is set.

GMAC Register 449 Sub-second Increment Register (Sub_Second_Increment)

Register	Offset	R/W	Description	Reset Value
Sub_Second_Increment	GMACx_BA+0x0704	R/W	Register 449 (Sub-Second Increment Register)	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SSINC							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**SSINC**	**Sub-second Increment Value** The value programmed in this field is accumulated every clock cycle (of clk_ptp_i) with the contents of the sub-second register. For example, when PTP clock is 50 MHz (period is 20 ns), you should program 20 (0x14) when the System Time-Nanoseconds register has an accuracy of 1 ns (TSCTRLSSR bit is set). When TSCTRLSSR is clear, the Nanoseconds register has a resolution of ~0.465ns. In this case, you should program a value of 43 (0x2B) that is derived by 20ns/0.465.

GMAC Register 450 System Time - Seconds Register (System_Time_Seconds)

Register	Offset	R/W	Description	Reset Value
System_Time_Seconds	GMACx_BA+0x0708	R	Register 450 (System Time - Seconds Register)	0x00000000

31	30	29	28	27	26	25	24
TSS							
23	22	21	20	19	18	17	16
TSS							
15	14	13	12	11	10	9	8
TSS							
7	6	5	4	3	2	1	0
TSS							

Bits	Description	
[31:0]	TSS	**Timestamp Second** The value in this field indicates the current value in seconds of the System Time maintained by the MAC.

GMAC Register 451 System Time - Nanoseconds Register (System_Time_Nanoseconds)

Register	Offset	R/W	Description	Reset Value
System_Time_Nanoseconds	GMACx_BA+0x070c	R	Register 451 (System Time - Nanoseconds Register)	0x00000000

31	30	29	28	27	26	25	24
Reserved	TSSS						
23	22	21	20	19	18	17	16
TSSS							
15	14	13	12	11	10	9	8
TSSS							
7	6	5	4	3	2	1	0
TSSS							

Bits	Description	
[31]	**Reserved**	Reserved.
[30:0]	**TSSS**	**Timestamp Sub Seconds** The value in this field has the sub second representation of time, with an accuracy of 0.46 ns. When bit 9 (TSCTRLSSR) is set in Register 448 (Timestamp Control Register), each bit represents 1 ns and the maximum value is 0x3B9A_C9FF, after which it rolls-over to zero.

GMAC Register 452 System Time - Seconds Update Register (System_Time_Seconds_Update)

Register	Offset	R/W	Description	Reset Value
System_Time_Seconds_Update	GMACx_BA+0x0710	R/W	Register 452 (System Time - Seconds Update Register)	0x00000000

31	30	29	28	27	26	25	24
TSS							
23	22	21	20	19	18	17	16
TSS							
15	14	13	12	11	10	9	8
TSS							
7	6	5	4	3	2	1	0
TSS							

Bits	Description	
[31:0]	TSS	**Timestamp Second** The value in this field indicates the time in seconds to be initialized or added to the system time.

GMAC Register 453 System Time - Nanoseconds Update Register (System_Time_Nanoseconds_Update)

Register	Offset	R/W	Description	Reset Value
System_Time_Nanoseconds_Update	GMACx_BA+0x0714	R/W	Register 453 (System Time - Nanoseconds Update Register)	0x00000000

31	30	29	28	27	26	25	24
ADDSUB	TSSS						
23	22	21	20	19	18	17	16
TSSS							
15	14	13	12	11	10	9	8
TSSS							
7	6	5	4	3	2	1	0
TSSS							

Bits	Description	
[31]	ADDSUB	**Add or subtract time** When this bit is set, the time value is subtracted with the contents of the update register. When this bit is reset, the time value is added with the contents of the update register.
[30:0]	TSSS	**Timestamp Sub Second** The value in this field has the sub second representation of time, with an accuracy of 0.46 ns. When bit 9 (TSCTRLSSR) is set in Register 448 (Timestamp Control Register), each bit represents 1 ns and the programmed value should not exceed 0x3B9A_C9FF.

GMAC Register 454 Timestamp Addend Register (Timestamp_Addend)

Register	Offset	R/W	Description	Reset Value
Timestamp_Addend	GMACx_BA+0x0718	R/W	Register 454 (Timestamp Addend Register)	0x00000000

31	30	29	28	27	26	25	24
TSAR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TSAR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TSAR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TSAR							

Bits	Description	
[31:0]	**TSAR**	**Timestamp Addend Register** This field indicates the 32-bit time value to be added to the Accumulator register to achieve time synchronization.

GMAC Register 455 Target Time Seconds Register (Target_Time_Seconds)

Register	Offset	R/W	Description	Reset Value
Target_Time_Seconds	GMACx_BA+0x071c	R/W	Register 455 (Target Time Seconds Register)	0x00000000

31	30	29	28	27	26	25	24
TSTR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TSTR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TSTR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TSTR							

Bits	Description	
[31:0]	**TSTR**	**Target Time Seconds Register** This register stores the time in seconds. When the timestamp value matches or exceeds both Target Timestamp registers, then based on Bits [6:5] of Register 459 (PPS Control Register), the MAC starts or stops the PPS signal output and generates an interrupt (if enabled).

GMAC Register 456 Target Time Nanoseconds Register (Target_Time_Nanoseconds)

Register	Offset	R/W	Description	Reset Value
Target_Time_Nanoseconds	GMACx_BA+0x0720	R/W	Register 456 (Target Time Nanoseconds Register)	0x00000000

31	30	29	28	27	26	25	24
TRGTBUSY	TTSLO						
23	22	21	20	19	18	17	16
TTSLO							
15	14	13	12	11	10	9	8
TTSLO							
7	6	5	4	3	2	1	0
TTSLO							

Bits	Description	
[31]	**TRGTBUSY**	**Target Time Register Busy (read only)** The MAC sets this bit when the PPSCMD field (Bits[3:0]) in Register 459 (PPS Control Register) is programmed to 010 or 011. Programming the PPSCMD field to 010 or 011, instructs the MAC to synchronize the Target Time Registers to the PTP clock domain. The MAC clears this bit after synchronizing the Target Time Registers to the PTP clock domain The application must not update the Target Time Registers when this bit is read as 1. Otherwise, the synchronization of the previous programmed time gets corrupted. This bit is reserved when the Enable Flexible Pulse-Per-Second Output feature is not selected.
[30:0]	**TTSLO**	**Target Timestamp Low Register** This register stores the time in (signed) nanoseconds. When the value of the timestamp matches the both Target Timestamp registers, then based on the TRGTMODSEL0 field (Bits [6:5]) in Register 459 (PPS Control Register), the MAC starts or stops the PPS signal output and generates an interrupt (if enabled). This value should not exceed 0x3B9A_C9FF when TSCTRLSSR is set in the Timestamp control register. The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value.

GMAC Register 457 System Time - Higher Word Seconds Register (System_Time_Higher_Word_Seconds)

Register	Offset	R/W	Description	Reset Value
System_Time_Higher_Word_Seconds	GMACx_BA+0x0724	R/W	Register 457 (System Time - Higher Word Seconds Register)	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TSHWR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TSHWR							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**TSHWR**	**Timestamp Higher Word Register** This field contains the most significant 16-bits of the timestamp seconds value. The register is directly written to initialize the value. This register is incremented when there is an overflow from the 32-bits of the System Time - Seconds register.

GMAC Register 458 Timestamp Status Register (Timestamp_Status)

Register	Offset	R/W	Description	Reset Value
Timestamp_Status	GMACx_BA+0x0728	R	Register 458 (Timestamp Status Register)	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				**ATSSTN**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**TSTRGTERR**	**AUXTSTRIG**	**TSTARGT**	**TSSOVF**

Bits	Description	
[31:20]	**Reserved**	Reserved.
[19:16]	**ATSSTN**	**Auxiliary Timestamp Snapshot Trigger Identifier** These bits identify the Auxiliary trigger inputs for which the timestamp available in the Auxiliary Snapshot Register is applicable. When more than one bit is set at the same time, it means that corresponding auxiliary triggers were sampled at the same clock. These bits are applicable only if the number of Auxiliary snapshots is more than one. One bit is assigned for each trigger as shown in the following list: Bit 16 = Auxiliary trigger 0 Bit 17 = Auxiliary trigger 1 Bit 18 = Auxiliary trigger 2 Bit 19 = Auxiliary trigger 3 The software can read this register to find the triggers that are set when the timestamp is taken.
[15:4]	**Reserved**	Reserved.
[3]	**TSTRGTERR**	**Timestamp Target Time Error** This bit is set when the target time, being programmed in Target Time Registers, is already elapsed. This bit is cleared when read by the application.
[2]	**AUXTSTRIG**	**Auxiliary Timestamp Trigger Snapshot** This bit is set high when the auxiliary snapshot is written to the FIFO. This bit is valid only if the Enable IEEE 1588 Auxiliary Snapshot feature is selected.
[1]	**TSTARGT**	**Timestamp Target Time Reached** When set, this bit indicates that the value of system time is greater or equal to the value specified in the Register 455 (Target Time Seconds Register) and Register 456 (Target Time Nanoseconds Register).
[0]	**TSSOVF**	**Timestamp Seconds Overflow** When set, this bit indicates that the seconds value of the timestamp (when supporting version 2 format) has overflowed beyond 32'hFFFF_FFFF.

GMAC Register 459 PPS Control Register (PPS_Control)

Register	Offset	R/W	Description	Reset Value
PPS_Control	GMACx_BA+0x072c	R/W	Register 459 (PPS Control Register)	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	TRGTMODSEL0		PPSEN0	PPSCTRL_PPSCMD			

Bits	Description	
[31:7]	**Reserved**	Reserved.
[6:5]	**TRGTMODSEL0**	**Target Time Register Mode for PPS0 Output** This field indicates the Target Time registers (register 455 and 456) mode for PPS0 output signal: 00 = Indicates that the Target Time registers are programmed only for generating the interrupt event. 01 = Reserved 10 = Indicates that the Target Time registers are programmed for generating the interrupt event and starting or stopping the generation of the PPS0 output signal. 11 = Indicates that the Target Time registers are programmed only for starting or stopping the generation of the PPS0 output signal. No interrupt is asserted.
[4]	**PPSEN0**	**Flexible PPS Output Mode Enable** When set low, Bits[3:0] function as PPSCTRL (backward compatible). When set high, Bits[3:0] function as PPSCMD.

[3:0]	**PPSCTRL_PPSCMD**	**PPSCTRL0 or PPSCMD0** PPSCTRL0: PPS0 Output Frequency Control This field controls the frequency of the PPS0 output (ptp_pps_o[0]) signal. The default value of PPSCTRL is 0000, and the PPS output is 1 pulse (of width clk_ptp_i) every second. For other values of PPSCTRL, the PPS output becomes a generated clock of following frequencies: 0000 = The binary rollover is 1 Hz, and the digital rollover is 0.5 Hz. 0001 = The binary rollover is 2 Hz, and the digital rollover is 1 Hz. 0010 = The binary rollover is 4 Hz, and the digital rollover is 2 Hz. 0011 = The binary rollover is 8 Hz, and the digital rollover is 4 Hz. 0100 = The binary rollover is 16 Hz, and the digital rollover is 8 Hz. ... 1111 = The binary rollover is 32.768 kHz, and the digital rollover is 16.384 kHz. **Note:** In the binary rollover mode, the PPS output (ptp_pps_o) has a duty cycle of 50 percent with these frequencies. In the digital rollover mode, the PPS output frequency is an average number. The actual clock is of different frequency that gets synchronized every second. For example: * When PPSCTRL = 0001, the PPS (1 Hz) has a low period of 537 ms and a high period of 463 ms. * When PPSCTRL = 0010, the PPS (2 Hz) is a sequence of:. One clock of 50 percent duty cycle and 537 ms period Second clock of 463 ms period (268 ms low and 195 ms high) * When PPSCTRL = 0011, the PPS (4 Hz) is a sequence of:. Three clocks of 50 percent duty cycle and 268 ms period Fourth clock of 195 ms period (134 ms low and 61 ms high) This behavior is because of the non-linear toggling of bits in the digital rollover mode in Register 451 (System Time - Nanoseconds Register). Flexible PPS0 Output (ptp_pps_o[0]) Control Programming these bits with a non-zero value instructs the MAC to initiate an event. Once the command is transferred or synchronized to the PTP clock domain, these bits get cleared automatically. The Software should ensure that these bits are programmed only when they are all-zero. The following list describes the values of PPSCMD0: 0000 = No Command 0001 = START Single Pulse This command generates single pulse rising at the start point defined in Target Time Registers (register 455 and 456) and of a duration defined in the PPS0 Width Register. 0010 = START Pulse Train This command generates the train of pulses rising at the start point defined in the Target Time Registers and of a duration defined in the PPS0 Width Register and repeated at interval defined in the PPS Interval Register. By default, the PPS pulse train is free-running unless stopped by 'STOP Pulse train at time' or 'STOP Pulse Train immediately' commands. 0011 = Cancel START This command cancels the START Single Pulse and START Pulse Train commands if the system time has not crossed the programmed start time. 0100 = STOP Pulse Train at time This command stops the train of pulses initiated by the START Pulse Train command (PPSCMD = 0010) after the time programmed in the Target Time registers elapses. 0101 = STOP Pulse Train immediately This command immediately stops the train of pulses initiated by the START Pulse Train command (PPSCMD = 0010). 0110 = Cancel STOP Pulse Train This command cancels the STOP pulse train at time command if the programmed stop time has not elapsed. The PPS pulse train becomes free-running on the successful execution of this command.

		0111-1111 = Reserved

GMAC Register 472 PPS0 Interval Register (PPS0_Interval)

Register	Offset	R/W	Description	Reset Value
PPS0_Interval	GMACx_BA+0x0760	R/W	Register 472 (PPS0 Interval Register)	0x00000000

31	30	29	28	27	26	25	24
PPSINT							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PPSINT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PPSINT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PPSINT							

Bits	Description	
[31:0]	**PPSINT**	**PPS0 Output Signal Interval** These bits store the interval between the rising edges of PPS0 signal output in terms of units of sub-second increment value. You need to program one value less than the required interval. For example, if the PTP reference clock is 50 MHz (period of 20ns), and desired interval between rising edges of PPS0 signal output is 100ns (that is, five units of sub-second increment value), then you should program value 4 (5 -1) in this register.

GMAC Register 473 PPS0 Width Register (PPS0_Width)

Register	Offset	R/W	Description	Reset Value
PPS0_Width	GMACx_BA+0x0764	R/W	Register 473 (PPS0 Width Register)	0x00000000

31	30	29	28	27	26	25	24
PPSWIDTH							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PPSWIDTH							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PPSWIDTH							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PPSWIDTH							

Bits	Description	
[31:0]	**PPSWIDTH**	**PPS0 Output Signal Width** These bits store the width between the rising edge and corresponding falling edge of the PPS0 signal output in terms of units of sub-second increment value. You need to program one value less than the required interval. For example, if PTP reference clock is 50 MHz (period of 20ns), and desired width between the rising and corresponding falling edges of PPS0 signal output is 80ns (that is, four units of sub-second increment value), then you should program value 3 (4-1) in this register. **Note:** The value programmed in this register must be lesser than the value programmed in Register 472 (PPS0 Interval Register).

GMAC Register 1000 Bus Mode Register (Bus_Mode)

Register	Offset	R/W	Description	Reset Value
Bus_Mode	GMACx_BA+0x1000	R/W	Register 1000 (Bus Mode Register)	0x00020101

31	30	29	28	27	26	25	24
Reserved						AAL	PBLx8
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
USP	RPBL						FB
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		PBL					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ATDS	DSL					Reserved	SWR

Bits	Description	
[31:26]	**Reserved**	Reserved.
[25]	**AAL**	**Address Aligned Beats** When this bit is set high and the FB bit is equal to 1, the AHB or AXI interface generates all bursts aligned to the start address LS bits. If the FB bit is equal to 0, the first burst (accessing the data buffer's start address) is not aligned, but subsequent bursts are aligned to the address.
[24]	**PBLx8**	**PBLx8 Mode** When set high, this bit multiplies the programmed PBL value (Bits[22:17] and Bits[13:8]) eight times. Therefore, the DMA transfers the data in 8, 16, 32, 64, 128, and 256 beats depending on the PBL value.
[23]	**USP**	**Use Separate PBL** When set high, this bit configures the Rx DMA to use the value configured in Bits[22:17] as PBL. The PBL value in Bits[13:8] is applicable only to the Tx DMA operations. When reset to low, the PBL value in Bits[13:8] is applicable for both DMA engines.
[22:17]	**RPBL**	**Rx DMA PBL** This field indicates the maximum number of beats to be transferred in one Rx DMA transaction. This is the maximum value that is used in a single block Read or Write. The Rx DMA always attempts to burst as specified in the RPBL bit each time it starts a Burst transfer on the host bus. You can program RPBL with values of 1, 2, 4, 8, 16, and 32. Any other value results in undefined behavior. This field is valid and applicable only when USP is set high.
[16]	**FB**	**Fixed Burst** This bit controls whether the AHB or AXI Master interface performs fixed burst transfers or not. When set, the AHB interface uses only SINGLE, INCR4, INCR8, or INCR16 during start of the normal burst transfers. When reset, the AHB or AXI interface uses SINGLE and INCR burst transfer operations. For more information, see Bit 0 (UNDEF) of the AXI Bus Mode register.
[15:14]	**Reserved**	Reserved.

[13:8]	**PBL**	**Programmable Burst Length** These bits indicate the maximum number of beats to be transferred in one DMA transaction. This is the maximum value that is used in a single block Read or Write. The DMA always attempts to burst as specified in PBL each time it starts a Burst transfer on the host bus. PBL can be programmed with permissible values of 1, 2, 4, 8, 16, and 32. Any other value results in undefined behavior. When USP is set high, this PBL value is applicable only for Tx DMA transactions. If the number of beats to be transferred is more than 32, then perform the following steps: 1. Set the PBLx8 mode. 2. Set the PBL. For example, if the maximum number of beats to be transferred is 64, then first set PBLx8 to 1 and then set PBL to 8. The PBL values have the following limitation: The maximum number of possible beats (PBL) is limited by the size of the Tx FIFO and Rx FIFO in the MTL layer and the data bus width on the DMA. The FIFO has a constraint that the maximum beat supported is half the depth of the FIFO, except when specified. For different data bus widths and FIFO sizes, the valid PBL range (including x8 mode) is provided in the following list. If the PBL is common for both transmit and receive DMA, the minimum Rx FIFO and Tx FIFO depths must be considered. **Note:** In the half-duplex mode, the valid PBL range specified in the following list is applicable only for Tx FIFO. * 32-Bit Data Bus Width ● 128 Bytes FIFO Depth: In the full-duplex mode, the valid PBL range is 16 or less. In the half-duplex mode, the valid PBL range is 8 or less for the 10 or 100 Mbps mode. ● 256 Bytes FIFO Depth: In the full-duplex mode and the half-duplex (10 or 100 Mbps) modes, the valid PBL range is 32 or less. ● 512 Bytes FIFO Depth: In the full-duplex mode and the half-duplex (10 or 100 Mbps) modes, the valid PBL range is 64 or less. ● 1 Kbytes FIFO Depth: In the full-duplex mode, the valid PBL range is 128 or less. In the half-duplex mode, the valid PBL range is 128 or less in the 10 or 100 Mbps mode and 64 or less in the 1000 Mbps mode. ● 2 Kbytes and Higher FIFO Depth: All PBL values are supported in the full-duplex mode and half-duplex modes. * 64-Bit Data Bus Width ● 128 Bytes FIFO Depth: In the full-duplex mode, the valid PBL range is 8 or less. In the half-duplex mode, the valid PBL range is 4 or less for the 10 or 100 Mbps mode. ● 256 Bytes FIFO Depth: In the full-duplex mode and the half-duplex (10 or 100 Mbps) modes, the valid PBL range is 16 or less. ● 512 Bytes FIFO Depth: In the full-duplex mode and the half-duplex (10 or 100 Mbps) modes, the valid PBL range is 32 or less. ● 1 Kbytes FIFO Depth: In the full-duplex mode, the valid PBL range is 64 or less. In the half-duplex mode, the valid PBL range is 64 or less in the 10 or 100 Mbps mode and 32 or less in the 1000-Mbps mode. ● 2 Kbytes FIFO Depth: In the full-duplex mode and the half-duplex (10 or 100 Mbps) modes, the valid PBL range is 128 or less. ● 4 Kbytes and Higher FIFO Depth: All PBL values are supported in the full-duplex and half-duplex modes. * 128-Bit Data Bus Width ● 128 Bytes FIFO Depth: In the full-duplex mode, the valid PBL range is 4 or less. In the half-duplex mode, the valid PBL range is 2 or less for the 10 or 100 Mbps mode. ● 256 Bytes FIFO Depth: In the full-duplex mode and the half-duplex (10 or 100 Mbps) modes, the valid PBL range is 8 or less. ● 512 Bytes FIFO Depth: In the full-duplex mode and the half-duplex (10 or 100 Mbps) modes, the valid PBL range is 16 or less. ● 1 Kbytes FIFO Depth: In the full-duplex mode, the valid PBL range is 32 or less. In the half-duplex mode, the valid PBL range is 32 or less in the 10 or 100 Mbps mode and 16 or less in the 1000-Mbps mode. ● 2 Kbytes FIFO Depth: In the full-duplex mode and the half-duplex (10 or 100 Mbps) modes, the valid PBL range is 64 or less.

		• 4 Kbytes FIFO Depth: In the full-duplex mode and the half-duplex (10 or 100 Mbps) modes, the valid PBL range is 128 or less. • 8 Kbytes and Higher FIFO Depth: All PBL values are supported in the full-duplex and half-duplex modes.
[7]	**ATDS**	**Alternate Descriptor Size** When set, the size of the alternate descriptor increases to 32 bytes (8 DWORDS). This is required when the Advanced Timestamp feature or the IPC Full Offload Engine (Type 2) is enabled in the receiver. The enhanced descriptor is not required if the Advanced Timestamp and IPC Full Checksum Offload (Type 2) features are not enabled. In such cases, you can use the 16 bytes descriptor to save 4 bytes of memory. When reset, the descriptor size reverts back to 4 DWORDs (16 bytes). This bit preserves the backward compatibility for the descriptor size. In versions prior to 3.50a, the descriptor size is 16 bytes for both normal and enhanced descriptor. In version 3.50a, descriptor size is increased to 32 bytes because of the Advanced Timestamp and IPC Full Checksum Offload Engine (Type 2) features.
[6:2]	**DSL**	**Descriptor Skip Length** This bit specifies the number of Word, Dword, or Lword (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. The address skipping starts from the end of current descriptor to the start of next descriptor. When the DSL value is equal to zero, the descriptor table is taken as contiguous by the DMA in Ring mode.
[1]	**Reserved**	Reserved.
[0]	**SWR**	**Software Reset** When this bit is set, the MAC DMA Controller resets the logic and all internal registers of the MAC. It is cleared automatically after the reset operation has completed in all of the GMAC clock domains. Before reprogramming any register of the GMAC, you should read a zero (0) value in this bit . **Note:** • The Software reset function is driven only by this bit. • The reset operation is completed only when all resets in all active clock domains are de-asserted. Therefore, it is essential that all the PHY inputs clocks (applicable for the selected PHY interface) are present for the software reset completion.

GMAC Register 1001 Transmit Poll Demand Register (Transmit_Poll_Demand)

Register	Offset	R/W	Description	Reset Value
Transmit_Poll_Demand	GMACx_BA+0x1004	R/W	Register 1001 (Transmit Poll Demand Register)	0x00000000

31	30	29	28	27	26	25	24
TPD							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TPD							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TPD							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TPD							

Bits	Description	
[31:0]	**TPD**	**Transmit Poll Demand** When these bits are written with any value, the DMA reads the current descriptor pointed to by Register 1018 (Current Host Transmit Descriptor Register). If that descriptor is not available (owned by the Host), the transmission returns to the Suspend state and the Bit 2 (TU) of Register 1005 (Status Register) is asserted. If the descriptor is available, the transmission resumes.

GMAC Register 1002 Receive Poll Demand Register (Receive_Poll_Demand)

Register	Offset	R/W	Description	Reset Value
Receive_Poll_Demand	GMACx_BA+0x1008	R/W	Register 1002 (Receive Poll Demand Register)	0x00000000

31	30	29	28	27	26	25	24
RPD							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RPD							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RPD							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RPD							

Bits	Description	
[31:0]	**RPD**	**Receive Poll Demand** When these bits are written with any value, the DMA reads the current descriptor pointed to by Register 1019 (Current Host Receive Descriptor Register). If that descriptor is not available (owned by the Host), the reception returns to the Suspended state and the Bit 7 (RU) of Register 1005 (Status Register) is not asserted. If the descriptor is available, the Rx DMA returns to the active state.

GMAC Register 1003 Receive Descriptor List Address Register (Receive_Descriptor_List_Address)

Register	Offset	R/W	Description	Reset Value
Receive_Descriptor_List_Address	GMACx_BA+0x100c	R/W	Register 1003 (Receive Descriptor List Address Register)	0x00000000

31	30	29	28	27	26	25	24
RDESLA_64-bit							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RDESLA_64-bit							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RDESLA_64-bit							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RDESLA_64-bit					**Reserved**		

Bits	Description	
[31:3]	**RDESLA_64-bit**	**Start of Receive List** This field contains the base address of the first descriptor in the Receive Descriptor list. The LSB bits (2:0) for 64-bit bus width are ignored and are internally taken as all-zero by the DMA. Therefore, these LSB bits are read-only (RO).
[2:0]	**Reserved**	Reserved.

GMAC Register 1004 Transmit Descriptor List Address Register (Transmit_Descriptor_List_Address)

Register	Offset	R/W	Description	Reset Value
Transmit_Descriptor_List_Address	GMACx_BA+0x1010	R/W	Register 1004 (Transmit Descriptor List Address Register)	0x00000000

31	30	29	28	27	26	25	24
TDESLA_64-bit							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TDESLA_64-bit							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TDESLA_64-bit							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TDESLA_64-bit					**Reserved**		

Bits	Description	
[31:3]	**TDESLA_64-bit**	**Start of Transmit List** This field contains the base address of the first descriptor in the Transmit Descriptor list. The LSB bits (2:0) for 64-bit bus width are ignored and are internally taken as all-zero by the DMA. Therefore, these LSB bits are read-only (RO).
[2:0]	**Reserved**	Reserved.

GMAC Register 1005 Status Register (Status)

Register	Offset	R/W	Description	Reset Value
Status	GMACx_BA+0x1014	R/W	Register 1005 (Status Register)	0x00000000

31	30	29	28	27	26	25	24
Reserved	GLPII	TTI	GPI	Reserved	GLI	EB	
23	22	21	20	19	18	17	16
EB	TS			RS			NIS
15	14	13	12	11	10	9	8
AIS	ERI	FBI	Reserved		ETI	RWT	RPS
7	6	5	4	3	2	1	0
RU	RI	UNF	OVF	TJT	TU	TPS	TI

Bits	Description	
[31]	**Reserved**	Reserved.
[30]	**GLPII**	**GMAC LPI Interrupt (read only)** This bit indicates an interrupt event in the LPI logic of the GMAC. To reset this bit to 1'b0, the software must read the corresponding registers in the GMAC to get the exact cause of the interrupt and clear its source. **Note:** GLPII status is given only in Channel 0 DMA register and is applicable only when the Energy Efficient Ethernet feature is enabled. Otherwise, this bit is reserved. When this bit is high, the interrupt signal from the MAC (sbd_intr_o) is high.
[29]	**TTI**	**Timestamp Trigger Interrupt (read only)** This bit indicates an interrupt event in the Timestamp Generator block of GMAC. The software must read the corresponding registers in the GMAC to get the exact cause of interrupt and clear its source to reset this bit to 1'b0. When this bit is high, the interrupt signal from the GMAC subsystem (sbd_intr_o) is high. This bit is applicable only when the IEEE 1588 Timestamp feature is enabled. Otherwise, this bit is reserved.
[28]	**GPI**	**GMAC PMT Interrupt (read only)** This bit indicates an interrupt event in the PMT module of the GMAC. The software must read the PMT Control and Status Register in the MAC to get the exact cause of interrupt and clear its source to reset this bit to 1'b0. The interrupt signal from the GMAC subsystem (sbd_intr_o) is high when this bit is high. This bit is applicable only when the Power Management feature is enabled. Otherwise, this bit is reserved. **Note:** The GPI and pmt_intr_o interrupts are generated in different clock domains.
[27]	**Reserved**	Reserved.
[26]	**GLI**	**GMAC Line interface Interrupt (read only)** When set, this bit reflects any of the following interrupt events in the GMAC interfaces: • RGMII: Link change event To identify the exact cause of the interrupt, the software must first read Bit 11 and Bits[2:0] of Register 14 (Interrupt Status Register) and then to clear the source of interrupt (which also clears the GLI interrupt), read any of the following corresponding registers: • RGMII: Register 54 (RGMII Status Register) The interrupt signal from the GMAC subsystem (sbd_intr_o) is high when this bit is high.

[25:23]	**EB**	**Error Bits (read only)** This field indicates the type of error that caused a Bus Error, for example, error response on the AHB or AXI interface. This field is valid only when Bit 13 (FBI) is set. This field does not generate an interrupt. 000 = Error during Rx DMA Write Data Transfer 011 = Error during Tx DMA Read Data Transfer 100 = Error during Rx DMA Descriptor Write Access 101 = Error during Tx DMA Descriptor Write Access 110 = Error during Rx DMA Descriptor Read Access 111 = Error during Tx DMA Descriptor Read Access **Note:** 001 and 010 are reserved.
[22:20]	**TS**	**Transmit Process State (read only)** This field indicates the Transmit DMA FSM state. This field does not generate an interrupt. 3'b000 = Stopped; Reset or Stop Transmit Command issued 3'b001 = Running; Fetching Transmit Transfer Descriptor 3'b010 = Running; Waiting for status 3'b011 = Running; Reading Data from host memory buffer and queuing it to transmit buffer (Tx FIFO) 3'b100 = TIME_STAMP write state 3'b101 = Reserved for future use 3'b110 = Suspended; Transmit Descriptor Unavailable or Transmit Buffer Underflow 3'b111 = Running; Closing Transmit Descriptor
[19:17]	**RS**	**Received Process State (read only)** This field indicates the Receive DMA FSM state. This field does not generate an interrupt. 3'b000 = Stopped::Reset or Stop Receive Command issued 3'b001 = Running: Fetching Receive Transfer Descriptor 3'b010 = Reserved for future use 3'b011 = Running: Waiting for receive packet 3'b100 = Suspended: Receive Descriptor Unavailable 3'b101 = Running: Closing Receive Descriptor 3'b110 = TIME_STAMP write state 3'b111 =:Running: Transferring the receive packet data from receive buffer to host memory
[16]	**NIS**	**Normal Interrupt Summary** Normal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in Register 1007 (Interrupt Enable Register): Register 1005[0] = Transmit Interrupt Register 1005[2] = Transmit Buffer Unavailable Register 1005[6] = Receive Interrupt Register 1005[14] = Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in Register 1007) affect the Normal Interrupt Summary bit. This is a sticky bit and must be cleared (by writing 1 to this bit) each time a corresponding bit, which causes NIS to be set, is cleared.

[15]	**AIS**	**Abnormal Interrupt Summary** Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in Register 1007 (Interrupt Enable Register): Register 1005[1] = Transmit Process Stopped Register 1005[3] = Transmit Jabber Timeout Register 1005[4] = Receive FIFO Overflow Register 1005[5] = Transmit Underflow Register 1005[7] = Receive Buffer Unavailable Register 1005[8] = Receive Process Stopped Register 1005[9] = Receive Watchdog Timeout Register 1005[10] = Early Transmit Interrupt Register 1005[13] = Fatal Bus Error Only unmasked bits affect the Abnormal Interrupt Summary bit. This is a sticky bit and must be cleared each time a corresponding bit, which causes AIS to be set, is cleared.
[14]	**ERI**	**Early Receive Interrupt** This bit indicates that the DMA filled the first data buffer of the packet. This bit is cleared when the software writes 1 to this bit or Bit 6 (RI) of this register is set (whichever occurs earlier).
[13]	**FBI**	**Fatal Bus Error Interrupt** This bit indicates that a bus error occurred, as described in Bits[25:23]. When this bit is set, the corresponding DMA engine disables all of its bus accesses.
[12:11]	**Reserved**	Reserved.
[10]	**ETI**	**Early Transmit Interrupt** This bit indicates that the frame to be transmitted is fully transferred to the MTL Transmit FIFO.
[9]	**RWT**	**Receive Watchdog Timeout** When set, this bit indicates that the Receive Watchdog Timer expired while receiving the current frame and the current frame is truncated after the watchdog timeout.
[8]	**RPS**	**Receive Process Stopped** This bit is asserted when the Receive Process enters the Stopped state.
[7]	**RU**	**Receive Buffer Unavailable** This bit indicates that the host owns the Next Descriptor in the Receive List and the DMA cannot acquire it. The Receive Process is suspended. To resume processing Receive descriptors, the host should change the ownership of the descriptor and issue a Receive Poll Demand command. If no Receive Poll Demand is issued, the Receive Process resumes when the next recognized incoming frame is received. This bit is set only when the previous Receive Descriptor is owned by the DMA.
[6]	**RI**	**Receive Interrupt** This bit indicates that the frame reception is complete. When reception is complete, the Bit 31 of RDES1 (Disable Interrupt on Completion) is reset in the last Descriptor, and the specific frame status information is updated in the descriptor. The reception remains in the Running state.
[5]	**UNF**	**Transmit Underflow** This bit indicates that the Transmit Buffer had an Underflow during frame transmission. Transmission is suspended and an Underflow Error TDES0[1] is set.
[4]	**OVF**	**Receive Overflow** This bit indicates that the Receive Buffer had an Overflow during frame reception. If the partial frame is transferred to the application, the overflow status is set in RDES0[11].

[3]	**TJT**	**Transmit Jabber Timeout** This bit indicates that the Transmit Jabber Timer expired, which happens when the frame size exceeds 2,048 (10,240 bytes when the Jumbo frame is enabled). When the Jabber Timeout occurs, the transmission process is aborted and placed in the Stopped state. This causes the Transmit Jabber Timeout TDES0[14] flag to assert.
[2]	**TU**	**Transmit Buffer Unavailable** This bit indicates that the host owns the Next Descriptor in the Transmit List and the DMA cannot acquire it. Transmission is suspended. Bits[22:20] explain the Transmit Process state transitions. To resume processing Transmit descriptors, the host should change the ownership of the descriptor by setting TDES0[31] and then issue a Transmit Poll Demand command.
[1]	**TPS**	**Transmit Process Stopped** This bit is set when the transmission is stopped.
[0]	**TI**	**Transmit Interrupt** This bit indicates that the frame transmission is complete. When transmission is complete, Bit 31 (OWN) of TDES0 is reset, and the specific frame status information is updated in the descriptor.

GMAC Register 1006 Operation Mode Register (Operation_Mode)

Register	Offset	R/W	Description	Reset Value
Operation_Mode	GMACx_BA+0x1018	R/W	Register 1006 (Operation Mode Register)	0x00000000

31	30	29	28	27	26	25	24
Reserved					DT	RSF	DFF
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		TSF	FTF	Reserved			TTC
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TTC		ST	RFD		RFA		EFC
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FEF	FUF	DGF	RTC		OSF	SR	Reserved

Bits	Description	
[31:27]	**Reserved**	Reserved.
[26]	**DT**	**Disable Dropping of TCP/IP Checksum Error Frames** When this bit is set, the MAC does not drop the frames which only have errors detected by the Receive Checksum Offload engine. Such frames do not have any errors (including FCS error) in the Ethernet frame received by the MAC but have errors only in the encapsulated payload. When this bit is reset, all error frames are dropped if the FEF bit is reset. If the IPC Full Checksum Offload Engine (Type 2) is disabled, this bit is reserved (RO with value 1'b0).
[25]	**RSF**	**Receive Store and Forward** When this bit is set, the MTL reads a frame from the Rx FIFO only after the complete frame has been written to it, ignoring the RTC bits. When this bit is reset, the Rx FIFO operates in the cut-through mode, subject to the threshold specified by the RTC bits
[24]	**DFF**	**Disable Flushing of Received Frames** When this bit is set, the Rx DMA does not flush any frames because of the unavailability of receive descriptors or buffers as it does normally when this bit is reset.
[23:22]	**Reserved**	Reserved.
[21]	**TSF**	**Transmit Store and Forward** When this bit is set, transmission starts when a full frame resides in the MTL Transmit FIFO. When this bit is set, the TTC values specified in Bits[16: 14] are ignored. This bit should be changed only when the transmission is stopped.
[20]	**FTF**	**Flush Transmit FIFO** When this bit is set, the transmit FIFO controller logic is reset to its default values and thus all data in the Tx FIFO is lost or flushed. This bit is cleared internally when the flushing operation is completed. The Operation Mode register should not be written to until this bit is cleared. The data which is already accepted by the MAC transmitter is not flushed. It is scheduled for transmission and results in underflow and runt frame transmission. **Note:** The flush operation is complete only when the Tx FIFO is emptied of its contents and all the pending Transmit Status of the transmitted frames are accepted by the host. To complete this flush operation, the PHY transmit clock (clk_tx_i) is required to be active.
[19:17]	**Reserved**	Reserved.

[16:14]	**TTC**	**Transmit Threshold Control** These bits control the threshold level of the MTL Transmit FIFO. Transmission starts when the frame size within the MTL Transmit FIFO is larger than the threshold. In addition, full frames with a length less than the threshold are also transmitted. These bits are used only when Bit 21 (TSF) is reset. 000 = 64 bytes 001 = 128 bytes 010 = 192 bytes 011 = 256 bytes 100 = 40 bytes 101 = 32 bytes 110 = 24 bytes 111 = 16 bytes
[13]	**ST**	**Start or Stop Transmission Command** When this bit is set, transmission is placed in the Running state, and the DMA checks the Transmit List at the current position for a frame to be transmitted. Descriptor acquisition is attempted either from the current position in the list, which is the Transmit List Base Address set by Register 1004 (Transmit Descriptor List Address Register), or from the position retained when transmission was stopped previously. If the DMA does not own the current descriptor, transmission enters the Suspended state and Bit 2 (Transmit Buffer Unavailable) of Register 1005 (Status Register) is set. The Start Transmission command is effective only when transmission is stopped. If the command is issued before setting Register 1004 (Transmit Descriptor List Address Register), then the DMA behavior is unpredictable. When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current frame. The Next Descriptor position in the Transmit List is saved, and it becomes the current position when transmission is restarted. To change the list address, you need to program Register 1004 (Transmit Descriptor List Address Register) with a new value when this bit is reset. The new value is considered when this bit is set again. The stop transmission command is effective only when the transmission of the current frame is complete or the transmission is in the Suspended state.
[12:11]	**RFD**	**Threshold for Deactivating Flow Control** These bits control the threshold (Fill-level of Rx FIFO) at which the flow control is de-asserted after activation. 00 = Full minus 1 Kbytes, that is, FULL - 1 Kbytes 01 = Full minus 2 Kbytes, that is, FULL - 2 Kbytes 10 = Full minus 3 Kbytes, that is, FULL - 3 Kbytes 11 = Full minus 4 Kbytes, that is, FULL - 4 Kbytes The de-assertion is effective only after flow control is asserted. If the Rx FIFO is 8 Kbytes or more, an additional bit (RFD_2) is used for more threshold levels as described in Bit 22. These bits are reserved and read-only when the Rx FIFO depth is less than 4 Kbytes. **Note =** For proper flow control, the value programmed in the "RFD_2, RFD" fields should be equal to or more than the value programmed in the "RFA_2, RFA" fields.
[10:9]	**RFA**	**Threshold for Activating Flow Control** These bits control the threshold (Fill level of Rx FIFO) at which the flow control is activated. 00 = Full minus 1 Kbytes, that is, FULL - 1 Kbytes 01 = Full minus 2 Kbytes, that is, FULL - 2 Kbytes 10 = Full minus 3 Kbytes, that is, FULL - 3 Kbytes 11 = Full minus 4 Kbytes, that is, FULL - 4 Kbytes These values are applicable only to Rx FIFOs of 4 Kbytes or more and when Bit 8 (EFC) is set high. If the Rx FIFO is 8 Kbytes or more, an additional Bit (RFA_2) is used for more threshold levels as described in Bit 23. These bits are reserved and read-only when the depth of Rx FIFO is less than 4 Kbytes. **Note:** When FIFO size is exactly 4 Kbytes, although the GMAC allows you to program the value of these bits to 11, the software should not program these bits to 2'b11. The value 2'b11 means flow control on FIFO empty condition.
[8]	**EFC**	**Enable HW Flow Control** When this bit is set, the flow control signal operation based on the fill-level of Rx FIFO is enabled. When reset, the flow control operation is disabled. This bit is not used (reserved and always reset) when the Rx FIFO is less than 4 Kbytes.

[7]	**FEF**	**Forward Error Frames** When this bit is reset, the Rx FIFO drops frames with error status (CRC error, collision error, GMII_ER, giant frame, watchdog timeout, or overflow). However, if the start byte (write) pointer of a frame is already transferred to the read controller side (in Threshold mode), then the frame is not dropped. When the FEF bit is set, all frames except runt error frames are forwarded to the DMA. If the Bit 25 (RSF) is set and the Rx FIFO overflows when a partial frame is written, then the frame is dropped irrespective of the FEF bit setting. However, if the Bit 25 (RSF) is reset and the Rx FIFO overflows when a partial frame is written, then a partial frame may be forwarded to the DMA.
[6]	**FUF**	**Forward Undersized Good Frames** When set, the Rx FIFO forwards Undersized frames (frames with no Error and length less than 64 bytes) including pad-bytes and CRC. When reset, the Rx FIFO drops all frames of less than 64 bytes, unless a frame is already transferred because of the lower value of Receive Threshold, for example, RTC = 01.
[5]	**DGF**	**Drop Giant Frames** When set, the MAC drops the received giant frames in the Rx FIFO, that is, frames that are larger than the computed giant frame limit. When reset, the MAC does not drop the giant frames in the Rx FIFO.
[4:3]	**RTC**	**Receive Threshold Control** These two bits control the threshold level of the MTL Receive FIFO. Transfer (request) to DMA starts when the frame size within the MTL Receive FIFO is larger than the threshold. In addition, full frames with length less than the threshold are transferred automatically. These bits are valid only when the RSF bit is zero, and are ignored when the RSF bit is set to 1. 00 = 64 bytes 01 = 32 bytes 10 = 96 bytes 11 = 128 bytes
[2]	**OSF**	**Operate on Second Frame** When this bit is set, it instructs the DMA to process the second frame of the Transmit data even before the status for the first frame is obtained.
[1]	**SR**	**Start or Stop Receive** When this bit is set, the Receive process is placed in the Running state. The DMA attempts to acquire the descriptor from the Receive list and processes the incoming frames. The descriptor acquisition is attempted from the current position in the list, which is the address set by Register 1003 (Receive Descriptor List Address Register) or the position retained when the Receive process was previously stopped. If the DMA does not own the descriptor, reception is suspended and Bit 7 (Receive Buffer Unavailable) of Register 1005 (Status Register) is set. The Start Receive command is effective only when the reception has stopped. If the command is issued before setting Register 1003 (Receive Descriptor List Address Register), the DMA behavior is unpredictable. When this bit is cleared, the Rx DMA operation is stopped after the transfer of the current frame. The next descriptor position in the Receive list is saved and becomes the current position after the Receive process is restarted. The Stop Receive command is effective only when the Receive process is in either the Running (waiting for receive packet) or in the Suspended state.
[0]	**Reserved**	Reserved.

GMAC Register 1007 Interrupt Enable Register (Interrupt_Enable)

Register	Offset	R/W	Description	Reset Value
Interrupt_Enable	GMACx_BA+0x101c	R/W	Register 1007 (Interrupt Enable Register)	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							NIE
15	14	13	12	11	10	9	8
AIE	ERE	FBE	Reserved		ETE	RWE	RSE
7	6	5	4	3	2	1	0
RUE	RIE	UNE	OVE	TJE	TUE	TSE	TIE

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**NIE**	**Normal Interrupt Summary Enable** When this bit is set, normal interrupt summary is enabled. When this bit is reset, normal interrupt summary is disabled. This bit enables the following interrupts in Register 1005 (Status Register): Register 1005[0] = Transmit Interrupt Register 1005[2] = Transmit Buffer Unavailable Register 1005[6] = Receive Interrupt Register 1005[14] = Early Receive Interrupt
[15]	**AIE**	**Abnormal Interrupt Summary Enable** When this bit is set, abnormal interrupt summary is enabled. When this bit is reset, the abnormal interrupt summary is disabled. This bit enables the following interrupts in Register 1005 (Status Register): Register 1005[1] = Transmit Process Stopped Register 1005[3] = Transmit Jabber Timeout Register 1005[4] = Receive Overflow Register 1005[5] = Transmit Underflow Register 1005[7] = Receive Buffer Unavailable Register 1005[8] = Receive Process Stopped Register 1005[9] = Receive Watchdog Timeout Register 1005[10] = Early Transmit Interrupt Register 1005[13] = Fatal Bus Error
[14]	**ERE**	**Early Receive Interrupt Enable** When this bit is set with Normal Interrupt Summary Enable (Bit 16), the Early Receive Interrupt is enabled. When this bit is reset, the Early Receive Interrupt is disabled.
[13]	**FBE**	**Fatal Bus Error Enable** When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Fatal Bus Error Interrupt is enabled. When this bit is reset, the Fatal Bus Error Enable Interrupt is disabled.
[12:11]	**Reserved**	Reserved.

[10]	**ETE**	**Early Transmit Interrupt Enable** When this bit is set with an Abnormal Interrupt Summary Enable (Bit 15), the Early Transmit Interrupt is enabled. When this bit is reset, the Early Transmit Interrupt is disabled.
[9]	**RWE**	**Receive Watchdog Timeout Enable** When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Receive Watchdog Timeout Interrupt is enabled. When this bit is reset, the Receive Watchdog Timeout Interrupt is disabled.
[8]	**RSE**	**Receive Stopped Enable** When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Receive Stopped Interrupt is enabled. When this bit is reset, the Receive Stopped Interrupt is disabled.
[7]	**RUE**	**Receive Buffer Unavailable Enable** When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Receive Buffer Unavailable Interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable Interrupt is disabled.
[6]	**RIE**	**Receive Interrupt Enable** When this bit is set with Normal Interrupt Summary Enable (Bit 16), the Receive Interrupt is enabled. When this bit is reset, the Receive Interrupt is disabled.
[5]	**UNE**	**Underflow Interrupt Enable** When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Transmit Underflow Interrupt is enabled. When this bit is reset, the Underflow Interrupt is disabled.
[4]	**OVE**	**Overflow Interrupt Enable** When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Receive Overflow Interrupt is enabled. When this bit is reset, the Overflow Interrupt is disabled.
[3]	**TJE**	**Transmit Jabber Timeout Enable** When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Transmit Jabber Timeout Interrupt is enabled. When this bit is reset, the Transmit Jabber Timeout Interrupt is disabled.
[2]	**TUE**	**Transmit Buffer Unavailable Enable** When this bit is set with Normal Interrupt Summary Enable (Bit 16), the Transmit Buffer Unavailable Interrupt is enabled. When this bit is reset, the Transmit Buffer Unavailable Interrupt is disabled.
[1]	**TSE**	**Transmit Stopped Enable** When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Transmission Stopped Interrupt is enabled. When this bit is reset, the Transmission Stopped Interrupt is disabled.
[0]	**TIE**	**Transmit Interrupt Enable** When this bit is set with Normal Interrupt Summary Enable (Bit 16), the Transmit Interrupt is enabled. When this bit is reset, the Transmit Interrupt is disabled.

GMAC Register 1008 Missed Frame and Buffer Overflow Counter Register (Missed_Frame_And_Buffer_Overflow_Counter)

Register	Offset	R/W	Description	Reset Value
Missed_Frame_And_Buffer_Overflow_Counter	GMACx_BA+0x1020	R	Register 1008 (Missed Frame and Buffer Overflow Counter Register)	0x00000000

31	30	29	28	27	26	25	24
Reserved			OVFCNTOVF	OVFFRMCNT			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
OVFFRMCNT							MISCNTOVF
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
MISFRMCNT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MISFRMCNT							

Bits	Description	
[31: 29]	**Reserved**	Reserved.
[28]	**OVFCNTOVF**	**Overflow Bit for FIFO Overflow Counter** This bit is set every time the Overflow Frame Counter (Bits[27: 17]) overflows, that is, the Rx FIFO overflows with the overflow frame counter at maximum value. In such a scenario, the overflow frame counter is reset to all-zeros and this bit indicates that the rollover happened.
[27: 17]	**OVFFRMCNT**	**Overflow Frame Counter** This field indicates the number of frames missed by the application. This counter is incremented each time the MTL FIFO overflows. The counter is cleared when this register is read with mci_be_i[2] at 1'b1.
[16]	**MISCNTOVF**	**Overflow Bit for Missed Frame Counter** This bit is set every time Missed Frame Counter (Bits[15:0]) overflows, that is, the DMA discards an incoming frame because of the Host Receive Buffer being unavailable with the missed frame counter at maximum value. In such a scenario, the Missed frame counter is reset to all-zeros and this bit indicates that the rollover happened.
[15:0]	**MISFRMCNT**	**Missed Frame Counter** This field indicates the number of frames missed by the controller because of the Host Receive Buffer being unavailable. This counter is incremented each time the DMA discards an incoming frame. The counter is cleared when this register is read with mci_be_i[0] at 1'b1.

GMAC Register 1009 Receive Interrupt Watchdog Timer Register (Receive_Interrupt_Watchdog_Timer)

Register	Offset	R/W	Description	Reset Value
Receive_Interrupt_Watchdog_Timer	GMACx_BA+0x1024	R/W	Register 1009 (Receive Interrupt Watchdog Timer Register)	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RIWT							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**RIWT**	**RI Watchdog Timer Count** This bit indicates the number of system clock cycles multiplied by 256 for which the watchdog timer is set. The watchdog timer gets triggered with the programmed value after the Rx DMA completes the transfer of a frame for which the RI status bit is not set because of the setting in the corresponding descriptor RDES1[31]. When the watchdog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when the RI bit is set high because of automatic setting of RI as per RDES1[31] of any received frame.

GMAC Register 1010 AXI Bus Mode Register (AXI_Bus_Mode)

Register	Offset	R/W	Description	Reset Value
AXI_Bus_Mode	GMACx_BA+0x1028	R/W	Register 1010 (AXI Bus Mode Register)	0x00110001

31	30	29	28	27	26	25	24
EN_LPI	LPI_XIT_FRM	Reserved					
23	22	21	20	19	18	17	16
WR_OSR_LMT_GT8	WR_OSR_LMT_GT4	WR_OSR_LMT		RD_OSR_LMT_GT8	RD_OSR_LMT_GT4	RD_OSR_LMT	
15	14	13	12	11	10	9	8
Reserved		ONEKBBE	AXI_AAL	Reserved			
7	6	5	4	3	2	1	0
Reserved				BLEN16	BLEN8	BLEN4	UNDEF

Bits	Description	
[31]	**EN_LPI**	**Enable Low Power Interface** When set to 1, this bit enables the Low Power Interface (LPI) mode and accepts the LPI request from the AXI System Clock controller. When set to 0, this bit disables the LPI mode and always denies the LPI request from the AXI System Clock controller.
[30]	**LPI_XIT_FRM**	**Unlock on Magic Packet or Remote Wake Up** When set to 1, this bit enables the GMAC-AXI to come out of the LPI mode only when the Magic Packet or Remote Wake Up Packet is received. When set to 0, this bit enables the GMAC-AXI to come out of LPI mode when any frame is received.
[29:24]	**Reserved**	Reserved.
[23]	**WR_OSR_LMT_GT8**	Reserved.
[22]	**WR_OSR_LMT_GT4**	Reserved.
[21:20]	**WR_OSR_LMT**	AXI Maximum Write OutStanding Request Limit
[19]	**RD_OSR_LMT_GT8**	Reserved.
[18]	**RD_OSR_LMT_GT4**	Reserved.
[17:16]	**RD_OSR_LMT**	**AXI Maximum Read OutStanding Request Limit** This value limits the maximum outstanding request on the AXI read interface. Maximum outstanding requests = RD_OSR_LMT+1. **Note:** The Bit 18 is reserved if AXI_GM_MAX_RD_REQUESTS = 4. The Bit 19 is reserved if AXI_GM_MAX_RD_REQUESTS !.= 16.
[15:14]	**Reserved**	Reserved.
[13]	**ONEKBBE**	**1 Kbytes Boundary Crossing Enable for the GMAC-AXI Master When set, the GMAC-AXI Master performs burst transfers that do not cross 1 Kbytes boundary** When reset, the GMAC-AXI Master performs burst transfers that do not cross 4 Kbytes boundary.

[12]	**AXI_AAL**	**Address-Aligned Beats (read only)** This bit is read-only bit and reflects the Bit 25 (AAL) of Register 1000 (Bus Mode Register). When this bit is set to 1, the GMAC-AXI performs address-aligned burst transfers on both read and write channels.
[11:4]	**Reserved**	Reserved.
[3]	**BLEN16**	**AXI Burst Length 16** When this bit is set to 1 or UNDEF is set to 1, the GMAC-AXI is allowed to select a burst length of 16 on the AXI Master interface.
[2]	**BLEN8**	**AXI Burst Length 8** When this bit is set to 1, the GMAC-AXI is allowed to select a burst length of 8 on the AXI Master interface. Setting this bit has no effect when UNDEF is set to 1.
[1]	**BLEN4**	**AXI Burst Length 4** When this bit is set to 1, the GMAC-AXI is allowed to select a burst length of 4 on the AXI Master interface. Setting this bit has no effect when UNDEF is set to 1.
[0]	**UNDEF**	**AXI Undefined Burst Length (read only)** This bit is read-only bit and indicates the complement (invert) value of Bit 16 (FB) in Register 1000 (Bus Mode Register[16]). ● When this bit is set to 1, the GMAC-AXI is allowed to perform any burst length equal to or below the maximum allowed burst length programmed in Bits[7:1]. ● When this bit is set to 0, the GMAC-AXI is allowed to perform only fixed burst lengths as indicated by BLEN256, BLEN128, BLEN64, BLEN32, BLEN16, BLEN8, or BLEN4, or a burst length of 1.

GMAC Register 1011 AHB or AXI Status Register (AHB_or_AXI_Status)

Register	Offset	R/W	Description	Reset Value
AHB_or_AXI_Status	GMACx_BA+0x102c	R	Register 1011 (AHB or AXI Status Register)	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						AXIRDSTS	AXWHSTS

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**AXIRDSTS**	**AXI Master Read Channel Status** When high, it indicates that AXI Master's read channel is active and transferring data.
[0]	**AXWHSTS**	**AXI Master Write Channel or AHB Master Status** When high, it indicates that AXI Master's write channel is active and transferring data.

GMAC Register 1018 Current Host Transmit Descriptor Register (Current_Host_Transmit_Descriptor)

Register	Offset	R/W	Description	Reset Value
Current_Host_Transmit_Descriptor	GMACx_BA+0x1048	R	Register 1018 (Current Host Transmit Descriptor Register)	0x00000000

31	30	29	28	27	26	25	24
CURTDESAPTR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CURTDESAPTR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CURTDESAPTR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CURTDESAPTR							

Bits	Description	
[31:0]	**CURTDESAPTR**	**Host Transmit Descriptor Address Pointer** Cleared on Reset. Pointer updated by the DMA during operation.

GMAC Register 1019 Current Host Receive Descriptor Register (Current_Host_Receive_Descriptor)

Register	Offset	R/W	Description	Reset Value
Current_Host_Receive_Descriptor	GMACx_BA+0x104c	R	Register 1019 (Current Host Receive Descriptor Register)	0x00000000

31	30	29	28	27	26	25	24
CURRDESAPTR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CURRDESAPTR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CURRDESAPTR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CURRDESAPTR							

Bits	Description	
[31:0]	**CURRDESAPTR**	**Host Receive Descriptor Address Pointer** Cleared on Reset. Pointer updated by the DMA during operation.

GMAC Register 1020 Current Host Transmit Buffer Address Register (Current_Host_Transmit_Buffer_Address)

Register	Offset	R/W	Description	Reset Value
Current_Host_Transmit_Buffer_Address	GMACx_BA+0x1050	R	Register 1020 (Current Host Transmit Buffer Address Register)	0x00000000

31	30	29	28	27	26	25	24
CURTBUFAPTR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CURTBUFAPTR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CURTBUFAPTR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CURTBUFAPTR							

Bits	Description	
[31:0]	**CURTBUFAPTR**	**Host Transmit Buffer Address Pointer** Cleared on Reset. Pointer updated by the DMA during operation.

GMAC Register 1021 Current Host Receive Buffer Address Register (Current_Host_Receive_Buffer_Address)

Register	Offset	R/W	Description	Reset Value
Current_Host_Receive_Buffer_Address	GMACx_BA+0x1054	R	Register 1021 (Current Host Receive Buffer Address Register)	0x00000000

31	30	29	28	27	26	25	24
CURRBUFAPTR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CURRBUFAPTR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CURRBUFAPTR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CURRBUFAPTR							

Bits	Description	
[31:0]	**CURRBUFAPTR**	**Host Receive Buffer Address Pointer** Cleared on Reset. Pointer updated by the DMA during operation.

GMAC Register 1022 HW Feature Register (HW_Feature)

Register	Offset	R/W	Description	Reset Value
HW_Feature	GMACx_BA+0x1058	R	Register 1022 (HW Feature Register)	0x1F0D6527

31	30	29	28	27	26	25	24
Reserved	ACTPHYIF			SAVLANINS	FLEXIPPSEN	INTTSEN	ENHDESSEL
23	22	21	20	19	18	17	16
TXCHCNT		RXCHCNT		RXFIFOSIZE	RXTYP2COE	RXTYP1COE	TXCOESEL
15	14	13	12	11	10	9	8
AVSEL	EEESEL	TSVER2SEL	TSVER1SEL	MMCSEL	MGKSEL	RWKSEL	SMASEL
7	6	5	4	3	2	1	0
L3L4FLTREN	PCSSEL	ADDMACADR SEL	HASHSEL	EXTHASHEN	HDSEL	GMIISEL	MIISEL

Bits	Description	
[31]	**Reserved**	Reserved.
[30:28]	**ACTPHYIF**	**Active or Selected PHY interface** When you have multiple PHY interfaces in your configuration, this field indicates the sampled value of phy_intf_sel_i during reset de-assertion 0001 = RGMII 0100 = RMII All Others = Reserved
[27]	**SAVLANINS**	Source Address or VLAN Insertion
[26]	**FLEXIPPSEN**	Flexible Pulse-Per-Second Output
[25]	**INTTSEN**	Timestamping with Internal System Time
[24]	**ENHDESSEL**	Alternate (Enhanced Descriptor)
[23:22]	**TXCHCNT**	Number of additional Tx channels
[21:20]	**RXCHCNT**	Number of additional Rx channels
[19]	**RXFIFOSIZE**	Rx FIFO > 2,048 Bytes
[18]	**RXTYP2COE**	IP Checksum Offload (Type 2) in Rx
[17]	**RXTYP1COE**	IP Checksum Offload (Type 1) in Rx
[16]	**TXCOESEL**	Checksum Offload in Tx
[15]	**AVSEL**	AV Feature
[14]	**EEESEL**	Energy Efficient Ethernet
[13]	**TSVER2SEL**	IEEE 1588-2008 Advanced Timestamp
[12]	**TSVER1SEL**	Only IEEE 1588-2002 Timestamp
[11]	**MMCSEL**	RMON Module

[10]	**MGKSEL**	PMT Magic Packet
[9]	**RWKSEL**	PMT Remote Wakeup
[8]	**SMASEL**	SMA (MDIO) Interface
[7]	**L3L4FLTREN**	Layer 3 and Layer 4 Filter Feature
[6]	**PCSSEL**	PCS registers (TBI, SGMII, or RTBI PHY interface)
[5]	**ADDMACADRSEL**	Multiple MAC Address Registers
[4]	**HASHSEL**	HASH Filter
[3]	**EXTHASHEN**	Expanded DA Hash Filter
[2]	**HDSEL**	Half-Duplex support
[1]	**GMIISEL**	1000 Mbps support
[0]	**MIISEL**	10 or 100 Mbps support

6.35 Secure Digital Host Controller (SDH)

6.35.1 Overview

The Secure Digital Host Controller (SD Host) has DMA engine, host controller registers, FIFO controller and SD/UHS-I/eMMC unit. The DMA engine provides a DMA (Direct Memory Access) function for SD to exchange data between system memory and shared buffer, and DMA options such as SDMA(Single operation DMA), ADMA2 (Advanced DMA) as specified in the SD host controller standard. The SD host controller can support SD or eMMC specification and cooperate with the DMA engine to provide a fast data transfer between system memory and cards.

6.35.2 Features

- AMBA AHB master/slave interface compatible, for data transfer and register read/write.
- Supports data transfer types such as CPU, SDMA, ADMA2 for SD, eMMC mode.
- Supports 1-bit and 4-bit data bus widths for the SD memory card specification version 3.0. (SDR104 speed limited to maximum allowed I/O speed. SPI mode, DDR50 and UHS-II mode not supported).
- Supports 1-bit, 4-bit and 8-bit data bus widths for the eMMC interface. (HS200 speed limited to maximum allowed I/O speed and HS400 is not supported).
- Supports SD/SDHC/SDXC/SDIO, eMMC card.
- Supports gating of controller base clock if host controller is inactive.
- Supports two set of SD host controllers, only one can support UHS-I mode.
- Supports SD/eMMC tuning, CMD19(SD) or CMD21(eMMC).
- Supports 50 MHz to achieve SD 25 Mbyte/s for 4-bit mode.
- Supports 200 MHz to achieve eMMC HS200 at 1.8V I/O operation.

6.35.3 Block Diagram

The block diagram and Card Pad Assignment of SD Host Controller is shown as follows.

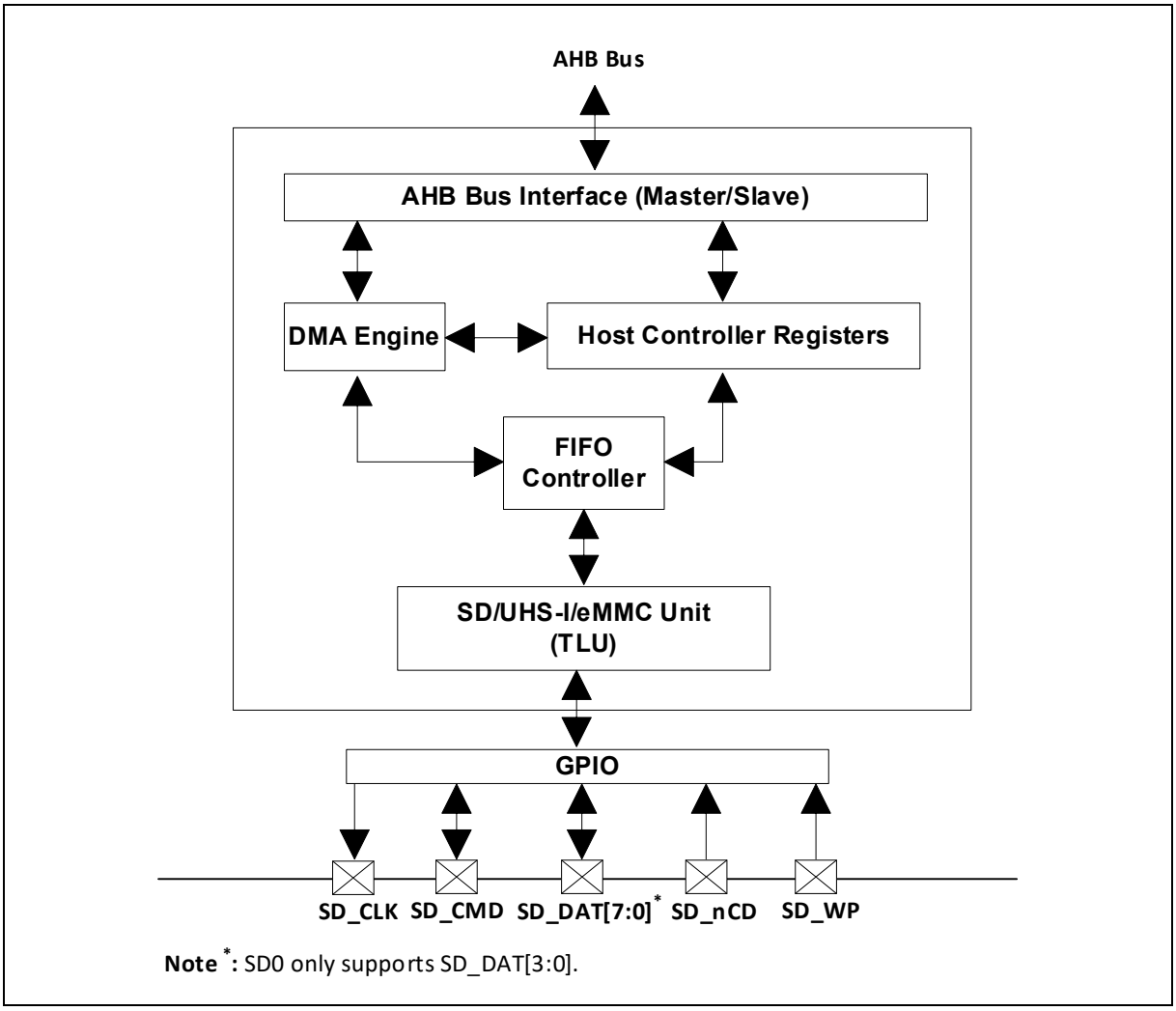


Figure 6.35-1 SD Host Controller Block Diagram

6.35.4 Basic Configuration

6.35.4.1 SD0 Basic Configuration

- Clock Source Configuration
  - Select the source of SD0 engine clock on SD0SEL (CLK_CLKSEL0[17:16]).
  - Select the clock divider number of SD0 engine clock on FREQ_SEL (CLK_CTRL_R[15:8]).
  - Enable SD0 engine clock in SDH0EN (CLK_SYSCLK0[16]).
- Reset Configuration
  - Reset SD0 controller in SDH0RST (SYS_IPRST0[24]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
SD0	SD0_CLK / eMMC0_CLK	PC.1	MFP6
	SD0_CMD / eMMC0_CMD	PC.0	MFP6

	SD0_DAT0 / eMMC0_DAT0	PC.2	MFP6
	SD0_DAT1 / eMMC0_DAT1	PC.3	MFP6
	SD0_DAT2 / eMMC0_DAT2	PC.4	MFP6
	SD0_DAT3 / eMMC0_DAT3	PC.5	MFP6
	SD0_nCD	PC.6	MFP6
	SD0_WP	PC.7	MFP6

Table 6.35-1 SD0 Pin Configuration

6.35.4.2 SD1 Basic Configuration

- Clock Source Configuration
  - Select the source of SD1 engine clock on SD1SEL (CLK_CLKSEL0[19:18]).
  - Select the clock divider number of SD1 engine clock on FREQ_SEL (CLK_CTRL_R[15:8]).
  - Enable SD1 engine clock in SDH1EN (CLK_SYSCLK0[17]).
- Reset Configuration
  - Reset SD1 controller in SDH1RST (SYS_IPRST0[25]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
SD1	SD1_CLK / eMMC1_CLK	PJ.7	MFP6
	SD1_CMD / eMMC1_CMD	PJ.6	MFP6
	SD1_DAT0 / eMMC1_DAT0	PJ.8	MFP6
	SD1_DAT1 / eMMC1_DAT1	PJ.9	MFP6
	SD1_DAT2 / eMMC1_DAT2	PJ.10	MFP6
	SD1_DAT3 / eMMC1_DAT3	PJ.11	MFP6
	SD1_nCD	PJ.5	MFP6
	SD1_WP	PJ.4	MFP6
	eMMC1_DAT4	PJ.0	MFP6
	eMMC1_DAT5	PJ.1	MFP6
	eMMC1_DAT6	PJ.2	MFP6
	eMMC1_DAT7	PJ.3	MFP6

Table 6.35-2 SD1 Pin Configuration

6.35.5 Functional Description

The SD host provides an interface for SD/SDHC/SDXC/SDIO card access with 1-bit/4-bit data bus width and 1-bit/4-bit/8-bit data bus width for eMMC card.

6.35.5.1 Card Detection

User can enable interrupt for card detect, set CARD_INSERTION_STAT_EN (NORMAL_INT_STAT_EN_R[6]), CARD_REMOVAL_STAT_EN (NORMAL_INT_STAT_EN_R[7]),

CARD_INSERTION_SIGNAL_EN (NORMAL_INT_SIGNAL_EN_R[6]) and CARD_REMOVAL_SIGNAL_EN (NORMAL_INT_SIGNAL_EN_R[7]) to 1. After card insertion or removal interrupt is generated, write one for CARD_INSERTION_STAT_EN (NORMAL_INT_STAT_EN_R[6]) and CARD_REMOVAL_STAT_EN (NORMAL_INT_STAT_EN_R[7]). Then check CARD_INSERTED (PSTATE_REG[16]) if card is inserted.

6.35.5.2 Basic Operation

This SD controller can generate all types of command to the SD card and retrieve all types of response from SD card. User needs to wait for command complete interrupt, CMD_COMPLETE (NORMAL_INT_STAT_R[0]), and set CMD_COMPLETE (NORMAL_INT_STAT_R[0]) again to clear command complete status after command complete. After command execution is completed, XFER_COMPLETE (NORMAL_INT_STAT_R[1]) will be set. After response in, the content of response will be stored at RESP01_R, RESP23_R, RESP45_R and RESP67_R. If any error happened, the error can be find in ERROR_INT_STAT_R, ERROR_INT_STAT_EN_R and ERROR_INT_SIGNAL, software should notice that after response in.

If issuing command with data transfer, wait for buffer write or read ready interrupt, BUF_WR_READY (NORMAL_INT_STAT_R[4]) or BUF_RD_READY (NORMAL_INT_STAT_R[5]) after command transfer completed. When BUF_WR_READY or BUF_RD_READY set to 1, means buffer write or read ready interrupt occurs. Write BUF_WR_READY or BUF_RD_READY one to clear its status after interrupt.

For data transfer, set XFER_BLOCK_SIZE (BLOCKSIZE_R[11:0]) to determine transfer block size, set BLOCK_CNT (BLOCKCOUNT_R[15:0]) to determine transfer block count and set BLOCKCNT_SDMASA (SDMASA_R[31:0]) for SDMA system address. Wait for transfer complete interrupt, XFER_COMPLETE (NORMAL_INT_STAT_R[1]) set to 1 and data transfer done.

6.35.5.3 Data Transfer (Using ADMA2)

Figure 6.35-2 shows the ADMA2 data transfer between the host controller and the system memory and it shows the descriptor table that is created by the host driver. In the descriptor table, each line contains information about the address, its length and an attributes that specifies the operation of the descriptor line. An ADMA2 transfer is triggered by writing to the command register and ADMA2 processes each line in the descriptor table.

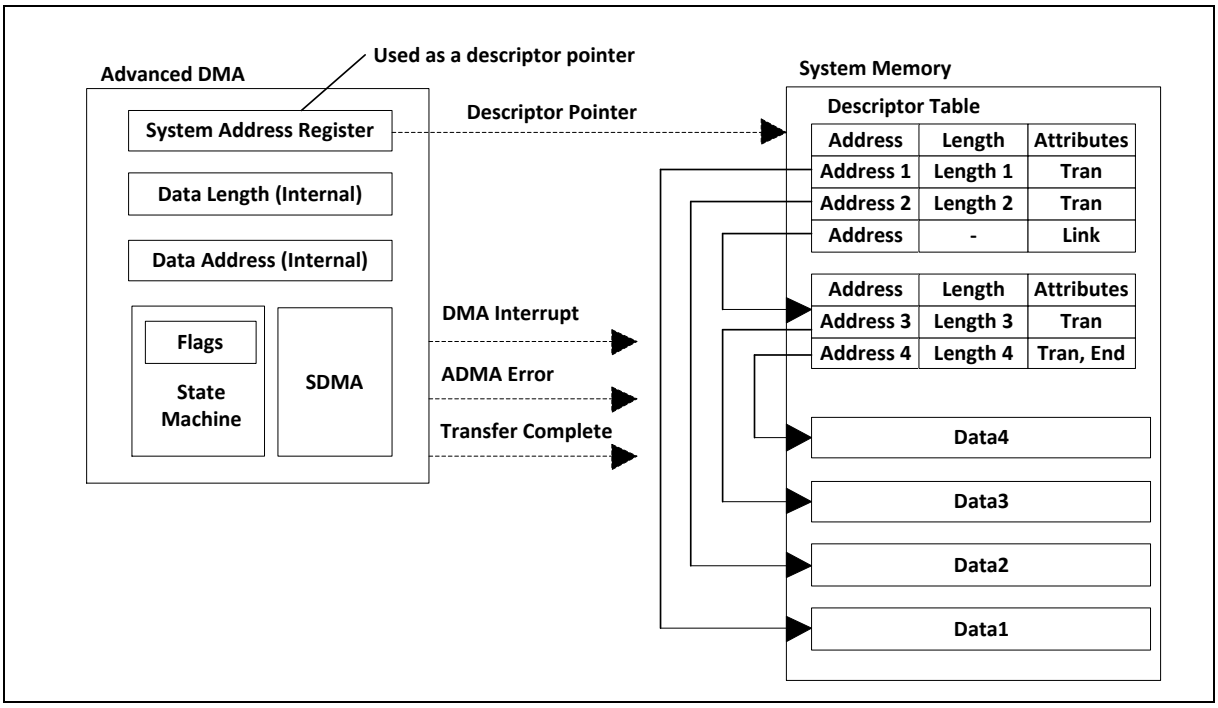


Figure 6.35-2 ADMA2 Block Diagram

Set DMA_SEL (HOST_CTRL1_R[4:3]) to select the DMA type and create descriptor table for ADMA in the system memory. Then set descriptor address in the ADMA_SA_LOW_R register. Setting BLOCKSIZE_R, BLOCKCOUNT_R and argument value to ARGUMENT_R. Setting Single or multiple block transfer bit, block count enable bit, DMA enable bit and determines data transfer direction in XFER_MODE_R register. Issuing command with data transfer after above related setting, then wait for transfer complete interrupt and check ADMA_ERR (ERROR_INT_STAT_R[9]) if ADMA error interrupt occurs.

It is recommended that set 'End' bit in the last transfer descriptor. End bit should always be set on the descriptor of type 'TRAN'. Figure 6.35-3 and Table 6.35-3 shows the ADMA2 descriptor format.

64-bit ADMA Descriptor Line (32-bit Addressing Mode)

32-bit Address		16-bit Length	10-bit Length	Attribute					
63 32		31 16	15 06	05	04	03	02	01	00
xxxx_xxxh		xxxh	xxxxxxxxxxb	Act2	Act1	Act0	Int	End	Valid
		Lower	Upper						

Valid	Indicates Validity of a Descriptor Line
End	End of Descriptor
Int	Force to generate ADMA Interrupt

Figure 6.35-3 ADMA2 Descriptor Table

Act2	Act1	Act0	Symbol	Operation
0	0	0	Nop	No operation
0	1	0	Rsv	Reserved
1	0	0	TRAN	Transfer data
1	1	0	LINK	Link descriptor

Table 6.35-3 The Format of Descriptor

6.35.5.4 Tuning Operation

Process of adjusting the sampling clock to optimally sample the received data. Normally achieved using the tuning command by host controller. Therefore, tuning is required to identify the correct phase where the data can be robustly sampled. It is used as a training sequence for sampling high frequency incoming data on multiple data lines. To sample such an input stream, the tuning engine in host controller requires an external programmable DelayLine(DL) to generate the multiple phases. SD host controller works with DelayLines that generate 8 phases. As shown in Figure 6.35-4, the output of host controller signals tuning_cclk_sel and tuning_cclk_sel_update are driven by the internal tuning engine and must be used for phase control of DelayLine. There outputs are used by tuning engine to change the incoming phase of cclk_rx clock so that it can find the phase at which incoming data on SD_DAT port can be robustly sampled.

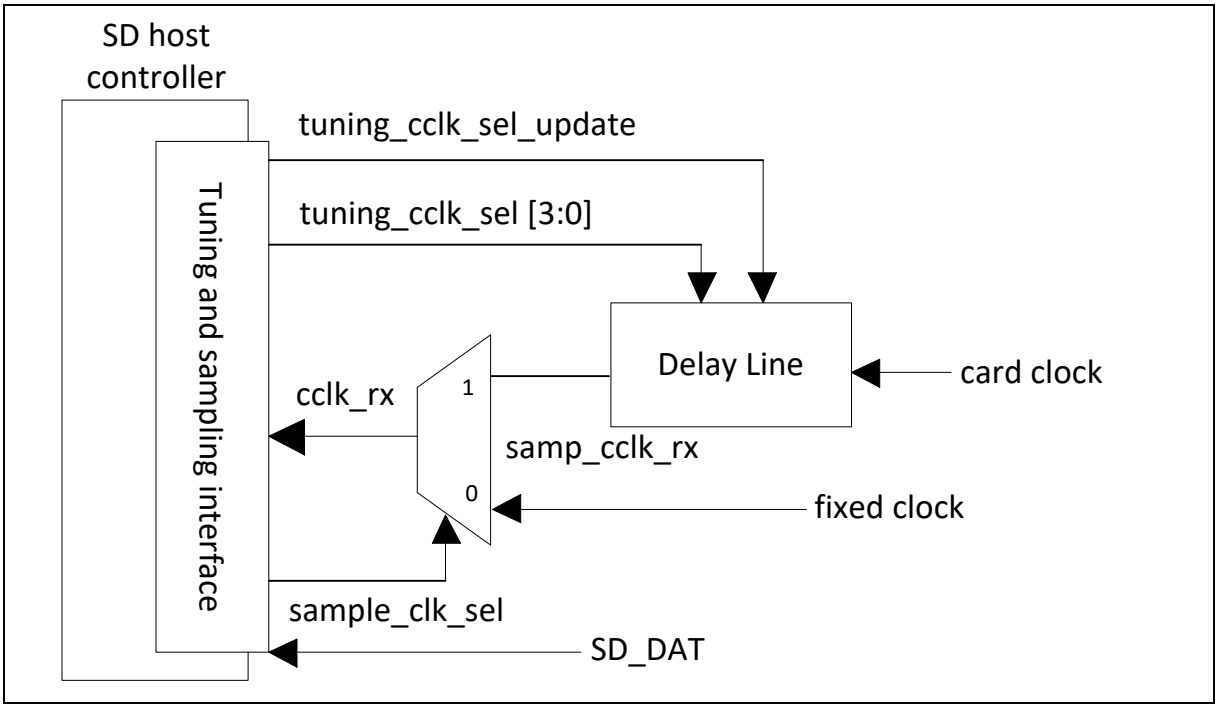


Figure 6.35-4 Tuning

The high-speed SDR modes that need tuning are eMMC HS200 and SD SDR104. The host controller controls the DelayLine and calculates the phase for robust sampling and the software determines the necessity for tuning and starting the tuning sequence.

Tuning engines look for errors in the received data or response frame at every selected phase step. If errors are found, then the phase is considered unfit for sampling and such a step is called a Fail step. However, if errors are not found, then the step is considered fit for sampling and such a step is called a Pass step. You must design the DelayLine with a resolution such that at least three successive Pass steps can be recognized by the tuning engine. A combination of Pass step and Fail step leads to scenarios show in Figure 6.35-5, all of which are supported by the tuning engine. Figure 6.35-5 uses a DelayLine with 16 phase steps to describe the supported tuning scenarios that are discussed in subsequent sections.

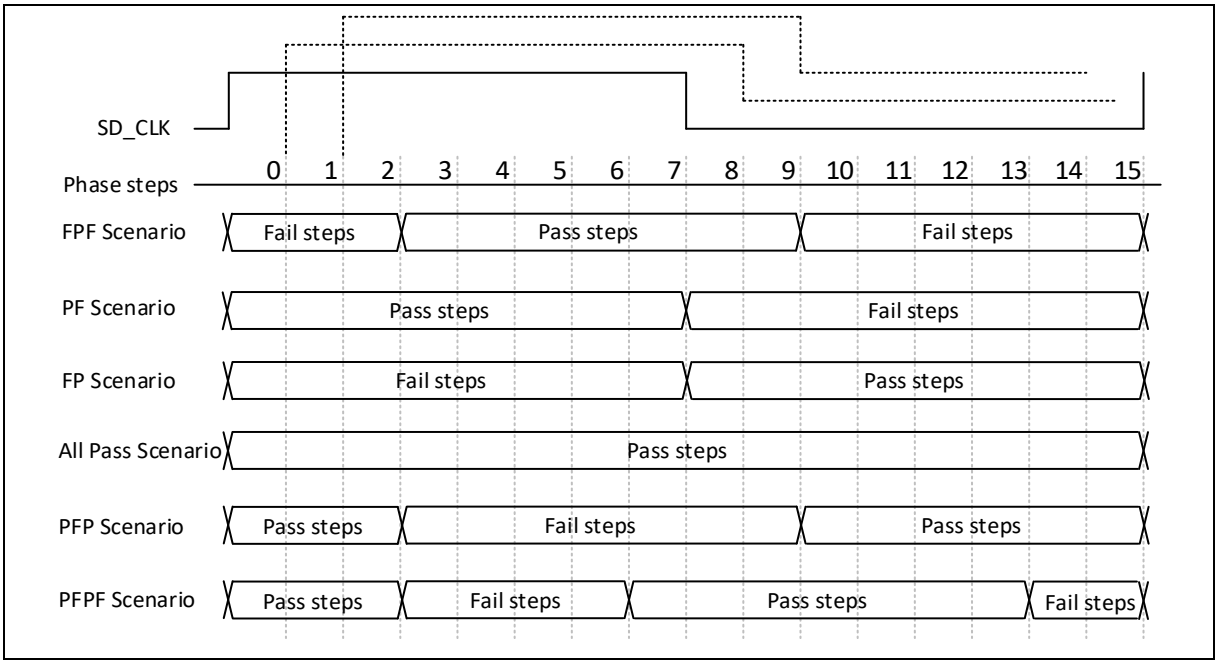


Figure 6.35-5 Pass Step and Fail Step Scenarios

Fail-Pass-Fail (FPF) Scenario

In the Fail-Pass-Fail scenario, a single burst of Pass steps is sandwiched between two bursts of Fail steps. In the example shown in Figure 6.35-5, the phase steps 2–9 are determined as Pass steps by the tuning engine. Among eight passing phase steps, phase step six is determined as the phase for robustly sampling the incoming data. If the default step was set to 0, the tuning engine stops at phase step 10 as it has identified a complete passing burst, that is a bust of Pass steps between Fail steps. This also avoids double locking, where the DelayLine can generate more than 360 degrees phase shift.

Pass-Fail (PF) Scenario

In case of a Pass-Fail, the total length of the DelayLine and the skews are such that tuning engine notices only a single burst of Fail steps at the end of Pass step burst. In the example shown in Figure 6.35-5, the phase steps 0 to 7 are Pass steps and the tuning engine determines phase step 4 as the phase for robustly sampling incoming data.

Fail-Pass (FP) Scenario

In case of a Fail-Pass, the total length of the DelayLine and the skews are such that the tuning engine notices only a single burst of Fail steps at the beginning of the Pass-step burst. In the example shown in Figure 6.35-5, the phase steps 7 to 15 are Pass steps and the tuning engine determines phase step 11 as the phase for robustly sampling incoming data.

All Pass Scenario

An All pass case occurs if the skew between data lines is less than one step of the DelayLine and due to resolution loss, the tuning engine determines all steps as Pass steps. The phase step for robustly sampling incoming data depends on the default tuning step. With default step as 0, sampling step is selected as step 8.

Pass-Fail-Pass (PFP) Scenario

In case of Pass-Fail-Pass, the tuning engine notices two passing bursts. This happens when the DelayLine length and the step 0 phase is such that the tuning engine can sample data launched in CycleN in the initial phase-steps and the data launched in CycleN+1 in the latter phase steps. This is shown in Figure 6.35-6.

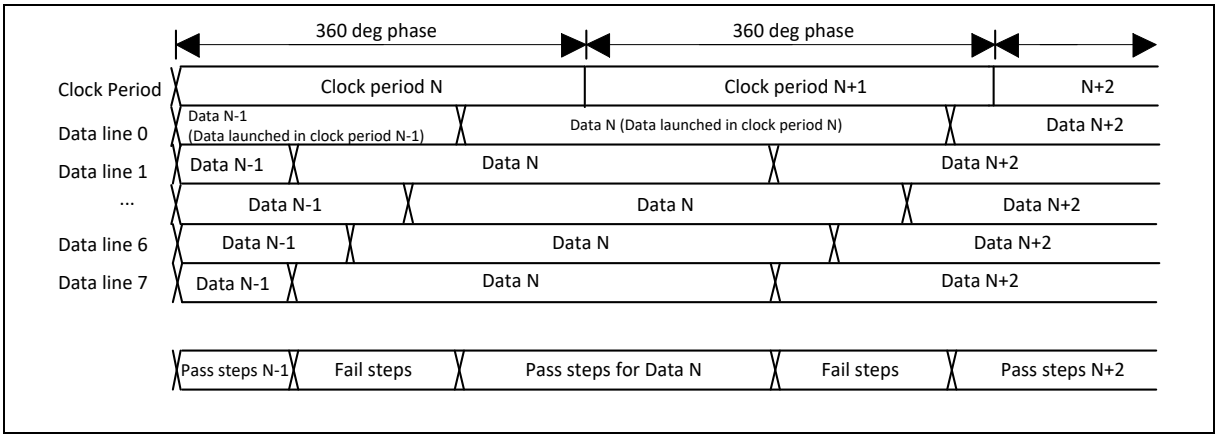


Figure 6.35-6 Sampling of Data In PFP

In these cases, the Tuning engines assume that the Pass steps for Data N-1 in clock period N would also repeat for Data N in clock period N+1. Considering this assumption, tuning engine adds the Pass steps from both the bursts to calculate a robust sampling point. In the example shown in Figure 6.35-5, the tuning engine selects phase step15 as the phase for robustly sampling incoming data.

Pass-Fail-Pass-Fail (PFPF) Scenario

In case of Pass-Fail-Pass-Fail, the tuning engine notices one partial Pass step burst for Data N-1 and a complete Pass-step burst for Data N+1. The tuning engine prefers a complete Pass step over partial Pass-step bursts and selects a phase within the complete Pass-step burst for robustly sampling incoming data. In the example shown in Figure 6.35-5 phase step 10 is selected as the phase for robustly sampling incoming data.

Threshold Based Selection Tuning Schema

This mode allows the tuning engine to select the first complete sampling window that meets the threshold criteria defined. In this mode a complete sampling window is preferred over a partial sampling window. Tuning stops when a complete sampling window meets the threshold criteria. Tuning also stops when all the taps are parsed as shown in Figure 6.35-7. You can enable this mode using SWIN_TH_VAL (AT_CTRL_R[x:24]) as required for design, and set SWIN_TH_EN (AT_CTRL_R[2]) to '1'.

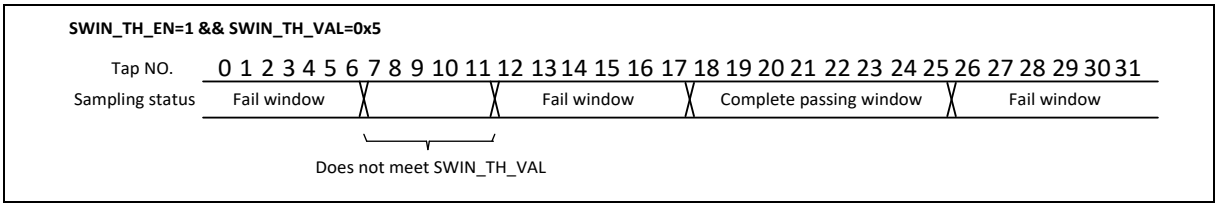


Figure 6.35-7 Threshold Based Selection Tuning Schema

Large Sampling Window Tuning Schema

This mode allows the tuning algorithm to move through all the taps of delay line to identify the largest sampling window available. In this mode, tuning stops only when all the taps are parsed. The algorithm picks the largest sampling window as shown in Figure 6.35-8. If there are two sampling windows of same length, then the first one is selected. You can enable this mode using SWIN_TH_EN (AT_CTRL_R[2]) to '0', and do not care SWIN_TH_VAL (AT_CTRL_R[x:24]).

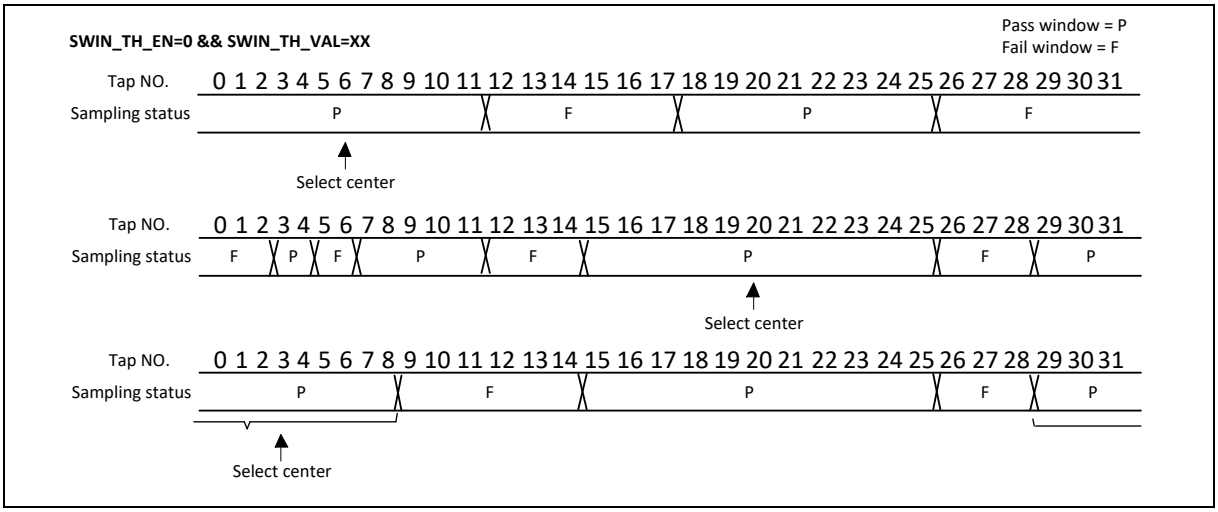


Figure 6.35-8 Large Sampling Window Tuning Schema

Auto-Tuning

Auto-tuning is a hardware-managed re-tuning feature. Auto-tuning removes the need for the host software to re-tune the sampling clock for every 4 Mbytes of data transfer. A tuning sampling clock is recommended in both SD and eMMC modes to ensure ease in timing closure and for robust operation while operating at high SDR speeds, such as SDR104 and HS200.

Auto-tuning requires a dedicated Delayline (DL) that is an instance of the sampling DelayLine. This instance of DL is called as the Drift Measurement (DM) DelayLine. Figure 6.35-9 shows the top level block diagram for auto-tuning.

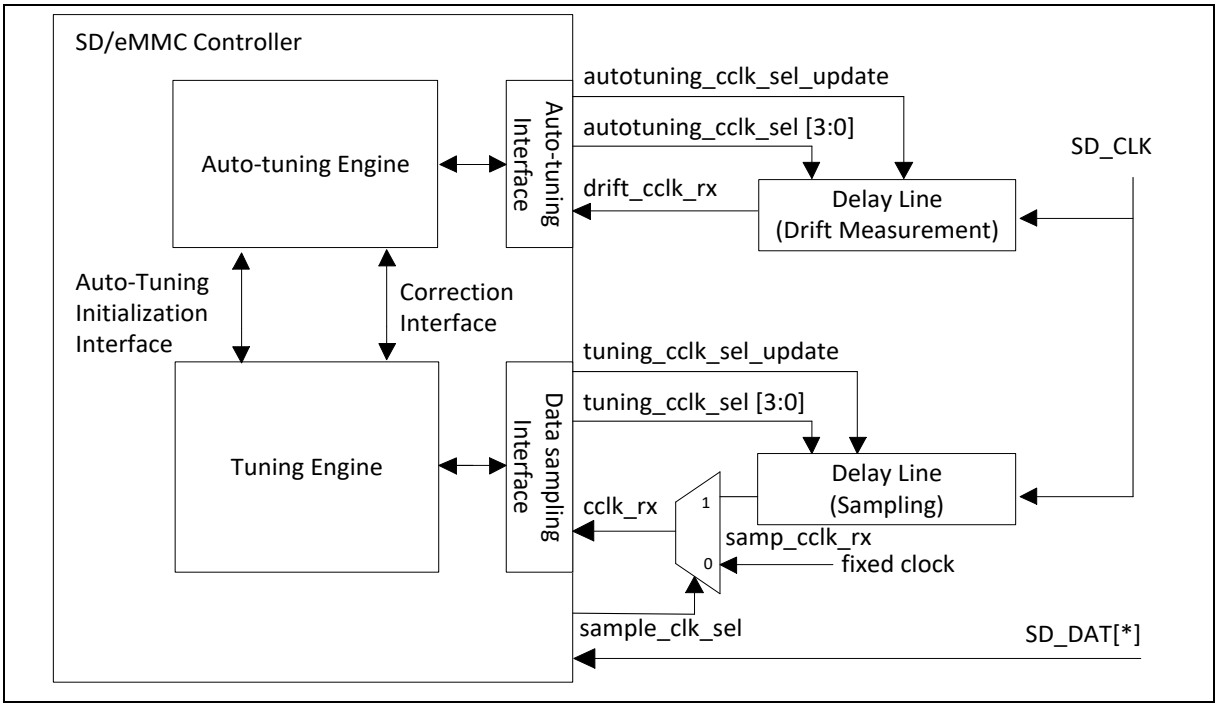


Figure 6.35-9 Auto-Tuning Top Level Block Diagram

Software must initiate tuning sequence based on CMD19 (for SD) or CMD21 (for eMMC) for initializing the Auto-tuning engine. After initialization, the auto-tuning engine is activated by default for all Data read transfers. Auto-Tuning engine uses a DM DelayLine to generate drift_cclk_rx. The drift_cclk_rx is generated using the same source as cclk_rx; however, its phase is controlled by the

autotuning_cclk_sel[3:0] outputs of the auto-tuning engine. Auto-Tuning engine uses the data sampled on drift_cclk_rx to check for drifts on the data lines. A drift is detected if data CRC errors, start bit error and end bit error. These drift checks are active only when READ transfers are in progress in HS200/SDR104 modes. On drift detection, auto-tuning engine sweeps through DM DelayLine phases until it finds a sampling window without drifts. The new sampling window range allows the auto-tuning engine to calculate a correction for the sampling phase. The calculated correction is applied on the sampling DelayLine using tuning_cclk_sel[3:0] outputs at the software-selected correction interval.

Figure 6.35-10 shows auto-tuning interface timing. SDR104 and HS200 speed modes allow a minimum of 8 idle cycles between 2 blocks (block-gap) where the device/card tri-states its output and is not driving any data. SD host controller changes auto_tuning_cclk_sel[3:0] and auto_tuning_cclk_update outputs synchronous to SD_CLK in this block gap.

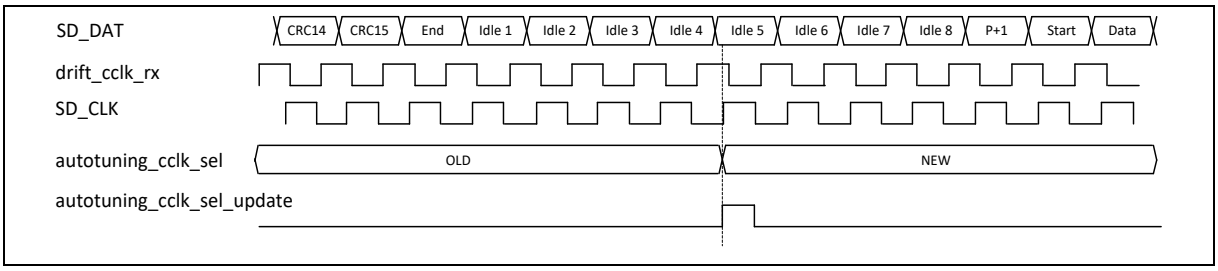


Figure 6.35-10 Auto-Tuning Interface Timing

The tuning and auto-tuning interface provides the autotuning_cclk_sel_update and tuning_cclk_sel_update signals to indicate that the phase select code is being changed. The DelayLine can use these signals to update the phase at a timing where no glitches are generated at the output of DelayLines. If the update cannot be managed, SD host controller enables the phase select code to be changed only after stopping cclk_rx and drift_cclk_rx clocks, and after changed clocks are restarted. This results in safe phase code switching. Clocks are stopped by pulling down SD_CLK_EN (CLK_CTRL_R[2]) output that is generated on SD_CLK. The gated clock has an unknown phase relationship with cclk_rx and drift_cclk_rx due to DelayLines. This being the case programmable latency fields of AT_CTRL_R can be used to define this relationship as shown in Figure 6.35-11.

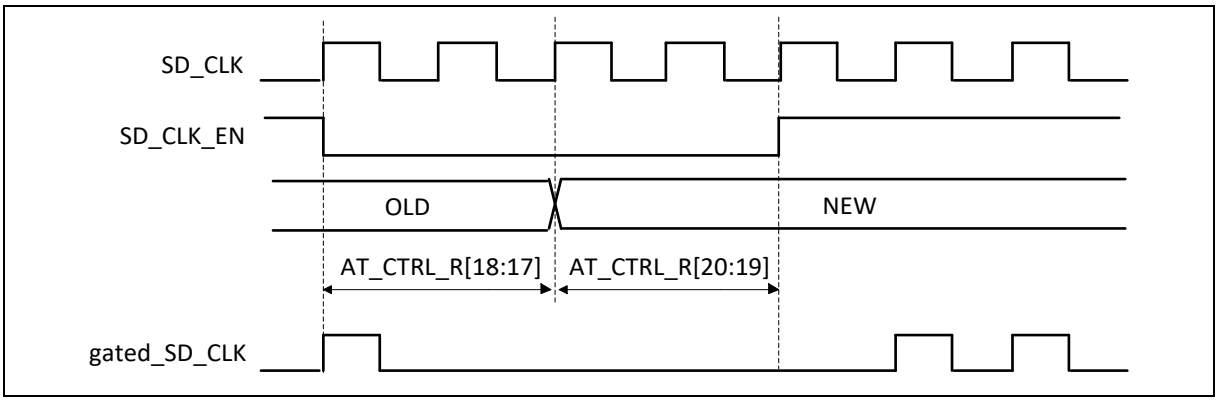


Figure 6.35-11 Glitch-Free Phase Switching

The Glitch-free clock switching feature is disabled by default and can be enabled by programming the TUNE_CLK_STOP_EN (AT_CTRL_R[16]) bit to 1. However, this feature can reduce the performance by a maximum of 2% as a few cycles of clocks are stopped between a block.

The auto-tuning is enabled by default in SD Host controller when the speed mode is switched to SDR104 or HS200. If there is a need to change any of its parameters or to disable it, AT_CTRL_R register can be used to enable or disable. AT_CTRL_R can be programmed only when SAMPLE_CLK_SEL (HOST_CTRL2_R[7]) is set to 0, that is before starting tuning.

AT_CTRL_R is the register that controls some aspects of tuning and auto-tuning features. Do not program this register when SAMPLE_CLK_SEL (HOST_CTRL2_R[7]) is ‘1’.

Setting AT_EN (AT_CTRL_R[0]) to enable auto tuning engine, and set EXEC_TUNING (HOST_CTRL2_R[6]) to '1' before issue tuning command (CMD19). A CMD19 is used to start the tuning procedure. CMD19 is similar to any read command. Therefore, before writing CMD19 into the CMD_R, the settings of BLOCKSIZE_R, BLOCKCOUNT_R, and XFERMODE_R registers must be proper. After issue CMD19, check BUF_RD_READY (NORMAL_INT_STAT_R[5]) to '1' and wait for transfer to complete.

6.35.5.5 Ultra-High-Speed phase I (UHS-I) Voltage Switch

UHS-I mode requires the support for 1.8V signaling. Voltage switch command (CMD11) invokes the voltage switch sequence to the 1.8V mode. After power up, signaling level is 3.3V, when the voltage switch sequence is completed that the card enters UHS-I mode with default mode (SDR12), and the card signaling level is changed to 1.8V.

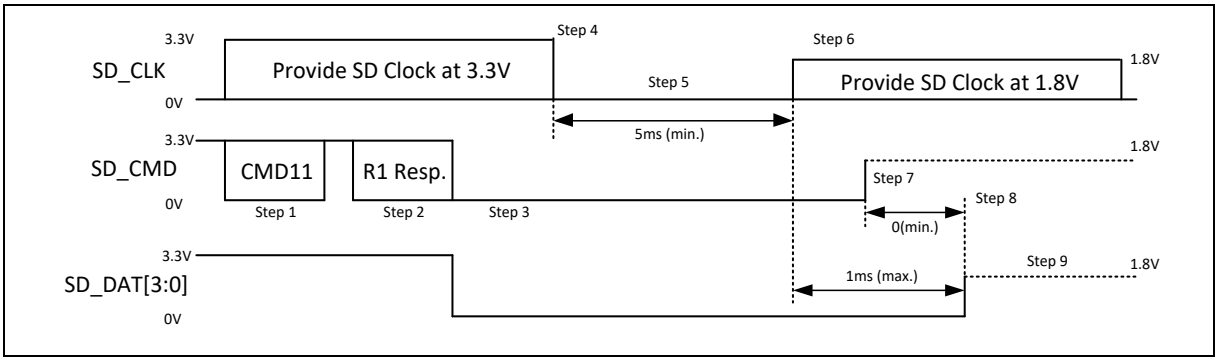


Figure 6.35-12 Signal Voltage Switch Sequence

Figure 6.35-12 shows voltage switch sequence. To perform the signal voltage switch sequence the following steps are needed:

1. Step 1. Host issue CMD11 to start voltage switch sequence.
2. Step 2. The card returns R1 response.
3. Step 3. The card drives SD_CMD and SD_DAT[3:0] to low immediately after the response.
4. Step 4. The host stops supplying SD_CLK. The card shall start switching voltage after host stops SD_CLK. The time to stop SD_CLK is not specified. The host can detect whether the sequence starts by checking signal level of either one of CMD, SD_DAT[3:0]. Which signal should be checked depends on ability of the host. If low level is not detected, the host should abort the sequence and execute power cycle.
5. Step 5. 1.8V output of voltage regulator in card shall be stable within 5ms. Host keeps SD_CLK low at least 5ms. This means that 5ms is the maximum for the card and the minimum for the host.
6. Step 6. After 5ms from step 4 and host voltage regulator is stable, the host starts providing SD_CLK at 1.8V. The card can check whether SD_CLK voltage is 1.8V.
7. Step 7. By detecting SD_CLK, the card drives SD_CMD to high at 1.8V at least one clock and then stop driving (tri-state). SD_CMD is triggered by rising edge of SD_CLK (SDR timing).
8. Step 8. The card can check whether host drives SD_CMD to 1.8V through the host pull-up resister.
9. Step 9. If switching to 1.8V signaling is completed successfully, the card drives SD_DAT[3:0] to high at 1.8V at least one clock and then stop driving (tri-state). SD_DAT[3:0] is triggered by rising edge of SD_CLK (SDR timing). SD_DAT[3:0] shall be high within 1ms from start of providing SD_CLK. Host check whether SD_DAT[3:0] is high after 1ms from supplying SD_CLK. This means that 1ms is the maximum for the card and the minimum for the host.

10.After the sequence is completed, the host and the card start communication in SDR12 timing.

6.35.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
SDH Base address: **SDHx_BA = 0x4018_0000+(0x0001_0000*x)** **x=0,1**				
SDMASA_R	SDHx_BA+0x000	R/W	SDMA System Address Register	0x0000_0000
BLOCKSIZE_R	SDHx_BA+0x004	R/W	Block Size Register	0x0000
BLOCKCOUNT_R	SDHx_BA+0x006	R/W	16-bit Block Count Register	0x0000
ARGUMENT_R	SDHx_BA+0x008	R/W	Argument Register	0x0000_0000
XFER_MODE_R	SDHx_BA+0x00c	R/W	Transfer Mode Register	0x0000
CMD_R	SDHx_BA+0x00e	R/W	Command Register	0x0000
RESP01_R	SDHx_BA+0x010	R	Response Register 0/1	0x0000_0000
RESP23_R	SDHx_BA+0x014	R	Response Register 2/3	0x0000_0000
RESP45_R	SDHx_BA+0x018	R	Response Register 4/5	0x0000_0000
RESP67_R	SDHx_BA+0x01c	R	Response Register 6/7	0x0000_0000
BUF_DATA_R	SDHx_BA+0x020	R/W	Buffer Data Port Register	0x0000_0000
PSTATE_REG	SDHx_BA+0x024	R	Present State Register	0x0000_0000
HOST_CTRL1_R	SDHx_BA+0x028	R/W	Host Control 1 Register	0x00
PWR_CTRL_R	SDHx_BA+0x029	R/W	Power Control Register	0x00
BGAP_CTRL_R	SDHx_BA+0x02a	R/W	Block Gap Control Register	0x00
WUP_CTRL_R	SDHx_BA+0x02b	R/W	Wakeup Control Register	0x00
CLK_CTRL_R	SDHx_BA+0x02c	R/W	Clock Control Register	0x0000
TOUT_CTRL_R	SDHx_BA+0x02e	R/W	Timeout Control Register	0x00
SW_RST_R	SDHx_BA+0x02f	R/W	Software Reset Register	0x00
NORMAL_INT_STAT_R	SDHx_BA+0x030	R/W	Normal Interrupt Status Register	0x0000
ERROR_INT_STAT_R	SDHx_BA+0x032	R/W	Error Interrupt Status Register	0x0000
NORMAL_INT_STAT_EN_R	SDHx_BA+0x034	R/W	Normal Interrupt Status Enable Register	0x0000
ERROR_INT_STAT_EN_R	SDHx_BA+0x036	R/W	Error Interrupt Status Enable Register	0x0000
NORMAL_INT_SIGNAL_EN_R	SDHx_BA+0x038	R/W	Normal Interrupt Signal Enable Register	0x0000
ERROR_INT_SIGNAL_EN_R	SDHx_BA+0x03a	R/W	Error Interrupt Signal Enable Register	0x0000
AUTO_CMD_STAT_R	SDHx_BA+0x03c	R	Auto CMD Status Register	0x0000
HOST_CTRL2_R	SDHx_BA+0x03e	R/W	Host Control 2 Register	0x0000
CAPABILITIES1_R	SDHx_BA+0x040	R	Capabilities 1 Register - 0 to 31	0x276E_C898
CAPABILITIES2_R	SDHx_BA+0x044	R	Capabilities Register - 32 to 63	0x0800_8077

CURR_CAPABILITIES1_R	SDHx_BA+0x048	R	Maximum Current Capabilities Register - 0 to 31	0x0000_0000
CURR_CAPABILITIES2_R	SDHx_BA+0x04c	R	Maximum Current Capabilities Register - 32 to 63	0x0000_0000
FORCE_AUTO_CMD_STAT_R	SDHx_BA+0x050	R/W	Force Event Register for Auto CMD Error Status register	0x0000
FORCE_ERROR_INT_STAT_R	SDHx_BA+0x052	R/W	Force Event Register for Error Interrupt Status	0x0000
ADMA_ERR_STAT_R	SDHx_BA+0x054	R	ADMA Error Status Register	0x00
ADMA_SA_LOW_R	SDHx_BA+0x058	R/W	ADMA System Address Register - Low	0x0000_0000
PRESET_INIT_R	SDHx_BA+0x060	R	Preset Value for Initialization	0x0000
PRESET_DS_R	SDHx_BA+0x062	R/W	Preset Value for Default Speed	0x0000
PRESET_HS_R	SDHx_BA+0x064	R/W	Preset Value for High Speed	0x0000
PRESET_SDR12_R	SDHx_BA+0x066	R/W	Preset Value for SDR12	0x0000
PRESET_SDR25_R	SDHx_BA+0x068	R/W	Preset Value for SDR25	0x0000
PRESET_SDR50_R	SDHx_BA+0x06a	R/W	Preset Value for SDR50	0x0000
PRESET_SDR104_R	SDHx_BA+0x06c	R/W	Preset Value for SDR104	0x0000
PRESET_DDR50_R	SDHx_BA+0x06e	R	Preset Value for DDR50	0x0000
PRESET_UHS2_R	SDHx_BA+0x074	R/W	Preset Value for UHS-II	0x0000
P_EMBEDDED_CNTRL	SDHx_BA+0xe6	R	Pointer for Embedded Control	0x0F6C
P_VENDOR_SPECIFIC_AREA	SDHx_BA+0xe8	R	Pointer for Vendor Specific Area 1	0x0500
P_VENDOR2_SPECIFIC_AREA	SDHx_BA+0xea	R	Pointer for Vendor Specific Area 2	0x0180
SLOT_INTR_STATUS_R	SDHx_BA+0xfc	R	Slot Interrupt Status Register	0x0000
HOST_CNTRL_VERS_R	SDHx_BA+0xfe	R	Host Controller Version	0x0005
EMBEDDED_CTRL_R	SDHx_BA+0xf6c	R/W	Embedded Control register	0x0000_000
MSHC_VER_ID_R	SDHx_BA+0x500	R	MSHC version	0x3138_302A
MSHC_VER_TYPE_R	SDHx_BA+0x504	R	MSHC version type	0x6761_2A2A
MSHC_CTRL_R	SDHx_BA+0x508	R/W	MSHC Control register	0x01
MBIU_CTRL_R	SDHx_BA+0x510	R/W	MBIU Control register	0x0F
EMMC_CTRL_R	SDHx_BA+0x52c	R/W	eMMC Control register	0x000C
BOOT_CTRL_R	SDHx_BA+0x52e	R/W	eMMC Boot Control register	0x0000
AT_CTRL_R	SDHx_BA+0x540	R/W	Tuning and Auto-tuning control register	0x0300_0005
AT_STAT_R	SDHx_BA+0x544	R/W	Tuning and Auto-tuning status register	0x0000_0006
CQCAP	SDHx_BA+0x184	R	Command Queuing Capabilities register	0x0000_30C8

6.35.7 Register Description

SDMASA R Register (SDMASA_R)

Register	Offset	R/W	Description	Reset Value
SDMASA_R	SDHx_BA+0x000	R/W	SDMA System Address Register	0x0000_0000

31	30	29	28	27	26	25	24
BLOCKCNT_SDMASA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
BLOCKCNT_SDMASA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
BLOCKCNT_SDMASA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BLOCKCNT_SDMASA							

Bits	Description	
[31:0]	**BLOCKCNT_SDMASA**	**32-bit Block Count** • SDMA System Address (Host Version 4 Enable = 0): This register contains the system memory address for an SDMA transfer in the 32-bit addressing mode. When the Host Controller stops an SDMA transfer, this register points to the system address of the next contiguous data position. It can be accessed only if no transaction is executing. Reading this register during data transfers may return an invalid value. • 32-bit Block Count (Host Version 4 Enable = 1): From the Host Controller Version 4.10 specification, this register is redefined as 32-bit Block Count. The Host Controller decrements the block count of this register for every block transfer and the data transfer stops when the count reaches zero. This register must be accessed when no transaction is executing. Reading this register during data transfers may return invalid value. Following are the values for BLOCKCNT_SDMASA: 0xFFFF_FFFF = 4G - 1 Block ... 0x0000_0002 = 2 Blocks 0x0000_0001 = 1 Block 0x0000_0000 = Stop Count **Note:** • For Host Version 4 Enable = 0, the Host driver does not program the system address in this register while operating in ADMA mode. The system address must be programmed in the ADMA System Address register. • For Host Version 4 Enable = 0, the Host driver programs a non-zero 32-bit block count value in this register when Auto CMD23 is enabled for non-DMA and ADMA modes. Auto CMD23 cannot be used with SDMA. • This register must be programmed with a non-zero value for data transfer if the 32-bit Block count register is used instead of the 16-bit Block count register.

BLOCKSIZE R Register (BLOCKSIZE_R)

Register	Offset	R/W	Description	Reset Value
BLOCKSIZE_R	SDHx_BA+0x004	R/W	Block Size Register	0x0000

15	14	13	12	11	10	9	8
RSVD_BLOCKSIZE15	SDMA_BUF_BDARY			XFER_BLOCK_SIZE			
7	6	5	4	3	2	1	0
XFER_BLOCK_SIZE							

Bits	Description	
[15]	**RSVD_BLOCKSIZE15**	This bit of the BLOCKSIZE_R register is reserved and always returns 0.
[14:12]	**SDMA_BUF_BDARY**	**SDMA Buffer Boundary** These bits specify the size of contiguous buffer in system memory. The SDMA transfer waits at every boundary specified by these fields and the Host Controller generates the DMA interrupt to request the Host Driver to update the SDMA System Address register. 0x0 = 4 Kbytes SDMA Buffer Boundary 0x1 = 8 Kbytes SDMA Buffer Boundary 0x2 = 16 Kbytes SDMA Buffer Boundary 0x3 = 32 Kbytes SDMA Buffer Boundary 0x4 = 64 Kbytes SDMA Buffer Boundary 0x5 = 128 Kbytes SDMA Buffer Boundary 0x6 = 256 Kbytes SDMA Buffer Boundary 0x7 = 512 Kbytes SDMA Buffer Boundary
[11:0]	**XFER_BLOCK_SIZE**	**Transfer Block Size** These bits specify the block size of data transfers. In case of memory, it is set to 512 bytes. It can be accessed only if no transaction is executing. Read operations during transfers may return an invalid value, and write operations are ignored. Following are the values for XFER_BLOCK_SIZE: 0x1 = 1 byte 0x2 = 2 bytes 0x3 = 3 bytes ..... 0x1FF = 511 byte 0x200 = 512 bytes ..... 0x800 = 2048 bytes **Note:** This register must be programmed with a non-zero value for data transfer.

BLOCKCOUNT R Register (BLOCKCOUNT_R)

Register	Offset	R/W	Description	Reset Value
BLOCKCOUNT_R	SDHx_BA+0x006	R/W	16-bit Block Count Register	0x0000

15	14	13	12	11	10	9	8
BLOCK_CNT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BLOCK_CNT							

Bits	Description	
[15:0]	**BLOCK_CNT**	**16-bit Block Count** • If the Host Version 4 Enable bit is set 0 or the 16-bit Block Count register is set to non-zero, the 16-bit Block Count register is selected. • If the Host Version 4 Enable bit is set 1 and the 16-bit Block Count register is set to zero, the 32-bit Block Count register is selected. Following are the values for BLOCK_CNT: 0x0 = Stop Count 0x1 = 1 Block 0x2 = 2 Blocks ... - ... 0xFFFF = 65535 Blocks **Note:** For Host Version 4 Enable = 0, this register must be set to 0000h before programming the 32-bit block count register when Auto CMD23 is enabled for non-DMA and ADMA modes.

ARGUMENT R Register (ARGUMENT_R)

Register	Offset	R/W	Description	Reset Value
ARGUMENT_R	SDHx_BA+0x008	R/W	Argument Register	0x0000_0000

31	30	29	28	27	26	25	24
ARGUMENT							
23	22	21	20	19	18	17	16
ARGUMENT							
15	14	13	12	11	10	9	8
ARGUMENT							
7	6	5	4	3	2	1	0
ARGUMENT							

Bits	Description	
[31:0]	ARGUMENT	**Command Argument** These bits specify the SD/eMMC command argument that is specified in bits 39-8 of the Command format.

XFER MODE R Register (XFER_MODE_R)

Register	Offset	R/W	Description	Reset Value
XFER_MODE_R	SDHx_BA+0x00c	R/W	Transfer Mode Register	0x0000

15	14	13	12	11	10	9	8
RSVD							RESP_INT_DISABLE
7	6	5	4	3	2	1	0
RESP_ERR_CHK_ENABLE	RESP_TYPE	MULTI_BLK_SEL	DATA_XFER_DIR	AUTO_CMD_ENABLE		BLOCK_COUNT_ENABLE	DMA_ENABLE

Bits	Description	
[15:9]	**RSVD**	These bits of the XFER_MODE_R register are reserved and always return 0.
[8]	**RESP_INT_DISABLE**	**Response Interrupt Disable** The Host Controller supports response check function to avoid overhead of response error check by the Host driver. Response types of only R1 and R5 can be checked by the Controller. If Host Driver checks the response error, set this bit to 0 and wait for Command Complete Interrupt and then check the response register. If the Host Controller checks the response error, set this bit to 1 and set the Response Error Check Enable bit to 1. The Command Complete Interrupt is disabled by this bit regardless of the Command Complete Signal Enable. **Note:** During tuning (when the Execute Tuning bit in the Host Control2 register is set), the Command Complete Interrupt is not generated irrespective of the Response Interrupt Disable setting. 0x0 = Response Interrupt is enabled 0x1 = Response Interrupt is disabled
[7]	**RESP_ERR_CHK_ENABLE**	**Response Error Check Enable** 0x0 = Response Error Check is disabled 0x1 = Response Error Check is enabled The Host Controller supports response check function to avoid overhead of response error check by Host driver. Response types of only R1 and R5 can be checked by the Controller. If the Host Controller checks the response error, set this bit to 1 and set Response Interrupt Disable to 1. If an error is detected, the Response Error interrupt is generated in the Error Interrupt Status register. **Note:** • Response error check must not be enabled for any response type other than R1 and R5. • Response check must not be enabled for the tuning command.
[6]	**RESP_TYPE**	**Response Type R1/R5** This bit selects either R1 or R5 as a response type when the Response Error Check is selected. Error statuses checked in R1: • OUT_OF_RANGE • ADDRESS_ERROR • BLOCK_LEN_ERROR • WP_VIOLATION • CARD_IS_LOCKED • COM_CRC_ERROR • CARD_ECC_FAILED

		• CC_ERROR • ERROR Response Flags checked in R5: • COM_CRC_ERROR • ERROR • FUNCTION_NUMBER • OUT_OF_RANGE 0x0 (RESP_R1) = R1 (Memory) 0x1 (RESP_R5) = R5 (SDIO)
[5]	**MULTI_BLK_SEL**	**Multi/Single Block Select** This bit is set when issuing multiple-block transfer commands using the DAT line. If this bit is set to 0, it is not necessary to set the Block Count register. 0x0 (SINGLE) = Single Block 0x1 (MULTI) = Multiple Block
[4]	**DATA_XFER_DIR**	**Data Transfer Direction Select** This bit defines the direction of DAT line data transfers. This bit is set to 1 by the Host Driver to transfer data from the SD/eMMC card to the Host Controller and it is set to 0 for all other commands. 0x1 (READ) = Read (Card to Host) 0x0 (WRITE) = Write (Host to Card)
[3:2]	**AUTO_CMD_ENABLE**	**Auto Command Enable** This field determines use of Auto Command functions. **Note:** In SDIO, this field must be set as 00b (Auto Command Disabled). 0x0 (AUTO_CMD_DISABLED) = Auto Command Disabled 0x1 (AUTO_CMD12_ENABLED) = Auto CMD12 Enable 0x2 (AUTO_CMD23_ENABLED) = Auto CMD23 Enable 0x3 (AUTO_CMD_AUTO_SEL) = Auto CMD Auto Select
[1]	**BLOCK_COUNT_ENABLE**	**Block Count Enable** This bit is used to enable the Block Count register, which is relevant for multiple block transfers. If this bit is set to 0, the Block Count register is disabled, which is useful in executing an infinite transfer. The Host Driver must set this bit to 0 when ADMA is used. 0x1 (ENABLED) = Enable 0x0 (DISABLED) = Disable
[0]	**DMA_ENABLE**	**DMA Enable** This bit enables the DMA functionality. If this bit is set to 1, a DMA operation begins when the Host Driver writes to the Command register. You can select one of the DMA modes by using DMA Select in the Host Control 1 register. 0x1 (ENABLED) = DMA Data transfer 0x0 (DISABLED) = No data transfer or Non-DMA data transfer

CMD R Register (CMD_R)

Register	Offset	R/W	Description	Reset Value
CMD_R	SDHx_BA+0x00e	R/W	Command Register	0x0000

15	14	13	12	11	10	9	8
RSVD		CMD_INDEX					
7	6	5	4	3	2	1	0
CMD_TYPE		DATA_PRESENT_SEL	CMD_IDX_CHK_ENABLE	CMD_CRC_CHK_ENABLE	SUB_CMD_FLAG	RESP_TYPE_SELECT	

Bits	Description	
[15:14]	**RSVD**	These bits of the CMD_R register are reserved and always return 0.
[13:8]	**CMD_INDEX**	**Command Index** These bits are set to the command number that is specified in bits 45-40 of the Command Format.
[7:6]	**CMD_TYPE**	**Command Type** 0x3 (ABORT_CMD) = Abort 0x2 (RESUME_CMD) = Resume 0x1 (SUSPEND_CMD) = Suspend 0x0 (NORMAL_CMD) = Normal These bits indicate the command type. **Note:** While issuing Abort CMD using CMD12/CMD52 or reset CMD using CMD0/CMD52, CMD_TYPE field shall be set to 0x3.
[5]	**DATA_PRESENT_SEL**	**Data Present Select** 0x0 (NO_DATA) = No Data Present 0x1 (DATA) = Data Present This bit is set to 1 to indicate that data is present and that the data is transferred using the DAT line. This bit is set to 0 in the following instances: • Command using the CMD line • Command with no data transfer but using busy signal on the DAT[0] line • Resume Command
[4]	**CMD_IDX_CHK_ENABLE**	**Command Index Check Enable** 0x0 (DISABLED) = Disable 0x1 (ENABLED) = Enable This bit enables the Host Controller to check the index field in the response to verify if it has the same value as the command index. If the value is not the same, it is reported as a Command Index error. **Note:** • Index Check enable must be set to 0 for the command with no response, R2 response, R3 response and R4 response. • For the tuning command, this bit must always be set to enable the index check.
[3]	**CMD_CRC_CHK_ENABLE**	**Command CRC Check Enable** 0x0 (DISABLED) = Disable 0x1 (ENABLED) = Enable This bit enables the Host Controller to check the CRC field in the response. If an error is detected, it is reported as a Command CRC error.

		Note: CRC Check enable must be set to 0 for the command with no response, R3 response, and R4 response. • For the tuning command, this bit must always be set to 1 to enable the CRC check.
[2]	**SUB_CMD_FLAG**	**Sub Command Flag** 0x0 (MAIN) = Main Command 0x1 (SUB) = Sub Command This bit distinguishes between a main command and a sub command.
[1:0]	**RESP_TYPE_SELECT**	**Response Type Select** 0x0 (NO_RESP) = No Response 0x1 (RESP_LEN_136) = Response Length 136 0x2 (RESP_LEN_48) = Response Length 48 0x3 (RESP_LEN_48B) = Response Length 48; Check Busy after response This bit indicates the type of response expected from the card.

RESP01 R Register (RESP01_R)

Register	Offset	R/W	Description	Reset Value
RESP01_R	SDHx_BA+0x010	R	Response Register 0/1	0x0000_0000

31	30	29	28	27	26	25	24
RESP01							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESP01							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESP01							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESP01							

Bits	Description	
[31:0]	**RESP01**	**Command Response** These bits reflect 39-8 bits of SD/eMMC Response Field. **Note:** For Auto CMD, the 32-bit response (bits 39-8 of the Response Field) is updated in the RESP67_R register.

RESP23 R Register (RESP23_R)

Register	Offset	R/W	Description	Reset Value
RESP23_R	SDHx_BA+0x014	R	Response Register 2/3	0x0000_0000

31	30	29	28	27	26	25	24
RESP23							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESP23							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESP23							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESP23							

Bits	Description	
[31:0]	**RESP23**	**Command Response** These bits reflect 71-40 bits of the SD/eMMC Response Field.

RESP45 R Register (RESP45_R)

Register	Offset	R/W	Description	Reset Value
RESP45_R	SDHx_BA+0x018	R	Response Register 4/5	0x0000_0000

31	30	29	28	27	26	25	24
RESP45							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESP45							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESP45							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESP45							

Bits	Description	
[31:0]	**RESP45**	**Command Response** These bits reflect 103-72 bits of the Response Field.

RESP67 R Register (RESP67_R)

Register	Offset	R/W	Description	Reset Value
RESP67_R	SDHx_BA+0x01c	R	Response Register 6/7	0x0000_0000

31	30	29	28	27	26	25	24
RESP67							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESP67							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESP67							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESP67							

Bits	Description	
[31:0]	**RESP67**	**Command Response** These bits reflect bits 135-104 of SD/EMMC Response Field. **Note:** For Auto CMD, this register also reflects the 32-bit response (bits 39-8 of the Response Field).

BUF DATA R Register (BUF_DATA_R)

Register	Offset	R/W	Description	Reset Value
BUF_DATA_R	SDHx_BA+0x020	R/W	Buffer Data Port Register	0x0000_0000

31	30	29	28	27	26	25	24
BUF_DATA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
BUF_DATA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
BUF_DATA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BUF_DATA							

Bits	Description	
[31:0]	**BUF_DATA**	**Buffer Data** These bits enable access to the Host Controller packet buffer.

PSTATE REG Register (PSTATE_REG)

Register	Offset	R/W	Description	Reset Value
PSTATE_REG	SDHx_BA+0x024	R	Present State Register	0x0000_0000

31	30	29	28	27	26	25	24
UHS2_IF_DETECT	LANE_SYNC	IN_DORMANT_ST	SUB_CMD_STAT	CMD_ISSUE_ERR	RSVD_26	HOST_REG_VOL	CMD_LINE_LVL
23	22	21	20	19	18	17	16
DAT_3_0				WR_PROTECT_SW_LVL	CARD_DETECT_PIN_LEVEL	CARD_STABLE	CARD_INSERTED
15	14	13	12	11	10	9	8
RSVD_15_12				BUF_RD_ENABLE	BUF_WR_ENABLE	RD_XFER_ACTIVE	WR_XFER_ACTIVE
7	6	5	4	3	2	1	0
DAT_7_4				RE_TUNE_REQ	DAT_LINE_ACTIVE	CMD_INHIBIT_DAT	CMD_INHIBIT

Bits	Description	
[31]	**UHS2_IF_DETECT**	**UHS-II Interface Detection** 0x0 (FALSE) = UHS-II interface is not detected 0x1 (TRUE) = UHS-II interface is detected For SD/eMMC mode, this bit always returns 0.
[30]	**LANE_SYNC**	**Lane Synchronization** 0x0 (FALSE) = UHS-II PHY is not initialized 0x1 (TRUE) = UHS-II PHY is initialized For SD/eMMC mode, this bit always returns 0.
[29]	**IN_DORMANT_ST**	**In Dormant Status** 0x0 (FALSE) = Not in DORMANT state 0x1 (TRUE) = In DORMANT state For SD/eMMC mode, this bit always returns 0.
[28]	**SUB_CMD_STAT**	**Sub Command Status** 0x0 (FALSE) = Main Command Status 0x1 (TRUE) = Sub Command Status This bit is used to distinguish between a main command and a sub command status.
[27]	**CMD_ISSUE_ERR**	**Command Not Issued by Error** 0x0 (FALSE) = Host Regulator Voltage is not stable 0x1 (TRUE) = Host Regulator Voltage is stable This bit is set if a command cannot be issued after setting the command register due to an error except the Auto CMD12 error.
[26]	**RSVD_26**	This bit of the PRESENT_ST_R register is reserved bits and always returns 0.
[25]	**HOST_REG_VOL**	**Host Regulator Voltage Stable** This bit is used to check whether the host regulator voltage is stable for switching the voltage of UHS-I mode. This bit reflects the synchronized value of the *host_reg_vol_stable* signal.
[24]	**CMD_LINE_LVL**	**Command-Line Signal Level**

		This bit is used to check the CMD line level to recover from errors and for debugging. These bits reflect the value of the *sd_cmd_in* signal.
[23:20]	**DAT_3_0**	**DAT[3:0] Line Signal Level** This bit is used to check the DAT line level to recover from errors and for debugging. These bits reflect the value of the *sd_dat_in* (lower nibble) signal.
[19]	**WR_PROTECT_SW_LVL**	**Write Protect Switch Pin Level** 0x0 (FALSE) = Write protected 0x1 (TRUE) = Write enabled This bit is supported only for memory and combo cards. This bit reflects the synchronized value of the *card_write_prot* signal.
[18]	**CARD_DETECT_PIN_LEVEL**	**Card Detect Pin Level** 0x0 (FALSE) = No card present 0x1 (TRUE) = Card Present This bit reflects the inverse synchronized value of the *card_detect_n* signal.
[17]	**CARD_STABLE**	**Card Stable** 0x0 (FALSE) = Reset or Debouncing 0x1 (TRUE) = No Card or Inserted This bit indicates the stability of the Card Detect Pin Level. A card is not detected if this bit is set to 1 and the value of the CARD_INSERTED bit is 0.
[16]	**CARD_INSERTED**	**Card Inserted** 0x0 (FALSE) = Reset, Debouncing, or No card 0x1 (TRUE) = Card Inserted This bit indicates whether a card has been inserted. The Host Controller debounces this signal so that Host Driver need not wait for it to stabilize.
[15:12]	**RSVD_15_12**	These bits of the PRESENT_STAT_R register are reserved and always return 0.
[11]	**BUF_RD_ENABLE**	**Buffer Read Enable** 0x0 (DISABLED) = Read disable 0x1 (ENABLED) = Read enable This bit is used for non-DMA transfers. This bit is set if valid data exists in the Host buffer.
[10]	**BUF_WR_ENABLE**	**Buffer Write Enable** 0x0 (DISABLED) = Write disable 0x1 (ENABLED) = Write enable This bit is used for non-DMA transfers. This bit is set if space is available for writing data.
[9]	**RD_XFER_ACTIVE**	**Read Transfer Active** 0x0 (INACTIVE) = No valid data 0x1 (ACTIVE) = Transferring data This bit indicates whether a read transfer is active for SD/eMMC mode.
[8]	**WR_XFER_ACTIVE**	**Write Transfer Active** This status indicates whether a write transfer is active for SD/eMMC mode.
[7:4]	**DAT_7_4**	**DAT[7:4] Line Signal Level** This bit is used to check the DAT line level to recover from errors and for debugging. These bits reflect the value of the *sd_dat_in* (upper nibble) signal.
[3]	**RE_TUNE_REQ**	**Re-Tuning Request** DWC_mshc does not generate retuning request. The software must maintain the Retuning timer.

[2]	**DAT_LINE_ACTIVE**	**DAT Line Active** 0x0 (INACTIVE) = DAT Line Inactive 0x1 (ACTIVE) = DAT Line Active This bit indicates whether one of the DAT lines on the SD/eMMC bus is in use. In the case of read transactions, this bit indicates whether a read transfer is executing on the SD/eMMC bus. In the case of write transactions, this bit indicates whether a write transfer is executing on the SD/eMMC bus. For a command with busy, this status indicates whether the command executing busy is executing on an SD or eMMC bus. **Note:** SD/eMMC Mode only.
[1]	**CMD_INHIBIT_DAT**	**Command Inhibit DAT** 0x0 (READY) = Can issue command which used DAT line 0x1 (NOT_READY) = Cannot issue command which used DAT line This bit is applicable for SD/eMMC mode and is generated if either DAT line active or Read transfer active is set to 1. If this bit is set to 0, it indicates that the Host Controller can issue subsequent SD/eMMC commands.
[0]	**CMD_INHIBIT**	**Command Inhibit** 0x0 (READY) = Host Controller is ready to issue a command 0x1 (NOT_READY) = Host Controller is not ready to issue a command This bit indicates the following : SD/eMMC mode: If this bit is set to 0, it indicates that the CMD line is not in use and the Host controller can issue an SD/eMMC command using the CMD line. This bit is set when the command register is written. This bit is cleared when the command response is received. This bit is not cleared by the response of auto CMD12/23 but cleared by the response of read/write command.

HOST CTRL1 R Register (HOST_CTRL1_R)

Register	Offset	R/W	Description	Reset Value
HOST_CTRL1_R	SDHx_BA+0x028	R/W	Host Control 1 Register	0x00

7	6	5	4	3	2	1	0
CARD_DETECT_SIG_SEL	CARD_DETECT_TEST_LVL	EXT_DAT_XFER	DMA_SEL		HIGH_SPEED_EN	DAT_XFER_WIDTH	LED_CTRL

Bits	Description	
[7]	**CARD_DETECT_SIG_SEL**	**Card Detect Signal Selection** 0x1 (CARD_DT_TEST_LEVEL) = Card Detect Test Level is selected (for test purpose) 0x0 (SDCD_PIN) = SDCD# (card_detect_n signal) is selected (for normal use) This bit selects a source for card detection. When the source for the card detection is switched, the interrupt must be disabled during the switching period.
[6]	**CARD_DETECT_TEST_LVL**	**Card Detect Test Level** 0x1 (CARD_INSERTED) = Card Inserted 0x0 (No_CARD) = No Card This bit is enabled while the Card Detect Signal Selection is set to 1 and it indicates whether a card inserted or not.
[5]	**EXT_DAT_XFER**	**Extended Data Transfer Width** 0x1 (CARD_INSERTED) = Card Inserted 0x0 (No_CARD) = No Card This bit controls 8-bit bus width mode of embedded device.
[4:3]	**DMA_SEL**	**DMA Select** 0x0 (SDMA) = SDMA is selected 0x1 (RSVD_BIT) = Reserved 0x2 (ADMA2) = ADMA2 is selected 0x3 (ADMA2_3) = ADMA2 is selected This field is used to select the DMA type. When Host Version 4 Enable is 1 in Host Control 2 register: 0x0 = SDMA is selected 0x1 = Reserved 0x2 = ADMA2 is selected 0x3 = ADMA2 is selected When Host Version 4 Enable is 0 in Host Control 2 register: 0x0 = SDMA is selected 0x1 = Reserved 0x2 = 32-bit Address ADMA2 is selected 0x3 = 64-bit Address ADMA2 is selected
[2]	**HIGH_SPEED_EN**	**High Speed Enable** 0x1 (HIGH_SPEED) = High Speed mode 0x0 (NORMAL_SPEED) = Normal Speed mode In SD/eMMC mode, this bit is used to determine the selection of preset value for High Speed mode. Before setting this bit, the Host Driver checks the High Speed Support in the Capabilities register. **Note =** DWC_MSHC always outputs the *sd_cmd_out* and *sd_dat_out* lines at the rising edge of *cclk_tx* clock irrespective of this bit. Please refer the section *Connecting the Clock*

		I/O interface in the Mobile Storage Host Controller user guide on clocking requirement for an SD/eMMC card. **Note:** SD/eMMC Mode only.
[1]	**DAT_XFER_WIDTH**	**Data Transfer Width** 0x1 (FOUR_BIT) = 4-bit mode 0x0 (ONE_BIT) = 1-bit mode For SD/eMMC mode, this bit selects the data transfer width of the Host Controller. The Host Driver sets it to match the data width of the SD/eMMC card.
[0]	**LED_CTRL**	**LED Control** 0x0 (OFF) = LED off 0x1 (ON) = LED on This bit is used to caution the user not to remove the card while the SD card is being accessed. The value is reflected on the *led_control* signal. **Note:** This bit is unused, do not set this bit to 0x1.

PWR CTRL R Register (PWR_CTRL_R)

Register	Offset	R/W	Description	Reset Value
PWR_CTRL_R	SDHx_BA+0x029	R/W	Power Control Register	0x00

7	6	5	4	3	2	1	0
SD_BUS_VOL_VDD2			SD_BUS_PWR_VDD2	SD_BUS_VOL_VDD1			SD_BUS_PWR_VDD1

Bits	Description	
[7:5]	**SD_BUS_VOL_VDD2**	**SD Bus Voltage Select for VDD2** This is irrelevant for SD/eMMC card. 0x0 (NO_VDD2) = VDD2 Not Supported 0x1 (RSVD1) = Reserved 0x2 (RSVD2) = Reserved 0x3 (RSVD3) = Reserved 0x4 (V_1_2) = Reserved for 1.2V 0x5 (V_1_8) = 1.8V 0x6 (NOT_USED6) = Not used 0x7 (NOT_USED7) = Not used
[4]	**SD_BUS_PWR_VDD2**	**SD Bus Power for VDD2** This is irrelevant for SD/eMMC card. 0x0 (OFF) = Power off 0x1 (ON) = Power on
[3:1]	**SD_BUS_VOL_VDD1**	**SD Bus Voltage Select for VDD1/eMMC Bus Voltage Select for V_{DD}** 0x7 (V_3_3) = 3.3V (Typ.) 0x6 (V_3_0) = 3.0V (Typ.) 0x5 (V_1_8) = 1.8V (Typ.) for Embedded 0x4 (RSVD4) = Reserved 0x3 (RSVD3) = Reserved 0x2 (RSVD2) = Reserved 0x1 (RSVD1) = Reserved 0x0 (RSVD0) = Reserved These bits enable the Host Driver to select the voltage level for an SD/eMMC card. Before setting this register, the Host Driver checks the Voltage Support bits in the Capabilities register. If an unsupported voltage is selected, the Host System does not supply the SD Bus voltage. The value set in this field is available on the DWC_mshc output signal (sd_vdd1_sel), which is used by the voltage switching circuitry. SD Bus Voltage Select options: 0x7 = 3.3V(Typical) 0x6 = 3.0V(Typical) 0x5 = 1.8V(Typical) for Embedded 0x4 = 0x0 - Reserved eMMC Bus Voltage Select options: 0x7 = 3.3V(Typical) 0x6 = 1.8V(Typical) 0x5 = 1.2V(Typical) 0x4 = 0x0 - Reserved

[0]	**SD_BUS_PWR_VDD1**	**SD Bus Power for VDD1** 0x0 (OFF) = Power off 0x1 (ON) = Power on This bit enables VDD1 power of the card. This setting is available on the sd_vdd1_on output of DWC_mshc so that it can be used to control the VDD1 power supply of the card. Before setting this bit, the SD Host Driver sets the SD Bus Voltage Select bit. If the Host Controller detects a No Card state, this bit is cleared. In SD mode, if this bit is cleared, the Host Controller stops the SD Clock by clearing the SD_CLK_IN bit in the CLK_CTRL_R register.

BGAP CTRL R Register (BGAP_CTRL_R)

Register	Offset	R/W	Description	Reset Value
BGAP_CTRL_R	SDHx_BA+0x02a	R/W	Block Gap Control Register	0x00

7	6	5	4	3	2	1	0
RSVD_7_4				INT_AT_BGAP	RD_WAIT_CTRL	CONTINUE_REQ	STOP_BG_REQ

Bits	Description	
[7:4]	**RSVD_7_4**	These bits of the Block Gap Control register are reserved and always return 0.
[3]	**INT_AT_BGAP**	**Interrupt At Block Gap** 0x0 (DISABLE) = Disabled 0x1 (ENABLE) = Enabled This bit is valid only in the 4-bit mode of an SDIO card and is used to select a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer.
[2]	**RD_WAIT_CTRL**	**Read Wait Control** 0x0 (DISABLE) = Disable Read Wait Control 0x1 (ENABLE) = Enable Read Wait Control This bit is used to enable the read wait protocol to stop read data using DAT[2] line if the card supports read wait. Otherwise, the Host Controller has to stop the card clock to hold the read data.
[1]	**CONTINUE_REQ**	**Continue Request** 0x0 (NO_AFFECT) = No Affect 0x1 (RESTART) = Restart This bit is used to restart the transaction, which was stopped using the Stop At Block Gap Request. The Host Controller automatically clears this bit when the transaction restarts. If stop at block gap request is set to 1, any write to this bit is ignored.
[0]	**STOP_BG_REQ**	**Stop At Block Gap Request** 0x0 (XFER) = Transfer 0x1 (STOP) = Stop This bit is used to stop executing read and write transactions at the next block gap for non-DMA, SDMA, and ADMA transfers.

WUP CTRL R Register (WUP_CTRL_R)

Register	Offset	R/W	Description	Reset Value
WUP_CTRL_R	SDHx_BA+0x02b	R/W	Wakeup Control Register	0x00

7	6	5	4	3	2	1	0
RSVD_7_3					CARD_REMOVAL	CARD_INSERT	CARD_INT

Bits	Description	
[7:3]	**RSVD_7_3**	These bits of Wakeup Control register are reserved and always return 0.
[2]	**CARD_REMOVAL**	**Wakeup Event Enable on SD Card Removal** 0x0 (DISABLED) = Disable 0x1 (ENABLED) = Enable This bit enables wakeup event through Card Removal assertion in the Normal Interrupt Status register. For the SDIO card, Wake Up Support (FN_WUS) in the Card Information Structure (CIS) register does not affect this bit.
[1]	**CARD_INSERT**	**Wakeup Event Enable on SD Card Insertion** 0x0 (DISABLED) = Disable 0x1 (ENABLED) = Enable This bit enables wakeup event through Card Insertion assertion in the Normal Interrupt Status register. FN_WUS (Wake Up Support) in CIS does not affect this bit.
[0]	**CARD_INT**	**Wakeup Event Enable on Card Interrupt** 0x0 (DISABLED) = Disable 0x1 (ENABLED) = Enable This bit enables wakeup event through a Card Interrupt assertion in the Normal Interrupt Status register. This bit can be set to 1 if FN_WUS (Wake Up Support) in CIS is set to 1.

CLK CTRL R Register (CLK_CTRL_R)

Register	Offset	R/W	Description	Reset Value
CLK_CTRL_R	SDHx_BA+0x02c	R/W	Clock Control Register	0x0000

15	14	13	12	11	10	9	8
FREQ_SEL							
7	6	5	4	3	2	1	0
UPPER_FREQ_SEL		CLK_GEN_SELECT	RSVD_4	PLL_ENABLE	SD_CLK_EN	INTERNAL_CLK_STABLE	INTERNAL_CLK_EN

Bits	Description	
[15:8]	**FREQ_SEL**	**SDCLK/RCLK Frequency Select** These bits are used to select the frequency of the SDCLK signal. These bits depend on setting of Preset Value Enable in the Host Control 2 register. If Preset Value Enable = 0, these bits are set by the Host Driver. If Preset Value Enable = 1, these bits are automatically set to a value specified in one of the Preset Value register. The value is reflected on the lower 8-bit of the *card_clk_freq_sel* signal. **10-bit Divided Clock Mode:** 0x3FF = 1/2046 Divided clock ..... N : 1/2N Divided Clock ..... 0x002 = 1/4 Divided Clock 0x001 = 1/2 Divided Clock 0x000 = Base clock
[7:6]	**UPPER_FREQ_SEL**	**These bits specify the upper 2 bits of 10-bit SDCLK/RCLK Frequency Select control The valu**e is reflected on the upper 2 bits of the *card_clk_freq_sel* signal.
[5]	**CLK_GEN_SELECT**	**Clock Generator Select** 0x0 (FALSE) = Divided Clock Mode 0x1 (TRUE) = Programmable Clock Mode This bit is used to select the clock generator mode in SDCLK/RCLK Frequency Select. If Preset Value Enable = 0, this bit is set by the Host Driver. If Preset Value Enable = 1, this bit is automatically set to a value specified in one of the Preset Value registers. The value is reflected on the *card_clk_gen_sel* signal.
[4]	**RSVD_4**	This bit of the CLK_CTRL_R register is reserved and always returns 0.
[3]	**PLL_ENABLE**	**PLL Enable** 0x0 (FALSE): PLL is in low power mode 0x1 (TRUE) = PLL is enabled This bit is used to activate the PLL (applicable when Host Version 4 Enable = 1). When Host Version 4 Enable = 0, INTERNAL_CLK_EN bit may be used to activate PLL. The value is reflected on the *card_clk_en* signal. **Note:** If this bit is not used to to active the PLL when Host Version 4 Enable = 1, it is recommended to set this bit to '1' .
[2]	**SD_CLK_EN**	**SD/eMMC Clock Enable** 0x0 (FALSE) = Disable providing SDCLK/RCLK 0x1 (TRUE) = Enable providing SDCLK/RCLK

		This bit stops the SDCLK or RCLK when set to 0. The SDCLK/RCLK Frequency Select bit can be changed when this bit is set to 0. The value is reflected on the *clk2card_on* pin.
[1]	**INTERNAL_CLK_STABLE**	**Internal Clock Stable** 0x0 (FALSE) = Not Ready 0x1 (TRUE) = Ready This bit enables the Host Driver to check the clock stability twice after the Internal Clock Enable bit is set and after the PLL Enable bit is set. This bit reflects the synchronized value of the *intclk_stable* signal after the Internal Clock Enable bit is set to 1 and also reflects the synchronized value of the *card_clk_stable* signal after the PLL Enable bit is set to 1.
[0]	**INTERNAL_CLK_EN**	**Internal Clock Enable** 0x0 (FALSE) = Stop 0x1 (TRUE) = Oscillate This bit is set to 0 when the Host Driver is not using the Host Controller or the Host Controller awaits a wakeup interrupt. The Host Controller must stop its internal clock to enter a very low power state. However, registers can still be read and written to. The value is reflected on the *intclk_en* signal. **Note:** If this bit is not used to control the internal clock (base clock and master clock), it is recommended to set this bit to '1' .

TOUT CTRL R Register (TOUT_CTRL_R)

Register	Offset	R/W	Description	Reset Value
TOUT_CTRL_R	SDHx_BA+0x02e	R/W	Timeout Control Register	0x00

7	6	5	4	3	2	1	0
RSVD_7_4				TOUT_CNT			

Bits	Description	
[7:4]	**RSVD_7_4**	These bits of the Timeout Control register are reserved and always return 0.
[3:0]	**TOUT_CNT**	**Data Timeout Counter Value** This value determines the interval by which DAT line timeouts are detected. The Timeout clock frequency is generated by dividing the base clock TMCLK value by this value. When setting this register, prevent inadvertent timeout events by clearing the Data Timeout Error Status Enable (in the Error Interrupt Status Enable register). The values for these bits are: 0xF = Reserved 0xE = TMCLK x 2^27 ..... 0x1 = TMCLK x 2^14 0x0 = TMCLK x 2^13 **Note:** During a boot operating in an eMMC mode, an application must configure the boot data timeout value (approximately 1 sec) in this bit.

SW RST R Register (SW_RST_R)

Register	Offset	R/W	Description	Reset Value
SW_RST_R	SDHx_BA+0x02f	R/W	Software Reset Register	0x00

7	6	5	4	3	2	1	0
RSVD_7_3					SW_RST_DAT	SW_RST_CMD	SW_RST_ALL

Bits	Description	
[7:3]	**RSVD_7_3**	These bits of the SW_RST_R register are reserved and always return 0.
[2]	**SW_RST_DAT**	**Software Reset For DAT line** 0x0 (FALSE) = Work 0x1 (TRUE) = Reset This bit is used in SD/eMMC mode and it resets only a part of the data circuit and the DMA circuit is also reset. The following registers and bits are cleared by this bit: **Buffer Data Port register** • Buffer is cleared and initialized. **Present state register** • Buffer Read Enable • Buffer Write Enable • Read Transfer Active • Write Transfer Active • DAT Line Active • Command Inhibit (DAT) **Block Gap Control register** • Continue Request • Stop At Block Gap Request **Normal Interrupt status register** • Buffer Read Ready • Buffer Write Ready • DMA Interrupt • Block Gap Event • Transfer Complete
[1]	**SW_RST_CMD**	**Software Reset For CMD line** 0x0 (FALSE) = Work 0x1 (TRUE) = Reset **This** bit resets only a part of the command circuit to be able to issue a command. This reset is effective only for a command issuing circuit (including response error statuses related to Command Inhibit (CMD) control) and does not affect the data transfer circuit. Host Controller can continue data transfer even after this reset is executed while handling subcommand-response errors. The following registers and bits are cleared by this bit: • Present State register : Command Inhibit (CMD) bit • Normal Interrupt Status register : Command Complete bit • Error Interrupt Status : Response error statuses related to Command Inhibit (CMD) bit
[0]	**SW_RST_ALL**	**Software Reset For All**

		0x0 (FALSE) = Work 0x1 (TRUE) = Reset **This** reset affects the entire Host Controller except for the card detection circuit. During its initialization, the Host Driver sets this bit to 1 to reset the Host Controller. All registers are reset except the capabilities register. If this bit is set to 1, the Host Driver must issue reset command and reinitialize the card.

NORMAL INT STAT R Register (NORMAL_INT_STAT_R)

Register	Offset	R/W	Description	Reset Value
NORMAL_INT_STAT_R	SDHx_BA+0x030	R/W	Normal Interrupt Status Register	0x0000

15	14	13	12	11	10	9	8
ERR_INTERRUPT	CQE_EVENT	FX_EVENT	RE_TUNE_EVENT	INT_C	INT_B	INT_A	CARD_INTERRUPT
7	6	5	4	3	2	1	0
CARD_REMOVAL	CARD_INSERTION	BUF_RD_READY	BUF_WR_READY	DMA_INTERRUPT	BGAP_EVENT	XFER_COMPLETE	CMD_COMPLETE

Bits	Description	
[15]	**ERR_INTERRUPT**	**Error Interrupt** 0x0 (FALSE) = No Error 0x1 (TRUE) = Error If any of the bits in the Error Interrupt Status register are set, then this bit is set.
[14]	**CQE_EVENT**	**Command Queuing Event** 0x0 (FALSE) = No Event 0x1 (TRUE) = Command Queuing Event is detected This status is set if Command Queuing/Crypto related event has occurred in eMMC/SD mode. Read CQHCI's CQIS/CRNQIS register for more details.
[13]	**FX_EVENT**	**FX Event** 0x0 (FALSE) = No Event 0x1 (TRUE) = FX Event is detected This status is set when R[14] of response register is set to 1 and Response Type R1/R5 is set to 0 in Transfer Mode register. This interrupt is used with response check function.
[12]	**RE_TUNE_EVENT**	**Re-tuning Event** This bit is set if the Re-Tuning Request changes from 0 to 1. Re-Tuning request is not supported.
[11]	**INT_C**	**INT_C** This bit is set if INT_C is enabled and if INT_C# pin is in low level. The INT_C# pin is not supported.
[10]	**INT_B**	**INT_B** This bit is set if INT_B is enabled and if INT_B# pin is in low level. The INT_B# pin is not supported.
[9]	**INT_A**	**INT_A** This bit is set if INT_A is enabled and if INT_A# pin is in low level. The INT_A# pin is not supported.
[8]	**CARD_INTERRUPT**	**Card Interrupt** 0x0 (FALSE) = No Card Interrupt 0x1 (TRUE) = Generate Card Interrupt This bit reflects the synchronized value of: • DAT[1] Interrupt Input for SD Mode • DAT[2] Interrupt Input for UHS-II Mode
[7]	**CARD_REMOVAL**	**Card Removal** 0x0 (FALSE) = Card state stable or Debouncing 0x1 (TRUE) = Card Removed This bit is set if the Card Inserted in the Present State register changes from 1 to 0.

[6]	**CARD_INSERTION**	**Card Insertion** 0x0 (FALSE) = Card state stable or Debouncing 0x1 (TRUE) = Card Inserted This bit is set if the Card Inserted in the Present State register changes from 0 to 1.
[5]	**BUF_RD_READY**	**Buffer Read Ready** 0x0 (FALSE) = Not ready to read buffer 0x1 (TRUE) = Ready to read buffer This bit is set if the Buffer Read Enable changes from 0 to 1.
[4]	**BUF_WR_READY**	**Buffer Write Ready** 0x0 (FALSE) = Not ready to write buffer 0x1 (TRUE) = Ready to write buffer This bit is set if the Buffer Write Enable changes from 0 to 1.
[3]	**DMA_INTERRUPT**	**DMA Interrupt** 0x0 (FALSE) = No DMA Interrupt 0x1 (TRUE) = DMA Interrupt is generated This bit is set if the Host Controller detects the SDMA Buffer Boundary during transfer. In case of ADMA, by setting the Int field in the descriptor table, the Host controller generates this interrupt. This interrupt is not generated after a Transfer Complete.
[2]	**BGAP_EVENT**	**Block Gap Event** 0x0 (FALSE) = No Block Gap Event 0x1 (TRUE) = Transaction stopped at block gap This bit is set when both read/write transaction is stopped at block gap due to a Stop at Block Gap Request.
[1]	**XFER_COMPLETE**	**Transfer Complete** 0x0 (FALSE) = Not complete 0x1 (TRUE) = Command execution is completed This bit is set when a read/write transfer and a command with status busy is completed.
[0]	**CMD_COMPLETE**	**Command Complete** 0x0 (FALSE) = No command complete 0x1 (TRUE) = Command Complete In an SD/eMMC Mode, this bit is set when the end bit of a response except for Auto CMD12 and Auto CMD23. This interrupt is not generated when the Response Interrupt Disable in Transfer Mode Register is set to 1.

ERROR INT STAT R Register (ERROR_INT_STAT_R)

Register	Offset	R/W	Description	Reset Value
ERROR_INT_STAT_R	SDHx_BA+0x032	R/W	Error Interrupt Status Register	0x0000

15	14	13	12	11	10	9	8
VENDOR_ERR3	VENDOR_ERR2	VENDOR_ERR1	BOOT_ACK_ERR	RESP_ERR	TUNING_ERR	ADMA_ERR	AUTO_CMD_ERR
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CUR_LMT_ERR	DATA_END_BIT_ERR	DATA_CRC_ERR	DATA_TOUT_ERR	CMD_IDX_ERR	CMD_END_BIT_ERR	CMD_CRC_ERR	CMD_TOUT_ERR

Bits	Description	
[15]	**VENDOR_ERR3**	This bit (VENDOR_ERR3) of the ERROR_INT_STAT_R register is reserved and always returns 0.
[14]	**VENDOR_ERR2**	This bit (VENDOR_ERR2) of the ERROR_INT_STAT_R register is reserved and always returns 0.
[13]	**VENDOR_ERR1**	This bit (VENDOR_ERR1) of the ERROR_INT_STAT_R register is reserved and always returns 0.
[12]	**BOOT_ACK_ERR**	**Boot Acknowledgement Error** This bit is set when there is a timeout for boot acknowledgement or when detecting boot ack status having a value other than 010. This is applicable only when boot acknowledgement is expected in eMMC mode. 0x0 (FALSE) = No error 0x1 (TRUE) = Error
[11]	**RESP_ERR**	**Response Error** Host Controller Version 4.00 supports response error check function to avoid overhead of response error check by Host Driver during DMA execution. If Response Error Check Enable is set to 1 in the Transfer Mode register, Host Controller Checks R1 or R5 response. If an error is detected in a response, this bit is set to 1.This is applicable in SD/eMMC mode. 0x0 (FALSE) = No error 0x1 (TRUE) = Error
[10]	**TUNING_ERR**	**Tuning Error** This bit is set when an unrecoverable error is detected in a tuning circuit except during the tuning procedure (occurrence of an error during tuning procedure is indicated by Sampling Clock Select in the Host Control 2 register). By detecting Tuning Error, Host Driver needs to abort a command executing and perform tuning. To reset tuning circuit, Sampling Clock Select is set to 0 before executing tuning procedure. The Tuning Error is higher priority than the other error interrupts generated during data transfer. By detecting Tuning Error, the Host Driver must discard data transferred by a current read/write command and retry data transfer after the Host Controller retrieved from the tuning circuit error. This is applicable in SD/eMMC mode. 0x0 (FALSE) = No error 0x1 (TRUE) = Error
[9]	**ADMA_ERR**	**ADMA Error** This bit is set when the Host Controller detects error during ADMA-based data transfer. The error could be due to following reasons: Error response received from System bus (Master I/F) ADMA2 Descriptors invalid CQE Task or Transfer descriptors invalid When the error occurs, the state of the ADMA is saved in the ADMA Error Status register. In eMMC CQE mode: The Host Controller generates this Interrupt when it detects an invalid descriptor data (Valid=0) at the ST_FDS state. ADMA Error State in the ADMA Error Status indicates that an error has

		occurred in ST_FDS state. The Host Driver may find that Valid bit is not set at the error descriptor. 0x0 (FALSE) = No error 0x1 (TRUE) = Error
[8]	**AUTO_CMD_ERR**	**Auto CMD Error** This error status is used by Auto CMD12 and Auto CMD23 in SD/eMMC mode. This bit is set when detecting that any of the bits D00 to D05 in Auto CMD Error Status register has changed from 0 to 1. D07 is effective in case of Auto CMD12. Auto CMD Error Status register is valid while this bit is set to 1 and may be cleared by clearing of this bit. 0x0 (FALSE) = No error 0x1 (TRUE) = Error
[7]	**CUR_LMT_ERR**	**Current Limit Error** By setting the SD Bus Power bit in the Power Control register, the Host Controller is requested to supply power for the SD Bus. If the Host Controller supports the Current Limit function, it can be protected from an illegal card by stopping power supply to the card in which case this bit indicates a failure status. A reading of 1 for this bit means that the Host Controller is not supplying power to the SD card due to some failure. A reading of 0 for this bit means that the Host Controller is supplying power and no error has occurred. The Host Controller may require some sampling time to detect the current limit. DWC_mshc Host Controller does not support this function, this bit is always set to 0. 0x0 (FALSE) = No error 0x1 (TRUE) = Power Fail
[6]	**DATA_END_BIT_ERR**	**Data End Bit Error** This error occurs in SD/eMMC mode either when detecting 0 at the end bit position of read data that uses the DAT line or at the end bit position of the CRC status. 0x0 (FALSE) = No error 0x1 (TRUE) = Error
[5]	**DATA_CRC_ERR**	**Data CRC Error** This error occurs in SD/eMMC mode when detecting CRC error when transferring read data which uses the DAT line, when detecting the Write CRC status having a value of other than 010 or when write CRC status timeout. 0x0 (FALSE) = No error 0x1 (TRUE) = Error
[4]	**DATA_TOUT_ERR**	**Data Timeout Error** This bit is set in SD/eMMC mode when detecting one of the following timeout conditions: Busy timeout for R1b, R5b type Busy timeout after Write CRC status Write CRC Status timeout Read Data timeout 0x0 (FALSE) = No error 0x1 (TRUE) = Time out
[3]	**CMD_IDX_ERR**	**Command Index Error** This bit is set if a Command Index error occurs in the command respons in SD/eMMC mode. 0x0 (FALSE) = No error 0x1 (TRUE) = Error
[2]	**CMD_END_BIT_ERR**	**Command End Bit Error** This bit is set when detecting that the end bit of a command response is 0 in SD/eMMC mode. 0x0 (FALSE) = No error 0x1 (TRUE) = End Bit error generated
[1]	**CMD_CRC_ERR**	**Command CRC Error** Command CRC Error is generated in SD/eMMC mode for following two cases.

		If a response is returned and the Command Timeout Error is set to 0 (indicating no timeout), this bit is set to 1 when detecting a CRC error in the command response. The Host Controller detects a CMD line conflict by monitoring the CMD line when a command is issued. If the Host Controller drives the CMD line to 1 level, but detects 0 level on the CMD line at the next SD clock edge, then the Host Controller aborts the command (stop driving CMD line) and set this bit to 1. The Command Timeout Error is also set to 1 to distinguish a CMD line conflict. 0x0 (FALSE) = No error 0x1 (TRUE) = CRC error generated
[0]	**CMD_TOUT_ERR**	**Command Timeout Error** In SD/eMMC Mode, this bit is set only if no response is returned within 64 SD clock cycles from the end bit of the command. If the Host Controller detects a CMD line conflict, along with Command CRC Error bit, this bit is set to 1, without waiting for 64 SD/eMMC card clock cycles. 0x0 (FALSE) = No error 0x1 (TRUE) = Time out

NORMAL INT STAT EN R Register (NORMAL_INT_STAT_EN_R)

Register	Offset	R/W	Description	Reset Value
NORMAL_INT_STAT_EN_R	SDHx_BA+0x034	R/W	Normal Interrupt Status Enable Register	0x0000

15	14	13	12	11	10	9	8
RSVD_15	CQE_EVENT_STAT_EN	FX_EVENT_STAT_EN	RE_TUNE_EVENT_STAT_EN	INT_C_STAT_EN	INT_B_STAT_EN	INT_A_STAT_EN	CARD_INTERRUPT_STAT_EN
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CARD_REMOVAL_STAT_EN	CARD_INSERTION_STAT_EN	BUF_RD_READY_STAT_EN	BUF_WR_READY_STAT_EN	DMA_INTERRUPT_STAT_EN	BGAP_EVENT_STAT_EN	XFER_COMPLETE_STAT_EN	CMD_COMPLETE_STAT_EN

Bits	Description	
[15]	**RSVD_15**	**This bit of the NORMAL_INT_STAT_EN_R register is reserved and always returns 0** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[14]	**CQE_EVENT_STAT_EN**	**CQE Event Status Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[13]	**FX_EVENT_STAT_EN**	**FX Event Status Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled This bit is added from Version 4.10.
[12]	**RE_TUNE_EVENT_STAT_EN**	**Re-Tuning Event Status Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled **Note:** UHS-I only.
[11]	**INT_C_STAT_EN**	**INT_C Status Enable** **If th**is bit is set to 0, the Host Controller clears the interrupt request to the System. The Host Driver may clear this bit before servicing the INT_C and may set this bit again after all interrupt requests to INT_C pin are cleared to prevent inadvertent interrupts. 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[10]	**INT_B_STAT_EN**	**INT_B Status Enable** If this bit is set to 0, the Host Controller clears the interrupt request to the System. The Host Driver may clear this bit before servicing the INT_B and may set this bit again after all interrupt requests to INT_B pin are cleared to prevent inadvertent interrupts. 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[9]	**INT_A_STAT_EN**	**INT_A Status Enable** **If th**is bit is set to 0, the Host Controller clears the interrupt request to the System. The Host Driver may clear this bit before servicing the INT_A and may set this bit again after all interrupt requests to INT_A pin are cleared to prevent inadvertent interrupts. 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[8]	**CARD_INTERRUPT_STAT_EN**	**Card Interrupt Status Enable**

		If this bit is set to 0, the Host Controller clears the interrupt request to the System. The Card Interrupt detection is stopped when this bit is cleared and restarted when this bit is set to 1. The Host Driver may clear the Card Interrupt Status Enable before servicing the Card Interrupt and may set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts. By setting this bit to 0, interrupt input must be masked by implementation so that the interrupt input is not affected by external signal in any state (for example, floating). 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[7]	**CARD_REMOVAL_STAT_EN**	**Card Removal Status Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[6]	**CARD_INSERTION_STAT_EN**	**Card Insertion Status Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[5]	**BUF_RD_READY_STAT_EN**	**Buffer Read Ready Status Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[4]	**BUF_WR_READY_STAT_EN**	**Buffer Write Ready Status Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[3]	**DMA_INTERRUPT_STAT_EN**	**DMA Interrupt Status Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[2]	**BGAP_EVENT_STAT_EN**	**Block Gap Event Status Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[1]	**XFER_COMPLETE_STAT_EN**	**Transfer Complete Status Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[0]	**CMD_COMPLETE_STAT_EN**	**Command Complete Status Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled

ERROR INT STAT EN R Register (ERROR_INT_STAT_EN_R)

Register	Offset	R/W	Description	Reset Value
ERROR_INT_STAT_EN_R	SDHx_BA+0x036	R/W	Error Interrupt Status Enable Register	0x0000

15	14	13	12	11	10	9	8
VENDOR_ERR_STAT_EN3	VENDOR_ERR_STAT_EN2	VENDOR_ERR_STAT_EN1	BOOT_ACK_ERR_STAT_EN	RESP_ERR_STAT_EN	TUNING_ERR_STAT_EN	ADMA_ERR_STAT_EN	AUTO_CMD_ERR_STAT_EN
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CUR_LMT_ERR_STAT_EN	DATA_END_BIT_ERR_STAT_	DATA_CRC_ERR_STAT_EN	DATA_TOUT_ERR_STAT_E	CMD_IDX_ERR_STAT_EN	CMD_END_BIT_ERR_STAT_E	CMD_CRC_ERR_STAT_EN	CMD_TOUT_ERR_STAT_EN

Bits	Description	
[15]	**VENDOR_ERR_STAT_EN3**	**The 15th bit of Error Interrupt Status Enable register is reserved** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[14]	**VENDOR_ERR_STAT_EN2**	**The 14th bit of Error Interrupt Status Enable register is reserved** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[13]	**VENDOR_ERR_STAT_EN1**	**The 13th bit of Error Interrupt Status Enable register is reserved** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[12]	**BOOT_ACK_ERR_STAT_EN**	**Boot Acknowledgment Error** Setting this bit to 1 enables setting of Boot Acknowledgment Error in Error Interrupt Status register (ERROR_INT_STAT_R). 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled **Note:** eMMC Mode only.
[11]	**RESP_ERR_STAT_EN**	**Response Error Status Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled **Note:** SD Mode only.
[10]	**TUNING_ERR_STAT_EN**	**Tuning Error Status Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled **Note:** UHS-I Mode only.
[9]	**ADMA_ERR_STAT_EN**	**ADMA Error Status Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[8]	**AUTO_CMD_ERR_STAT_EN**	**Auto CMD Error Status Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled **Note:** SD/eMMC Mode only.
[7]	**CUR_LMT_ERR_STAT_EN**	**Current Limit Error Status Enable** 0x0 (FALSE) = Masked

		0x1 (TRUE) = Enabled
[6]	**DATA_END_BIT_ERR_STAT_EN**	**Data End Bit Error Status Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled **Note:** SD/eMMC Mode only.
[5]	**DATA_CRC_ERR_STAT_EN**	**Data CRC Error Status Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled **Note:** SD/eMMC Mode only.
[4]	**DATA_TOUT_ERR_STAT_EN**	**Data Timeout Error Status Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled **Note:** SD/eMMC Mode only.
[3]	**CMD_IDX_ERR_STAT_EN**	**Command Index Error Status Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled **Note:** SD/eMMC Mode only.
[2]	**CMD_END_BIT_ERR_STAT_EN**	**Command End Bit Error Status Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled **Note:** SD/eMMC Mode only.
[1]	**CMD_CRC_ERR_STAT_EN**	**Command CRC Error Status Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled **Note:** SD/eMMC Mode only.
[0]	**CMD_TOUT_ERR_STAT_EN**	**Command Timeout Error Status Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled **Note:** SD/eMMC Mode only.

NORMAL INT SIGNAL EN R Register (NORMAL_INT_SIGNAL_EN_R)

Register	Offset	R/W	Description	Reset Value
NORMAL_INT_SIGNAL_EN_R	SDHx_BA+0x038	R/W	Normal Interrupt Signal Enable Register	0x0000

15	14	13	12	11	10	9	8
RSVD_15	CQE_EVENT_SIGNAL_EN	FX_EVENT_SIGNAL_EN	RE_TUNE_EVENT_SIGNAL_	INT_C_SIGNAL_EN	INT_B_SIGNAL_EN	INT_A_SIGNAL_EN	CARD_INTERRUPT_SIGNAL_
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CARD_REMOVAL_SIGNAL_E	CARD_INSERTION_SIGNAL_	BUF_RD_READY_SIGNAL_E	BUF_WR_READY_SIGNAL_E	DMA_INTERRUPT_SIGNAL_	BGAP_EVENT_SIGNAL_EN	XFER_COMPLETE_SIGNAL_	CMD_COMPLETE_SIGNAL_E

Bits	Description	
[15]	**RSVD_15**	This bit of the NORMAL_INT_STAT_R register is reserved and always returns 0.
[14]	**CQE_EVENT_SIGNAL_EN**	**Command Queuing Engine Event Signal Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[13]	**FX_EVENT_SIGNAL_EN**	**FX Event Signal Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[12]	**RE_TUNE_EVENT_SIGNAL_EN**	**Re-Tuning Event Signal Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled **Note:** UHS-I only.
[11]	**INT_C_SIGNAL_EN**	**INT_C Signal Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[10]	**INT_B_SIGNAL_EN**	**INT_B Signal Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[9]	**INT_A_SIGNAL_EN**	**INT_A Signal Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[8]	**CARD_INTERRUPT_SIGNAL_EN**	**Card Interrupt Signal Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[7]	**CARD_REMOVAL_SIGNAL_EN**	**Card Removal Signal Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[6]	**CARD_INSERTION_SIGNAL_EN**	**Card Insertion Signal Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[5]	**BUF_RD_READY_SIGNAL_EN**	**Buffer Read Ready Signal Enable** 0x0 (FALSE) = Masked

		0x1 (TRUE) = Enabled
[4]	**BUF_WR_READY_SIGNAL_EN**	**Buffer Write Ready Signal Enable** 0x0 (FALSE): Masked 0x1 (TRUE) = Enabled
[3]	**DMA_INTERRUPT_SIGNAL_EN**	**DMA Interrupt Signal Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[2]	**BGAP_EVENT_SIGNAL_EN**	**Block Gap Event Signal Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[1]	**XFER_COMPLETE_SIGNAL_EN**	**Transfer Complete Signal Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[0]	**CMD_COMPLETE_SIGNAL_EN**	**Command Complete Signal Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled

ERROR INT SIGNAL EN R Register (ERROR_INT_SIGNAL_EN_R)

Register	Offset	R/W	Description	Reset Value
ERROR_INT_SIGNAL_EN_R	SDHx_BA+0x03a	R/W	Error Interrupt Signal Enable Register	0x0000

15	14	13	12	11	10	9	8
VENDOR_ERR_SIGNAL_EN3	VENDOR_ERR_SIGNAL_EN2	VENDOR_ERR_SIGNAL_EN1	BOOT_ACK_ERR_SIGNAL_E	RESP_ERR_SIGNAL_EN	TUNING_ERR_SIGNAL_EN	ADMA_ERR_SIGNAL_EN	AUTO_CMD_ERR_SIGNAL_E
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CUR_LMT_ERR_SIGNAL_EN	DATA_END_BIT_ERR_SIGNA	DATA_CRC_ERR_SIGNAL_E	DATA_TOUT_ERR_SIGNAL_	CMD_IDX_ERR_SIGNAL_EN	CMD_END_BIT_ERR_SIGNAL	CMD_CRC_ERR_SIGNAL_EN	CMD_TOUT_ERR_SIGNAL_E

Bits	Description	
[15]	**VENDOR_ERR_SIGNAL_EN3**	**The 16th bit of Error Interrupt Signal Enable is reserved** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[14]	**VENDOR_ERR_SIGNAL_EN2**	**The 15th bit of Error Interrupt Signal Enable is reserved** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[13]	**VENDOR_ERR_SIGNAL_EN1**	**The 14th bit of Error Interrupt Signal Enable is reserved** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[12]	**BOOT_ACK_ERR_SIGNAL_EN**	**Boot Acknowledgment Error** Setting this bit to 1 enables generating interrupt signal when Boot Acknowledgement Error in Error Interrupt Status register is set. 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled **Note:** eMMC Mode only.
[11]	**RESP_ERR_SIGNAL_EN**	**Response Error Signal Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled **Note:** SD Mode only.
[10]	**TUNING_ERR_SIGNAL_EN**	**Tuning Error Signal Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled **Note:** UHS-I Mode only.
[9]	**ADMA_ERR_SIGNAL_EN**	**ADMA Error Signal Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled
[8]	**AUTO_CMD_ERR_SIGNAL_EN**	**Auto CMD Error Signal Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled **Note:** SD/eMMC Mode only.
[7]	**CUR_LMT_ERR_SIGNAL_EN**	**Current Limit Error Signal Enable** 0x0 (FALSE) = Masked

		0x1 (TRUE) = Enabled
[6]	**DATA_END_BIT_ERR_SIGNAL_EN**	**Data End Bit Error Signal Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled **Note:** SD/eMMC Mode only.
[5]	**DATA_CRC_ERR_SIGNAL_EN**	**Data CRC Error Signal Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled **Note:** SD/eMMC Mode only.
[4]	**DATA_TOUT_ERR_SIGNAL_EN**	**Data Timeout Error Signal Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled **Note:** SD/eMMC Mode only.
[3]	**CMD_IDX_ERR_SIGNAL_EN**	**Command Index Error Signal Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled **Note:** SD/eMMC Mode only.
[2]	**CMD_END_BIT_ERR_SIGNAL_EN**	**Command End Bit Error Signal Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled **Note:** SD/eMMC Mode only.
[1]	**CMD_CRC_ERR_SIGNAL_EN**	**Command CRC Error Signal Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled **Note:** SD/eMMC Mode only.
[0]	**CMD_TOUT_ERR_SIGNAL_EN**	**Command Timeout Error Signal Enable** 0x0 (FALSE) = Masked 0x1 (TRUE) = Enabled **Note:** SD/eMMC Mode only.

AUTO CMD STAT R Register (AUTO_CMD_STAT_R)

Register	Offset	R/W	Description	Reset Value
AUTO_CMD_STAT_R	SDHx_BA+0x03c	R	Auto CMD Status Register	0x0000

15	14	13	12	11	10	9	8
RSVD_15_8							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CMD_NOT_ISSUED_AUTO_C	RSVD_6	AUTO_CMD_RESP_ERR	AUTO_CMD_IDX_ERR	AUTO_CMD_EBIT_ERR	AUTO_CMD_CRC_ERR	AUTO_CMD_TOUT_ERR	AUTO_CMD12_NOT_EXEC

Bits	Description	
[15:8]	**RSVD_15_8**	These bits of the AUTO_CMD_STAT_R register are reserved and always return 0.
[7]	**CMD_NOT_ISSUED_AUTO_CMD12**	**Command Not Issued By Auto CMD12 Error** If this bit is set to 1, CMD_wo_DAT is not executed due to an Auto CMD12 Error (D04-D01) in this register. This bit is set to 0 when Auto CMD Error is generated by Auto CMD23. 0x1 (TRUE) = Not Issued 0x0 (FALSE) = No Error
[6]	**RSVD_6**	This bit of the AUTO_CMD_STAR_R register is reserved and always returns 0.
[5]	**AUTO_CMD_RESP_ERR**	**Auto CMD Response Error** This bit is set when Response Error Check Enable in the Transfer Mode register is set to 1 and an error is detected in R1 response of either Auto CMD12 or CMD13. This status is ignored if any bit between D00 to D04 is set to 1. 0x1 (TRUE) = Error 0x0 (FALSE) = No Error
[4]	**AUTO_CMD_IDX_ERR**	**Auto CMD Index Error** This bit is set if the command index error occurs in response to a command. 0x1 (TRUE) = Error 0x0 (FALSE) = No Error
[3]	**AUTO_CMD_EBIT_ERR**	**Auto CMD End Bit Error** This bit is set when detecting that the end bit of command response is 0. 0x0 (FALSE) = End Bit Error Generated 0x1 (TRUE) = No Error
[2]	**AUTO_CMD_CRC_ERR**	**Auto CMD CRC Error** This bit is set when detecting a CRC error in the command response. 0x0 (FALSE) = CRC Error Generated 0x1 (TRUE) = No Error
[1]	**AUTO_CMD_TOUT_ERR**	**Auto CMD Timeout Error** This bit is set if no response is returned with 64 SDCLK cycles from the end bit of the command. If this bit is set to 1, error status bits (D04-D01) are meaningless. 0x0 (FALSE) = Time out 0x1 (TRUE) = No Error
[0]	**AUTO_CMD12_NOT_EXEC**	**Auto CMD12 Not Executed**

		If multiple memory block data transfer is not started due to a command error, this bit is not set because it is not necessary to issue an Auto CMD12. Setting this bit to 1 means that the Host Controller cannot issue Auto CMD12 to stop multiple memory block data transfer, due to some error. If this bit is set to 1, error status bits (D04-D01) is meaningless. This bit is set to 0 when Auto CMD Error is generated by Auto CMD23. 0x0 (FALSE) = Not Executed 0x1 (TRUE) = Executed

HOST CTRL2 R Register (HOST_CTRL2_R)

Register	Offset	R/W	Description	Reset Value
HOST_CTRL2_R	SDHx_BA+0x03e	R/W	Host Control 2 Register	0x0000

15	14	13	12	11	10	9	8
PRESET_VAL_ENABLE	ASYNC_INT_ENABLE	ADDRESSING	HOST_VER4_ENABLE	CMD23_ENABLE	ADMA2_LEN_MODE	RSVD_9	UHS2_IF_ENABLE
7	6	5	4	3	2	1	0
SAMPLE_CLK_SEL	EXEC_TUNING	DRV_STRENGTH_SEL		SIGNALING_EN	UHS_MODE_SEL		

Bits	Description	
[15]	**PRESET_VAL_ENABLE**	**Preset Value Enable** This bit enables automatic selection of SDCLK frequency and Driver strength Preset Value registers. When Preset Value Enable is set, SDCLK frequency generation (Frequency Select and Clock Generator Select) and the driver strength selection are performed by the controller. These values are selected from set of Preset Value registers based on selected speed mode. **Note:** For more information, see the FAQ on Preset Register in the DWC_mshc Databook. 0x0 (FALSE) = SDCLK and Driver Strength are controlled by Host Driver 0x1 (TRUE) = Automatic Selection by Preset Value are Enabled
[14]	**ASYNC_INT_ENABLE**	**Asynchronous Interrupt Enable** This bit can be set if a card supports asynchronous interrupts and Asynchronous Interrupt Support is set to 1 in the Capabilities register 0x0 (FALSE) = Disabled 0x1 (TRUE) = Enabled
[13]	**ADDRESSING**	**64-bit Addressing** This bit is effective when Host Version 4 Enable is set to 1. 0x0 (FALSE) = 32 bits addressing 0x1 (TRUE) = 64 bits addressing
[12]	**HOST_VER4_ENABLE**	**Host Version 4 Enable** This bit selects either Version 3.00 compatible mode or Version 4 mode. Functions of following fields are modified for Host Version 4 mode: • SDMA Address: SDMA uses ADMA System Address (05Fh-058h) instead of SDMA System Address register (003h-000h) • ADMA2 selection: DMA select in Host Control 1 register • 64-bit ADMA Descriptor Size: 128-bit descriptor is used instead of 96-bit descriptor when 64-bit Addressing is set to 1 • Selection of 32-bit/64-bit System Addressing: Either 32-bit or 64-bit system addressing is selected by 64-bit Addressing bit in this register • 32-bit Block Count: SDMA System Address register (003h-000h) is modified to 32-bit Block Count register 0x0 (FALSE) = Version 3.00 compatible mode 0x1 (TRUE) = Version 4 mode
[11]	**CMD23_ENABLE**	**CMD23 Enable** If the card supports CMD23, this bit is set to 1. 0x0 (FALSE) = Auto CMD23 is disabled 0x1 (TRUE) = Auto CMD23 is enabled

[10]	**ADMA2_LEN_MODE**	**ADMA2 Length Mode** This bit selects ADMA2 Length mode to be either 16-bit or 26-bit. 0x0 (FALSE) = 16-bit Data Length Mode 0x1 (TRUE) = 26-bit Data Length Mode
[9]	**RSVD_9**	This bit of the HOST_CTRL2_R register is reserved and always returns 0.
[8]	**UHS2_IF_ENABLE**	**UHS-II Interface Enable** This bit is used to enable the UHS-II Interface. The value is reflected on the uhs2_if_en pin. 0x0 (FALSE) = SD/eMMC Interface Enabled 0x1 (TRUE) = UHS-II Interface Enabled **Note:** This Host Controller not supports UHS-II mode, please do not set this bit to 0x1.
[7]	**SAMPLE_CLK_SEL**	**Sampling Clock Select** This bit is used by the Host Controller to select the sampling clock in SD/eMMC mode to receive CMD and DAT. This bit is set by the tuning procedure and is valid after the completion of tuning (when Execute Tuning is cleared). Setting this bit to 1 means that tuning is completed successfully and setting this bit to 0 means that tuning has failed. The value is reflected on the sample_cclk_sel pin. 0x0 (FALSE) = Fixed clock is used to sample data 0x1 (TRUE) = Tuned clock is used to sample data
[6]	**EXEC_TUNING**	**Execute Tuning** This bit is set to 1 to start the tuning procedure in UHS-I/eMMC speed modes and this bit is automatically cleared when tuning procedure is completed.
[5:4]	**DRV_STRENGTH_SEL**	**Driver Strength Select** This bit is used to select the Host Controller output driver in 1.8V signaling UHS-I/eMMC speed modes. The bit depends on setting of Preset Value Enable. The value is reflected on the uhs1_drv_sth pin.
[3]	**SIGNALING_EN**	**1.8V Signaling Enable** This bit controls voltage regulator for I/O cell in UHS-I/eMMC speed modes. Setting this bit from 0 to 1 starts changing the signal voltage from 3.3V to 1.8V. Host Controller clears this bit if switching to 1.8 signaling fails. The value is reflected on the uhs1_swvolt_en pin. **Note:** This bit must be set for all UHS-I speed modes (SDR12/SDR25/SDR50/SDR104/DDR50).
[2:0]	**UHS_MODE_SEL**	**UHS Mode/eMMC Speed Mode Select** These bits are used to select UHS mode in the SD mode of operation. In eMMC mode, these bits are used to select eMMC Speed mode. UHS Mode (SD/UHS-II mode only): 0x0 = SDR12 0x1 = SDR25 0x2 = SDR50 0x3 = SDR104 0x4 = DDR50 0x5 = Reserved 0x6 = Reserved 0x7 = UHS-II eMMC Speed Mode (eMMC mode only): 0x0 = Legacy 0x1 = High Speed SDR 0x2 = Reserved 0x3 = HS200 0x4 = High Speed DDR

		0x5 = Reserved 0x6 = Reserved 0x7 = HS400 **Note:** This Host Controller not supports DDR50 and UHS-II mode(SD/UHS-II mode only), please do not set this bit to 0x4 and 0x7. **Note:** This Host Controller not supports High Speed DDR and HS400 mode(eMMC mode only), please do not set this bit to 0x4 and 0x7.

CAPABILITIES1 R Register (CAPABILITIES1_R)

Register	Offset	R/W	Description	Reset Value
CAPABILITIES1_R	SDHx_BA+0x040	R	Capabilities 1 Register - 0 to 31	0x276E_C898

31	30	29	28	27	26	25	24
SLOT_TYPE_R		ASYNC_INT_SUPPORT	SYS_ADDR_64_V3	SYS_ADDR_64_V4	VOLT_18	VOLT_30	VOLT_33
23	22	21	20	19	18	17	16
SUS_RES_SUPPORT	SDMA_SUPPORT	HIGH_SPEED_SUPPORT	RSVD_20	ADMA2_SUPPORT	Embedded_8_BIT	MAX_BLK_LEN	
15	14	13	12	11	10	9	8
BASE_CLK_FREQ							
7	6	5	4	3	2	1	0
TOUT_CLK_UNIT	RSVD_6	TOUT_CLK_FREQ					

Bits	Description	
[31:30]	**SLOT_TYPE_R**	**Slot Type** These bits indicate usage of a slot by a specific Host System. 0x0 (REMOVABLE_SLOT) = Removable Card Slot 0x1 (EMBEDDED_SLOT) = Embedded Slot for one Device 0x2 (SHARED_SLOT) = Shared Bus Slot (SD mode) 0x3 (UHS2_EMBEDDED_SLOT) = UHS-II Multiple Embedded Devices
[29]	**ASYNC_INT_SUPPORT**	**Asynchronous Interrupt Support** 0x0 (FALSE) = Asynchronous Interrupt Not Supported 0x1 (TRUE) = Asynchronous Interrupt Supported **Note:** SD Mode only.
[28]	**SYS_ADDR_64_V3**	**64-bit System Address Support for V3** This bit sets the Host controller to support 64-bit System Addressing of V3 mode. SDMA cannot be used in 64-bit Addressing in Version 3 Mode. If this bit is set to 1, 64-bit ADMA2 with using 96-bit Descriptor can be enabled by setting Host Version 4 Enable (*HOST_VER4_ENABLE* = 0) and DMA select (*DMA_SEL* = 11b). 0x0 (FALSE) = 64-bit System Address for V3 is Not Supported 0x1 (TRUE) = 64-bit System Address for V3 is Supported
[27]	**SYS_ADDR_64_V4**	**64-bit System Address Support for V4** This bit sets the Host Controller to support 64-bit System Addressing of V4 mode. When this bit is set to 1, full or part of 64-bit address must be used to decode the Host Controller Registers so that Host Controller Registers can be placed above system memory area. 64-bit address decode of Host Controller registers is effective regardless of setting to 64-bit Addressing in Host Control 2. If this bit is set to 1, 64-bit DMA Addressing for version 4 is enabled by setting Host Version 4 Enable (*HOST_VER4_ENABLE* = 1) and by setting 64-bit Addressing (*ADDRESSING* =1) in the Host Control 2 register. SDMA can be used and ADMA2 uses 128-bit Descriptor. 0x0 (FALSE) = 64-bit System Address for V4 is Not Supported 0x1 (TRUE) = 64-bit System Address for V4 is Supported
[26]	**VOLT_18**	**Voltage Support for 1.8V** 0x0 (FALSE) = 1.8V Not Supported

		0x1 (TRUE) = 1.8V Supported
[25]	**VOLT_30**	**Voltage Support for SD 3.0V or Embedded 1.2V** 0x0 (FALSE) = SD 3.0V or Embedded 1.2V Not Supported 0x1 (TRUE) = SD 3.0V or Embedded Supported
[24]	**VOLT_33**	**Voltage Support for 3.3V** 0x0 (FALSE) = 3.3V Not Supported 0x1 (TRUE) = 3.3V Supported
[23]	**SUS_RES_SUPPORT**	**Suspense/Resume Support** This bit indicates whether the Host Controller supports Suspend/Resume functionality. If this bit is 0, the Host Driver does not issue either Suspend or Resume commands because the Suspend and Resume mechanism is not supported. 0x0 (FALSE) = Not Supported 0x1 (TRUE) = Supported
[22]	**SDMA_SUPPORT**	**SDMA Support** This bit indicates whether the Host Controller is capable of using SDMA to transfer data between the system memory and the Host Controller directly. 0x0 (FALSE) = SDMA not Supported 0x1 (TRUE) = SDMA Supported
[21]	**HIGH_SPEED_SUPPORT**	**High Speed Support** This bit indicates whether the Host Controller and the Host System supports High Speed mode and they can supply the SD Clock frequency from 25 MHz to 50 MHz. 0x0 (FALSE) = High Speed not Supported 0x1 (TRUE) = High Speed Supported
[20]	**RSVD_20**	This bit of the CAPABILITIES1_R is a reserved and always returns 0.
[19]	**ADMA2_SUPPORT**	**ADMA2 Support** This bit indicates whether the Host Controller is capable of using ADMA2. 0x0 (FALSE) = ADMA2 not Supported 0x1 (TRUE) = ADMA2 Supported
[18]	**Embedded_8_BIT**	**8-bit Support for Embedded Device** This bit indicates whether the Host Controller is capable of using an 8-bit bus width mode. This bit is not effective when the Slot Type is set to 10b. 0x0 (FALSE) = 8-bit Bus Width not Supported 0x1 (TRUE) = 8-bit Bus Width Supported
[17:16]	**MAX_BLK_LEN**	**Maximum Block Length** This bit indicates the maximum block size that the Host driver can read and write to the buffer in the Host Controller. The buffer transfers this block size without wait cycles. The transfer block length is always 512 bytes for the SD Memory irrespective of this bit 0x0 (ZERO) = 512 Byte 0x1 (ONE) = 1024 Byte 0x2 (TWO) = 2048 Byte 0x3 (THREE) = Reserved
[15:8]	**BASE_CLK_FREQ**	**Base Clock Frequency for SD clock** These bits indicate the base (maximum) clock frequency for the SD Clock. The definition of these bits depend on the Host Controller Version. 6-Bit Base Clock Frequency: This mode is supported by the Host Controller version 1.00 and 2.00. The upper 2 bits are not effective and are always 0. The unit values are 1 MHz. The supported clock range is 10 MHz to 63 MHz. 0x00 = Get information through another method

		0x01 = 1 MHz 0x02 = 2 MHz ..... 0x3F = 63 MHz 0x40-0xFF = Not Supported 8-Bit Base Clock Frequency: This mode is supported by the Host Controller version 3.00. The unit values are 1 MHz. The supported clock range is 10 MHz to 255 MHz. 0x00 = Get information through another method 0x01 = 1 MHz 0x02 = 2 MHz ..... 0xFF = 255 MHz If the frequency is 16.5 MHz, the larger value is set to 0001001b (17 MHz) because the Host Driver uses this value to calculate the clock divider value and it does not exceed the upper limit of the SD Clock frequency. If these bits are all 0, the Host system has to get information using a different method.
[7]	**TOUT_CLK_UNIT**	**Timeout Clock Unit** This bit shows the unit of base clock frequency used to detect Data Timeout Error. 0x0 (kHz) = kHz 0x1 (MHz) = MHz
[6]	**RSVD_6**	This bit of the CAPABILITIES1_R register is reserved and always returns 0.
[5:0]	**TOUT_CLK_FREQ**	**Timeout Clock Frequency** This bit shows the base clock frequency used to detect Data Timeout Error. The Timeout Clock unit defines the unit of timeout clock frequency. It can be kHz or MHz. 0x00 = Get information through another method 0x01 = 1 kHz / 1 MHz 0x02 = 2 kHz / 2 MHz 0x03 = 3 kHz / 3 MHz ..... 0x3F = 63 kHz / 63 MHz

CAPABILITIES2 R Register (CAPABILITIES2_R)

Register	Offset	R/W	Description	Reset Value
CAPABILITIES2_R	SDHx_BA+0x044	R	Capabilities Register - 32 to 63	0x0800_8077

31	30	29	28	27	26	25	24
RSVD_62_63		RSVD_61	VDD2_18V_SUPPORT	Reserved	RSVD_56_58		
23	22	21	20	19	18	17	16
CLK_MUL							
15	14	13	12	11	10	9	8
RE_TUNING_MODES		USE_TUNING_SDR50	RSVD_44	RETUNE_CNT			
7	6	5	4	3	2	1	0
RSVD_39	DRV_TYPED	DRV_TYPEC	DRV_TYPEA	UHS2_SUPPORT	DDR50_SUPPORT	SDR104_SUPPORT	SDR50_SUPPORT

Bits	Description	
[31:30]	**RSVD_62_63**	These bits (RSVD_62_63) of the CAPABILITIES2_R register are reserved bits and always return 0.
[29]	**RSVD_61**	This bit (RSVD_61) of the CAPABILITIES2_R register is reserved and always returns 0.
[28]	**VDD2_18V_SUPPORT**	**1.8V VDD2 Support** This bit indicates support of VDD2 for the Host System. 0x0 (FALSE) = 1.8V VDD2 is not Supported 0x1 (TRUE) = 1.8V VDD2 is Supported
[27]	**Reserved**	Reserved.
[26:24]	**RSVD_56_58**	These bits (RSVD_56_58) of the CAPABILITIES2_R register are reserved and always return 0.
[23:16]	**CLK_MUL**	**Clock Multiplier** These bits indicate the clock multiplier of the programmable clock generator. Setting these bits to 0 means that the Host Controller does not support a programmable clock generator. 0x0 = Clock Multiplier is not Supported 0x1 = Clock Multiplier M = 2. 0x2 = Clock Multiplier M = 3. ..... 0xFF = Clock Multiplier M = 256.
[15:14]	**RE_TUNING_MODES**	**Re-Tuning Modes** These bits select the re-tuning method and limit the maximum data length. **Note:** UHS-I only.
[13]	**USE_TUNING_SDR50**	**Use Tuning for SDR50** **Note:** UHS-I only.
[12]	**RSVD_44**	This bit (RSVD_44) of the CAPABILITIES2_R register is reserved and always returns 0.
[11:8]	**RETUNE_CNT**	**Timer Count for Re-Tuning** 0x0 = Re-Tuning Timer disabled

		0x1 = 1 seconds 0x2 = 2 seconds 0x3 = 4 seconds ..... 0xB = 1024 seconds 0xC = Reserved 0xD = Reserved 0xE = Reserved 0xF = Get information from other source **Note:** UHS-I only.
[7]	**RSVD_39**	This bit (RSVD_39) of the CAPABILITIES2_R register is reserved and always returns 0.
[6]	**DRV_TYPED**	**Driver Type D Support** This bit indicates support of Driver Type D for 1.8 Signaling. **Note:** UHS-I only.
[5]	**DRV_TYPEC**	**Driver Type C Support** This bit indicates support of Driver Type C for 1.8 Signaling. **Note:** UHS-I only.
[4]	**DRV_TYPEA**	**Driver Type A Support** This bit indicates support of Driver Type A for 1.8 Signaling. **Note:** UHS-I only.
[3]	**UHS2_SUPPORT**	**UHS-II Support** This bit indicates whether Host Controller supports UHS-II. **Note:** UHS-I only.
[2]	**DDR50_SUPPORT**	**DDR50 Support** **Note:** UHS-I only.
[1]	**SDR104_SUPPORT**	**SDR104 Support** This bit mentions that SDR104 requires tuning. **Note:** UHS-I only.
[0]	**SDR50_SUPPORT**	**SDR50 Support** This bit indicates that SDR50 is supported. The bit 13 (USE_TUNING_SDR50) indicates whether SDR50 requires tuning or not. **Note:** UHS-I only.

CURR CAPABILITIES1 R Register (CURR_CAPABILITIES1_R)

Register	Offset	R/W	Description	Reset Value
CURR_CAPABILITIES1_R	SDHx_BA+0x048	R	Maximum Current Capabilities Register - 0 to 31	0x0000_0000

31	30	29	28	27	26	25	24
RSVD_31_24							
23	22	21	20	19	18	17	16
MAX_CUR_18V							
15	14	13	12	11	10	9	8
MAX_CUR_30V							
7	6	5	4	3	2	1	0
MAX_CUR_33V							

Bits	Description	
[31:24]	**RSVD_31_24**	These bits of the CURR_CAPABILITIES1_R register are reserved and always return 0.
[23:16]	**MAX_CUR_18V**	**Maximum Current for 1.8V** This bit specifies the Maximum Current for 1.8V VDD1 power supply for the card. 0 = Get information through another method 1 = 4mA 2 = 8mA 3 = 13mA ..... 255 = 1020mA
[15:8]	**MAX_CUR_30V**	**Maximum Current for 3.0V** This bit specifies the Maximum Current for 3.0V VDD1 power supply for the card. 0 = Get information through another method 1 = 4mA 2 = 8mA 3 = 13mA ..... 255 = 1020mA
[7:0]	**MAX_CUR_33V**	**Maximum Current for 3.3V** This bit specifies the Maximum Current for 3.3V VDD1 power supply for the card. 0 = Get information through another method 1 = 4mA 2 = 8mA 3 = 13mA ..... 255 = 1020mA

CURR CAPABILITIES2 R Register (CURR_CAPABILITIES2_R)

Register	Offset	R/W	Description	Reset Value
CURR_CAPABILITIES2_R	SDHx_BA+0x04c	R	Maximum Current Capabilities Register - 32 to 63	0x0000_0000

31	30	29	28	27	26	25	24
RSVD_63_40							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RSVD_63_40							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RSVD_63_40							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MAX_CUR_VDD2_18V							

Bits	Description	
[31:8]	**RSVD_63_40**	These bits of the CURR_CAPABILITIES2_R register are reserved and always return 0.
[7:0]	**MAX_CUR_VDD2_18V**	**Maximum Current for 1.8V VDD2** This bit specifies the Maximum Current for 1.8V VDD2 power supply for the UHS-II card. 0 = Get information through another method 1 = 4mA 2 = 8mA 3 = 13mA ..... 255 = 1020mA

FORCE AUTO CMD STAT R Register (FORCE_AUTO_CMD_STAT_R)

Register	Offset	R/W	Description	Reset Value
FORCE_AUTO_CMD_STAT_R	SDHx_BA+0x050	R/W	Force Event Register for Auto CMD Error Status register	0x0000

15	14	13	12	11	10	9	8
RSVD_15_8							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FORCE_CMD_NOT_ISSUED_	RSVD_6	FORCE_AUTO_CMD_RESP_	FORCE_AUTO_CMD_IDX_ER	FORCE_AUTO_CMD_EBIT_E	FORCE_AUTO_CMD_CRC_E	FORCE_AUTO_CMD_TOUT_	FORCE_AUTO_CMD12_NOT_

Bits	Description	
[15:8]	**RSVD_15_8**	These bits of the FORCE_AUTO_CMD_STAT_R register are reserved and always return 0.
[7]	**FORCE_CMD_NOT_ISSUED_AUTO_CMD12**	**Force Event for Command Not Issued By Auto CMD12 Error** 0x1 (TRUE) = Command Not Issued By Auto CMD12 Error Status is set 0x0 (FALSE) = Not Affected
[6]	**RSVD_6**	This bit of the FORCE_AUTO_CMD_STAT_R register are reserved and always return 0.
[5]	**FORCE_AUTO_CMD_RESP_ERR**	**Force Event for Auto CMD Response Error** 0x1 (TRUE) = Auto CMD Response Error Status is set 0x0 (FALSE) = Not Affected
[4]	**FORCE_AUTO_CMD_IDX_ERR**	**Force Event for Auto CMD Index Error** 0x1 (TRUE) = Auto CMD Index Error Status is set 0x0 (FALSE) = Not Affected
[3]	**FORCE_AUTO_CMD_EBIT_ERR**	**Force Event for Auto CMD End Bit Error** 0x1 (TRUE) = Auto CMD End Bit Error Status is set 0x0 (FALSE) = Not Affected
[2]	**FORCE_AUTO_CMD_CRC_ERR**	**Force Event for Auto CMD CRC Error** 0x1 (TRUE) = Auto CMD CRC Error Status is set 0x0 (FALSE) = Not Affected
[1]	**FORCE_AUTO_CMD_TOUT_ERR**	**Force Event for Auto CMD Timeout Error** 0x1 (TRUE) = Auto CMD Timeout Error Status is set 0x0 (FALSE) = Not Affected
[0]	**FORCE_AUTO_CMD12_NOT_EXEC**	**Force Event for Auto CMD12 Not Executed** 0x1 (TRUE) = Auto CMD12 Not Executed Status is set 0x0 (FALSE) = Not Affected

FORCE ERROR INT STAT R Register (FORCE_ERROR_INT_STAT_R)

Register	Offset	R/W	Description	Reset Value
FORCE_ERROR_INT_STAT_R	SDHx_BA+0x052	R/W	Force Event Register for Error Interrupt Status	0x0000

15	14	13	12	11	10	9	8
FORCE_VENDOR_ERR3	FORCE_VENDOR_ERR2	FORCE_VENDOR_ERR1	FORCE_BOOT_ACK_ERR	FORCE_RESP_ERR	FORCE_TUNING_ERR	FORCE_ADMA_ERR	FORCE_AUTO_CMD_ERR
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FORCE_CUR_LMT_ERR	FORCE_DATA_END_BIT_ER	FORCE_DATA_CRC_ERR	FORCE_DATA_TOUT_ERR	FORCE_CMD_IDX_ERR	FORCE_CMD_END_BIT_ERR	FORCE_CMD_CRC_ERR	FORCE_CMD_TOUT_ERR

Bits	Description	
[15]	**FORCE_VENDOR_ERR3**	This bit (FORCE_VENDOR_ERR3) of the FORCE_ERROR_INT_STAT_R register is reserved and always returns 0.
[14]	**FORCE_VENDOR_ERR2**	This bit (FORCE_VENDOR_ERR2) of the FORCE_ERROR_INT_STAT_R register is reserved and always returns 0.
[13]	**FORCE_VENDOR_ERR1**	This bit (FORCE_VENDOR_ERR1) of the FORCE_ERROR_INT_STAT_R register is reserved and always returns 0.
[12]	**FORCE_BOOT_ACK_ERR**	**Force Event for Boot Ack error** 0x0 (FALSE) = Not Affected 0x1 (TRUE) = Boot ack Error Status is set
[11]	**FORCE_RESP_ERR**	**Force Event for Response Error** 0x0 (FALSE) = Not Affected 0x1 (TRUE) = Response Error Status is set **Note:** SD Mode only.
[10]	**FORCE_TUNING_ERR**	**Force Event for Tuning Error** 0x0 (FALSE) = Not Affected 0x1 (TRUE) = Tuning Error Status is set **Note:** UHS-I Mode only.
[9]	**FORCE_ADMA_ERR**	**Force Event for ADMA Error** 0x0 (FALSE) = Not Affected 0x1 (TRUE) = ADMA Error Status is set
[8]	**FORCE_AUTO_CMD_ERR**	**Force Event for Auto CMD Error** 0x0 (FALSE) = Not Affected 0x1 (TRUE) = Auto CMD Error Status is set **Note:** SD/eMMC Mode only.
[7]	**FORCE_CUR_LMT_ERR**	**Force Event for Current Limit Error** 0x0 (FALSE) = Not Affected 0x1 (TRUE) = Current Limit Error Status is set
[6]	**FORCE_DATA_END_BIT_ERR**	**Force Event for Data End Bit Error** 0x0 (FALSE) = Not Affected 0x1 (TRUE) = Data End Bit Error Status is set **Note:** SD/eMMC Mode only.
[5]	**FORCE_DATA_CRC_ERR**	**Force Event for Data CRC Error**

		0x0 (FALSE) = Not Affected 0x1 (TRUE) = Data CRC Error Status is set **Note:** SD/eMMC Mode only.
[4]	**FORCE_DATA_TOUT_ERR**	**Force Event for Data Timeout Error** 0x0 (FALSE) = Not Affected 0x1 (TRUE) = Data Timeout Error Status is set **Note:** SD/eMMC Mode only.
[3]	**FORCE_CMD_IDX_ERR**	**Force Event for Command Index Error** 0x0 (FALSE) = Not Affected 0x1 (TRUE) = Command Index Error Status is set **Note:** SD/eMMC Mode only.
[2]	**FORCE_CMD_END_BIT_ERR**	**Force Event for Command End Bit Error** 0x0 (FALSE) = Not Affected 0x1 (TRUE) = Command End Bit Error Status is set **Note:** SD/eMMC Mode only.
[1]	**FORCE_CMD_CRC_ERR**	**Force Event for Command CRC Error** 0x0 (FALSE) = Not Affected 0x1 (TRUE) = Command CRC Error Status is set **Note:** SD/eMMC Mode only.
[0]	**FORCE_CMD_TOUT_ERR**	**Force Event for Command Timeout Error** 0x0 (FALSE) = Not Affected 0x1 (TRUE) = Command Timeout Error Status is set **Note:** SD/eMMC Mode only.

ADMA ERR STAT R Register (ADMA_ERR_STAT_R)

Register	Offset	R/W	Description	Reset Value
ADMA_ERR_STAT_R	SDHx_BA+0x054	R	ADMA Error Status Register	0x00

7	6	5	4	3	2	1	0
RSVD_7_3					ADMA_LEN_ERR	ADMA_ERR_STATES	

Bits	Description	
[7:3]	**RSVD_7_3**	These bits of the ADMA_ERR_STAT_R register are reserved and always return 0.
[2]	**ADMA_LEN_ERR**	**ADMA Length Mismatch Error States** 0x0 (NO_ERR) = No Error 0x1 (ERROR) = Error This error occurs in the following instances: • While the Block Count Enable is being set, the total data length specified by the Descriptor table is different from that specified by the Block Count and Block Length • When the total data length cannot be divided by the block length
[1:0]	**ADMA_ERR_STATES**	**ADMA Error States** 0x0 (ST_STOP) = Stop DMA - SYS_ADR register points to a location next to the error descriptor 0x1 (ST_FDS) = Fetch Descriptor - SYS_ADR register points to the error descriptor 0x2 (UNUSED) = Never set this state 0x3 (ST_TFR) = Transfer Data - SYS_ADR register points to a location next to the error descriptor These bits indicate the state of ADMA when an error occurs during ADMA data transfer.

ADMA SA LOW R Register (ADMA_SA_LOW_R)

Register	Offset	R/W	Description	Reset Value
ADMA_SA_LOW_R	SDHx_BA+0x058	R/W	ADMA System Address Register - Low	0x0000_0000

31	30	29	28	27	26	25	24
ADMA_SA_LOW							
23	22	21	20	19	18	17	16
ADMA_SA_LOW							
15	14	13	12	11	10	9	8
ADMA_SA_LOW							
7	6	5	4	3	2	1	0
ADMA_SA_LOW							

Bits	Description	
[31:0]	ADMA_SA_LOW	**ADMA System Address** These bits indicate the lower 32 bits of the ADMA system address. • SDMA = If Host Version 4 Enable is set to 1, this register stores the system address of the data location • ADMA2 = This register stores the byte address of the executing command of the descriptor table

PRESET INIT R Register (PRESET_INIT_R)

Register	Offset	R/W	Description	Reset Value
PRESET_INIT_R	SDHx_BA+0x060	R	Preset Value for Initialization	0x0000

15	14	13	12	11	10	9	8
DRV_SEL_VAL		RSVD_13_11			CLK_GEN_SEL_VAL	FREQ_SEL_VAL	
7	6	5	4	3	2	1	0
FREQ_SEL_VAL							

Bits	Description	
[15:14]	**DRV_SEL_VAL**	**Driver Strength Select Value** 0x0 (TYPEB) = Driver Type B is selected 0x1 (TYPEA) = Driver Type A is selected 0x2 (TYPEC) = Driver Type C is selected 0x3 (TYPED) = Driver Type D is selected These bits indicate that the Driver strength is supported by 1.8V signaling bus speed modes. These bits are meaningless for 3.3V signaling.
[13:11]	**RSVD_13_11**	These bits of the PRESET_INIT_R register are reserved and always return 0.
[10]	**CLK_GEN_SEL_VAL**	**Clock Generator Select Value** 0x0 (FALSE) = Host Controller Ver2.0 Compatible Clock Generator 0x1 (PROG) = Programmable Clock Generator This bit is effective when the Host Controller supports a programmable clock generator.
[9:0]	**FREQ_SEL_VAL**	**SDCLK/RCLK Frequency Select Value** 10-bit preset value to be set in SDCLK/RCLK Frequency Select field of the Clock Control register described by a Host System.

PRESET DS R Register (PRESET_DS_R)

Register	Offset	R/W	Description	Reset Value
PRESET_DS_R	SDHx_BA+0x062	R/W	Preset Value for Default Speed	0x0000

15	14	13	12	11	10	9	8
DRV_SEL_VAL		RSVD_13_11			CLK_GEN_SEL_VAL	FREQ_SEL_VAL	
7	6	5	4	3	2	1	0
FREQ_SEL_VAL							

Bits	Description	
[15:14]	**DRV_SEL_VAL**	**Driver Strength Select Value** 0x0 (TYPEB) = Driver Type B is selected 0x1 (TYPEA) = Driver Type A is selected 0x2 (TYPEC) = Driver Type C is selected 0x3 (TYPED) = Driver Type D is selected These bits indicate the Driver strength value supported by 1.8V signaling bus speed modes. This field is meaningless for the Default speed mode as it uses 3.3V signaling.
[13:11]	**RSVD_13_11**	These bits of the PRESET_DS_R register are reserved and always return 0.
[10]	**CLK_GEN_SEL_VAL**	**Clock Generator Select Value** 0x0 (FALSE) = Host Controller Ver2.0 Compatible Clock Generator 0x1 (PROG) = Programmable Clock Generator This bit is effective when Host Controller supports programmable clock generator.
[9:0]	**FREQ_SEL_VAL**	**SDCLK/RCLK Frequency Select Value** 10-bit preset value to be set in SDCLK/RCLK Frequency Select field of the Clock Control register described by a Host System.

PRESET HS R Register (PRESET_HS_R)

Register	Offset	R/W	Description	Reset Value
PRESET_HS_R	SDHx_BA+0x064	R/W	Preset Value for High Speed	0x0000

15	14	13	12	11	10	9	8
DRV_SEL_VAL		RSVD_13_11			CLK_GEN_SEL_VAL	FREQ_SEL_VAL	
7	6	5	4	3	2	1	0
FREQ_SEL_VAL							

Bits	Description	
[15:14]	**DRV_SEL_VAL**	**Driver Strength Select Value** 0x0 (TYPEB) = Driver Type B is selected 0x1 (TYPEA) = Driver Type A is selected 0x2 (TYPEC) = Driver Type C is selected 0x3 (TYPED) = Driver Type D is selected These bits indicate the Driver strength value supported by 1.8V signaling bus speed modes. This field is meaningless for High speed mode as it uses 3.3V signaling.
[13:11]	**RSVD_13_11**	These bits of the PRESET_HS_R register are reserved and always return 0.
[10]	**CLK_GEN_SEL_VAL**	**Clock Generator Select Value** 0x0 (FALSE) = Host Controller Ver2.0 Compatible Clock Generator 0x1 (PROG) = Programmable Clock Generator This bit is effective when Host Controller supports programmable clock generator.
[9:0]	**FREQ_SEL_VAL**	**SDCLK/RCLK Frequency Select Value** 10-bit preset value to be set in SDCLK/RCLK Frequency Select field of the Clock Control register described by a Host System.

PRESET SDR12 R Register (PRESET_SDR12_R)

Register	Offset	R/W	Description	Reset Value
PRESET_SDR12_R	SDHx_BA+0x066	R/W	Preset Value for SDR12	0x0000

15	14	13	12	11	10	9	8
DRV_SEL_VAL		RSVD_13_11			CLK_GEN_SEL_VAL	FREQ_SEL_VAL	
7	6	5	4	3	2	1	0
FREQ_SEL_VAL							

Bits	Description	
[15:14]	**DRV_SEL_VAL**	**Driver Strength Select Value** 0x0 (TYPEB) = Driver Type B is selected 0x1 (TYPEA) = Driver Type A is selected 0x2 (TYPEC) = Driver Type C is selected 0x3 (TYPED) = Driver Type D is selected These bits indicate the Driver strength value supported for the SDR12 bus speed mode. These bits are meaningless for 3.3V signaling.
[13:11]	**RSVD_13_11**	These bits of the PRESET_SDR12_R register are reserved and always return 0.
[10]	**CLK_GEN_SEL_VAL**	**Clock Generator Select Value** 0x0 (FALSE) = Host Controller Ver2.0 Compatible Clock Generator 0x1 (PROG) = Programmable Clock Generator This bit is effective when Host Controller supports programmable clock generato
[9:0]	**FREQ_SEL_VAL**	**SDCLK/RCLK Frequency Select Value** 10-bit preset value to be set in SDCLK/RCLK Frequency Select field of the Clock Control register described by a Host System.

PRESET SDR25 R Register (PRESET_SDR25_R)

Register	Offset	R/W	Description	Reset Value
PRESET_SDR25_R	SDHx_BA+0x068	R/W	Preset Value for SDR25	0x0000

15	14	13	12	11	10	9	8
DRV_SEL_VAL		RSVD_13_11			CLK_GEN_SEL_VAL	FREQ_SEL_VAL	
7	6	5	4	3	2	1	0
FREQ_SEL_VAL							

Bits	Description	
[15:14]	**DRV_SEL_VAL**	**Driver Strength Select Value** 0x0 (TYPEB) = Driver Type B is selected 0x1 (TYPEA) = Driver Type A is selected 0x2 (TYPEC) = Driver Type C is selected 0x3 (TYPED) = Driver Type D is selected These bits indicate the Driver strength value supported for the SDR25 bus speed mode. These bits are meaningless for 3.3V signaling.
[13:11]	**RSVD_13_11**	These bits of the PRESET_SDR25_R register are reserved and always return 0.
[10]	**CLK_GEN_SEL_VAL**	**Clock Generator Select Value** 0x0 (FALSE) = Host Controller Ver2.0 Compatible Clock Generator 0x1 (PROG) = Programmable Clock Generator This bit is effective when Host Controller supports programmable clock generator.
[9:0]	**FREQ_SEL_VAL**	**SDCLK/RCLK Frequency Select Value** 10-bit preset value to be set in SDCLK/RCLK Frequency Select field of the Clock Control register described by a Host System.

PRESET SDR50 R Register (PRESET_SDR50_R)

Register	Offset	R/W	Description	Reset Value
PRESET_SDR50_R	SDHx_BA+0x06a	R/W	Preset Value for SDR50	0x0000

15	14	13	12	11	10	9	8
DRV_SEL_VAL		RSVD_13_11			CLK_GEN_SEL_VAL	FREQ_SEL_VAL	
7	6	5	4	3	2	1	0
FREQ_SEL_VAL							

Bits	Description	
[15:14]	**DRV_SEL_VAL**	**Driver Strength Select Value** 0x0 (TYPEB) = Driver Type B is selected 0x1 (TYPEA) = Driver Type A is selected 0x2 (TYPEC) = Driver Type C is selected 0x3 (TYPED) = Driver Type D is selected These bits indicate the Driver strength value supported for the SDR50 bus speed mode. These bits are meaningless for 3.3V signaling.
[13:11]	**RSVD_13_11**	These bits of the PRESET_SDR50_R register are reserved and always return 0.
[10]	**CLK_GEN_SEL_VAL**	**Clock Generator Select Value** This bit is effective when Host Controller supports programmable clock generator
[9:0]	**FREQ_SEL_VAL**	**SDCLK/RCLK Frequency Select Value** 0x0 (FALSE) = Host Controller Ver2.0 Compatible Clock Generator 0x1 (PROG) = Programmable Clock Generator 10-bit preset value to be set in SDCLK/RCLK Frequency Select field of the Clock Control register described by a Host System.

RESET SDR104 R Register (PRESET_SDR104_R)

Register	Offset	R/W	Description	Reset Value
PRESET_SDR104_R	SDHx_BA+0x06c	R/W	Preset Value for SDR104	0x0000

15	14	13	12	11	10	9	8
DRV_SEL_VAL		RSVD_13_11			CLK_GEN_SEL_VAL	FREQ_SEL_VAL	
7	6	5	4	3	2	1	0
FREQ_SEL_VAL							

Bits	Description	
[15:14]	**DRV_SEL_VAL**	**Driver Strength Select Value** 0x0 (TYPEB) = Driver Type B is selected 0x1 (TYPEA) = Driver Type A is selected 0x2 (TYPEC) = Driver Type C is selected 0x3 (TYPED) = Driver Type D is selected These bits indicate the Driver strength value supported for the SDR104 bus speed mode. These bits are meaningless for 3.3V signaling.
[13:11]	**RSVD_13_11**	These bits of the PRESET_SDR104_R register are reserved and always return 0.
[10]	**CLK_GEN_SEL_VAL**	**Clock Generator Select Value** 0x0 (FALSE) = Host Controller Ver2.0 Compatible Clock Generator 0x1 (PROG) = Programmable Clock Generator This bit is effective when Host Controller supports programmable clock generator.
[9:0]	**FREQ_SEL_VAL**	**SDCLK/RCLK Frequency Select Value** 10-bit preset value to be set in SDCLK/RCLK Frequency Select field of the Clock Control register described by a Host System.

PRESET DDR50 R Register (PRESET_DDR50_R)

Register	Offset	R/W	Description	Reset Value
PRESET_DDR50_R	SDHx_BA+0x06e	R	Preset Value for DDR50	0x0000

15	14	13	12	11	10	9	8
DRV_SEL_VAL		RSVD_13_11			CLK_GEN_SEL_VAL	FREQ_SEL_VAL	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FREQ_SEL_VAL							

Bits	Description	
[15:14]	**DRV_SEL_VAL**	**Driver Strength Select Value** 0x0 (TYPEB) = Driver Type B is selected 0x1 (TYPEA) = Driver Type A is selected 0x2 (TYPEC) = Driver Type C is selected 0x3 (TYPED) = Driver Type D is selected These bits indicate Driver strength value supported for DDR50 bus speed mode. These bits are meaningless for 3.3V signaling.
[13:11]	**RSVD_13_11**	These bits of the PRESET_DDR50_R register are reserved and always return 0.
[10]	**CLK_GEN_SEL_VAL**	**Clock Generator Select Value** 0x0 (FALSE) = Host Controller Ver2.0 Compatible Clock Generator 0x1 (PROG) = Programmable Clock Generator This bit is effective when Host Controller supports programmable clock generator.
[9:0]	**FREQ_SEL_VAL**	**SDCLK/RCLK Frequency Select Value** 10-bit preset value to be set in SDCLK/RCLK Frequency Select field of the Clock Control register described by a Host System.

PRESET UHS2 R Register (PRESET_UHS2_R)

Register	Offset	R/W	Description	Reset Value
PRESET_UHS2_R	SDHx_BA+0x074	R/W	Preset Value for UHS-II	0x0000

15	14	13	12	11	10	9	8
DRV_SEL_VAL		RSVD_13_11			CLK_GEN_SEL_VAL	FREQ_SEL_VAL	
7	6	5	4	3	2	1	0
FREQ_SEL_VAL							

Bits	Description	
[15:14]	**DRV_SEL_VAL**	**Driver Strength Select Value** 0x0 (TYPEB) = Driver Type B is selected 0x1 (TYPEA) = Driver Type A is selected 0x2 (TYPEC) = Driver Type C is selected 0x3 (TYPED) = Driver Type D is selected These bits indicate Driver strength value supported for UHS2 bus speed mode. These bits are meaningless for 3.3V signaling.
[13:11]	**RSVD_13_11**	These bits of the PRESET_UHS2_R register are reserved and always return 0.
[10]	**CLK_GEN_SEL_VAL**	**Clock Generator Select Value** 0x0 (FALSE) = Host Controller Ver2.0 Compatible Clock Generator 0x1 (PROG) = Programmable Clock Generator This Host Controller not supports programmable clock generator, please do not set this bit to 0x1.
[9:0]	**FREQ_SEL_VAL**	**SDCLK/RCLK Frequency Select Value** These bits specify a 10-bit preset value that must be set in the SDCLK/RCLK Frequency Select field of the Clock Control register described by a Host System.

P EMBEDDED CNTRL Register (P_EMBEDDED_CNTRL)

Register	Offset	R/W	Description	Reset Value
P_EMBEDDED_CNTRL	SDHx_BA+0xe6	R	Pointer for Embedded Control	0x0F6C

15	14	13	12	11	10	9	8
Reserved				**REG_OFFSET_ADDR**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
REG_OFFSET_ADDR							

Bits	Description	
[15:12]	**Reserved**	These bits of the P_EMBEDDED_CNTRL register are reserved and always return 0.
[11:0]	**REG_OFFSET_ADDR**	**Offset Address of Embedded Control register.**

P VENDOR SPECIFIC AREA Register (P_VENDOR_SPECIFIC_AREA)

Register	Offset	R/W	Description	Reset Value
P_VENDOR_SPECIFIC_AREA	SDHx_BA+0xe8	R	Pointer for Vendor Specific Area 1	0x0500

15	14	13	12	11	10	9	8
Reserved				**REG_OFFSET_ADDR**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
REG_OFFSET_ADDR							

Bits	Description	
[15:12]	**Reserved**	These bits of the P_VENDOR_SPECIFIC_AREA register are reserved and always return 0.
[11:0]	**REG_OFFSET_ADDR**	**Base offset Address for Vendor-Specific registers.**

P VENDOR2 SPECIFIC AREA Register (P_VENDOR2_SPECIFIC_AREA)

Register	Offset	R/W	Description	Reset Value
P_VENDOR2_SPECIFIC_AREA	SDHx_BA+0xea	R	Pointer for Vendor Specific Area 2	0x0180

15	14	13	12	11	10	9	8
REG_OFFSET_ADDR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
REG_OFFSET_ADDR							

Bits	Description	
[15:0]	**REG_OFFSET_ADDR**	**Base offset Address for Command Queuing registers.**

SLOT INTR STATUS R Register (SLOT_INTR_STATUS_R)

Register	Offset	R/W	Description	Reset Value
SLOT_INTR_STATUS_R	SDHx_BA+0xfc	R	Slot Interrupt Status Register	0x0000

15	14	13	12	11	10	9	8
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INTR_SLOT							

Bits	Description	
[15:8]	**Reserved**	These bits of the SLOT_INTR_STATUS_R register are reserved and always return 0.
[7:0]	**INTR_SLOT**	**Interrupt signal for each Slot** These status bits indicate the logical OR of Interrupt signal and Wakeup signal for each slot. A maximum of 8 slots can be defined. If one interrupt signal is associated with multiple slots, the Host Driver can identify the interrupt that is generated by reading these bits. By a power on reset or by setting Software Reset For All bit, the interrupt signals are de-asserted and this status reads 00h. Bit 00 = Slot 1 Bit 01 = Slot 2 Bit 02 = Slot 3 .... Bit 07 = Slot 8 **Note:** MSHC Host Controller support single card slot. This register shall always return 0.

HOST CNTRL VERS R Register (HOST_CNTRL_VERS_R)

Register	Offset	R/W	Description	Reset Value
HOST_CNTRL_VERS_R	SDHx_BA+0xfe	R	Host Controller Version	0x0005

15	14	13	12	11	10	9	8
VENDOR_VERSION_NUM							
7	6	5	4	3	2	1	0
SPEC_VERSION_NUM							

Bits	Description	
[15:8]	VENDOR_VERSION_NUM	**Vendor Version Number** This field is reserved for the vendor version number. Host Driver must not use this status.
[7:0]	SPEC_VERSION_NUM	**Specification Version Number** 0x0 (VER_1_00) = SD Host Controller Specification Version 1.00 0x1 (VER_2_00) = SD Host Controller Specification Version 2.00 0x2 (VER_3_00) = SD Host Controller Specification Version 3.00 0x3 (VER_4_00) = SD Host Controller Specification Version 4.00 0x4 (VER_4_10) = SD Host Controller Specification Version 4.10 0x5 (VER_4_20) = SD Host Controller Specification Version 4.20 These bits indicate the Host controller specification version. The upper and lower 4-bits indicate the version. Values 0x06-0xFF are reserved.

EMBEDDED CTRL R Register (EMBEDDED_CTRL_R)

Register	Offset	R/W	Description	Reset Value
EMBEDDED_CTRL_R	SDHx_BA+0xf6c	R/W	Embedded Control register	0x0000_000

31	30	29	28	27	26	25	24
RSVD_31	BACK_END_PWR_CTRL						
23	22	21	20	19	18	17	16
RSVD_23	INT_PIN_SEL			RSVD_19	CLK_PIN_SEL		
15	14	13	12	11	10	9	8
RSVD_15	BUS_WIDTH_PRESET						
7	6	5	4	3	2	1	0
RSVD_7_6		NUM_INT_PIN		RSVD_3	NUM_CLK_PIN		

Bits	Description	
[31]	**RSVD_31**	This bit (RSVD_31) of the EMBEDDED_CTRL_R register is reserved and always returns 0.
[30:24]	**BACK_END_PWR_CTRL**	**Back-End Power Control** Each bit of this field controls back-end power supply for an embedded device. 0 = Back-End Power is off 1 = Back-End Power is supplied D24 = Back-End Power for Device 1 D25 = Back-End Power for Device 2 D26 = Back-End Power for Device 3 D27 = Back-End Power for Device 4 D28 = Back-End Power for Device 5 D29 = Back-End Power for Device 6 D30 = Back-End Power for Device 7 **Note: SD Mode**
[23]	**RSVD_23**	This bit (RSVD_23) of the EMBEDDED_CTRL_R register is reserved and always returns 0.
[22:20]	**INT_PIN_SEL**	**Interrupt Pin Select** These bits enable the interrupt pin inputs. 000 = Interrupts (INT_A,INT_B,INT_C) are disabled xx1 = INT_A is enabled x1x = INT_B is enabled 1xx = INT_C is enabled
[19]	**RSVD_19**	This bit (RSVD_19) of the EMBEDDED_CTRL_R register is reserved and always returns 0.
[18:16]	**CLK_PIN_SEL**	**Clock Pin Select** This bit is selected by one of clock pin outputs. 0x0 = Clock pins are disabled 0x1 = CLK[1] is selected 0x2 = CLK[2] is selected . .

		. . . . 0x7 = CLK[7] is selected **Note:** SD Mode.
[15]	**RSVD_15**	This bit (RSVD_15) of the EMBEDDED_CTRL_R register is reserved and always returns 0.
[14:8]	**BUS_WIDTH_PRESET**	**Bus Width Preset** Each bit of this field specifies the bus width for each embedded device. The shared bus supports mixing of 4-bit and 8-bit bus width devices. D08 = Bus Width Preset for Device 1 D09 = Bus Width Preset for Device 2 D10 = Bus Width Preset for Device 3 D11 = Bus Width Preset for Device 4 D12 = Bus Width Preset for Device 5 D13 = Bus Width Preset for Device 6 D14 = Bus Width Preset for Device 7 Function of each bit is defined as follows: 0 = 4-bit bus width mode 1 = 8-bit bus width mode **Note:** SD Mode.
[7:6]	**RSVD_7_6**	These bits (RSVD_7_6) of the EMBEDDED_CTRL_R register are reserved and always return 0.
[5:4]	**NUM_INT_PIN**	**Number of Interrupt Input Pins** This field indicates support of interrupt input pins for an embedded system.
[3]	**RSVD_3**	This bit (RSVD_3) of the EMBEDDED_CTRL_R register is reserved and always returns 0.
[2:0]	**NUM_CLK_PIN**	**Number of Clock Pins** This field indicates support of clock pins to select one of devices for shared bus system. Up to 7 clock pins can be supported. 0x0 = Shared bus is not supported 0x1 = 1 SDCLK is supported 0x2 - 2 SDCLK is supported . . . . . . 0x7 = 7 SDCLK is supported **Note:** SD Mode.

MSHC VER ID R Register (MSHC_VER_ID_R)

Register	Offset	R/W	Description	Reset Value
MSHC_VER_ID_R	SDHx_BA+0x500	R	MSHC version	0x3138_302A

31	30	29	28	27	26	25	24
MSHC_VER_ID							
23	22	21	20	19	18	17	16
MSHC_VER_ID							
15	14	13	12	11	10	9	8
MSHC_VER_ID							
7	6	5	4	3	2	1	0
MSHC_VER_ID							

Bits	Description	
[31:0]	**MSHC_VER_ID**	**Current release number** This field indicates the Synopsys DesignWare Cores DWC_mshc/DWC_mshc_lite current release number that is read by an application. An application reading this register in conjunction with the MSHC_VER_TYPE_R register, gathers details of the current release.

MSHC VER TYPE R Register (MSHC_VER_TYPE_R)

Register	Offset	R/W	Description	Reset Value
MSHC_VER_TYPE_R	SDHx_BA+0x504	R	MSHC version type	0x6761_2A2A

31	30	29	28	27	26	25	24
MSHC_VER_TYPE							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
MSHC_VER_TYPE							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
MSHC_VER_TYPE							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MSHC_VER_TYPE							

Bits	Description	
[31:0]	**MSHC_VER_TYPE**	**Current release type** This field indicates the Synopsys DesignWare Cores DWC_mshc/DWC_mshc_lite current release type that is read by an application. An application reading this register in conjunction with the MSHC_VER_ID_R register, gathers details of the current release.

MSHC CTRL R Register (MSHC_CTRL_R)

Register	Offset	R/W	Description	Reset Value
MSHC_CTRL_R	SDHx_BA+0x508	R/W	MSHC Control register	0x01

7	6	5	4	3	2	1	0
Reserved			SW_CG_DIS	RSVD1			CMD_CONFLICT_CHECK

Bits	Description	
[7:5]	**Reserved**	Reserved.
[4]	**SW_CG_DIS**	**Internal clock gating disable control** 0x0 (ENABLE) = Internal clock gates are active and clock gating is controlled internally 0x1 (DISABLE) = Internal clock gating is disabled, clocks are not gated internally This bit must be used to disable IP's internal clock gating when required. when disabled clocks are not gated. Clocks to the core (except hclk) must be stopped when programming this bit.
[3:1]	**RSVD1**	These bits (RSVD1) of the MSHC_CTRL_R register are reserved and always return 0.
[0]	**CMD_CONFLICT_CHECK**	**Command conflict check** 0x0 (DISABLE_CMD_CONFLICT_CHK) = Disable command conflict check 0x1 (CMD_CONFLICT_CHK_LAT1) = Check for command conflict after 1 card clock cycle This bit enables command conflict check. **Note:** DWC_mshc controller monitors the CMD line whenever a command is issued and checks whether the value driven on sd_cmd_out matches the value on sd_cmd_in at next subsequent edge of cclk_tx to determine command conflict error. This bit is cleared only if the feed back delay (including I/O Pad delay) is more than (t_card_clk_period - t_setup), where t_setup is the setup time of a flop in DWC_mshc. The I/O pad delay is consistent across CMD and DATA lines, and it is within the value: (2*t_card_clk_period - t_setup)

MBIU CTRL R Register (MBIU_CTRL_R)

Register	Offset	R/W	Description	Reset Value
MBIU_CTRL_R	SDHx_BA+0x510	R/W	MBIU Control register	0x0F

7	6	5	4	3	2	1	0
RSVD				**BURST_INCR16_EN**	**BURST_INCR8_EN**	**BURST_INCR4_EN**	**UNDEFL_INCR_EN**

Bits	Description	
[7:4]	**RSVD**	Reserved.
[3]	**BURST_INCR16_EN**	**INCR16 Burst** 0x0 (FALSE) = AHB INCR16 burst type is not generated on Master I/F 0x1 (TRUE) = AHB INCR16 burst type can be generated on Master I/F Controls generation of INCR16 transfers on Master interface.
[2]	**BURST_INCR8_EN**	**INCR8 Burst** 0x0 (FALSE) = AHB INCR8 burst type is not generated on Master I/F 0x1 (TRUE) = AHB INCR8 burst type can be generated on Master I/F Controls generation of INCR8 transfers on Master interface.
[1]	**BURST_INCR4_EN**	**INCR4 Burst** 0x0 (FALSE) = AHB INCR4 burst type is not generated on Master I/F 0x1 (TRUE) = AHB INCR4 burst type can be generated on Master I/F Controls generation of INCR4 transfers on Master interface.
[0]	**UNDEFL_INCR_EN**	**Undefined INCR Burst** 0x0 (FALSE) = Undefined INCR type burst is the least preferred burst on AHB Master I/F 0x1 (TRUE) = Undefined INCR type burst is the most preferred burst on AHB Master I/F Controls generation of undefined length INCR transfer on Master interface.

EMMC CTRL R Register (EMMC_CTRL_R)

Register	Offset	R/W	Description	Reset Value
EMMC_CTRL_R	SDHx_BA+0x52c	R/W	eMMC Control register	0x000C

15	14	13	12	11	10	9	8
RSVD					Reserved		
7	6	5	4	3	2	1	0
Reserved				EMMC_RST_N_OE	EMMC_RST_N	DISABLE_DATA_CRC_CHK	CARD_IS_EMMC

Bits	Description	
[15:11]	**RSVD**	These bits (RSVD) of the EMMC_CTRL_R register are reserved and always return 0.
[10:4]	**Reserved**	Reserved.
[3]	**EMMC_RST_N_OE**	**Output Enable control for EMMC Device Reset signal PAD control** 0x1 (ENABLE) = sd_rst_n_oe is 1 0x0 (DISABLE) = sd_rst_n_oe is 0 This field drived sd_rst_n_oe output of DWC_mshc
[2]	**EMMC_RST_N**	**EMMC Device Reset signal control** 0x1 (RST_DEASSERT) = Reset to eMMC device is deasserted 0x0 (RST_ASSERT) = Reset to eMMC device asserted (active low) This register field controls the sd_rst_n output of DWC_mshc
[1]	**DISABLE_DATA_CRC_CHK**	**Disable Data CRC Check** 0x1 (DISABLE) = DATA CRC check is disabled 0x0 (ENABLE) = DATA CRC check is enabled This bit controls masking of CRC16 error for Card Write in eMMC mode. This is useful in bus testing (CMD19) for an eMMC device. In bus testing, an eMMC card does not send CRC status for a block, which may generate CRC error. This CRC error can be masked using this bit during bus testing.
[0]	**CARD_IS_EMMC**	**eMMC Card present** 0x1 (EMMC_CARD) = Card connected to MSHC is an eMMC card 0x0 (NON_EMMC_CARD) = Card connected to MSHC is a non-eMMC card This bit indicates the type of card connected. An application program this bit based on the card connected to MSHC.

BOOT CTRL R Register (BOOT_CTRL_R)

Register	Offset	R/W	Description	Reset Value
BOOT_CTRL_R	SDHx_BA+0x52e	R/W	eMMC Boot Control register	0x0000

15	14	13	12	11	10	9	8
BOOT_TOUT_CNT				RSVD_11_9			BOOT_ACK_ENABLE
7	6	5	4	3	2	1	0
VALIDATE_BOOT	RSVD_6_1						MAN_BOOT_EN

Bits	Description	
[15:12]	**BOOT_TOUT_CNT**	**Boot Ack Timeout Counter Value** This value determines the interval by which boot ack timeout (50 ms) is detected when boot ack is expected during boot operation. 0xF = Reserved 0xE = TMCLK x 2^27 .. - ..... 0x1 = TMCLK x 2^14 0x0 = TMCLK x 2^13
[11:9]	**RSVD_11_9**	These bits (RSVD_11_9) of the BOOT_CTRL_R register are reserved and always return 0.
[8]	**BOOT_ACK_ENABLE**	**Boot Acknowledge Enable** 0x1 (TRUE) = Boot Ack enable 0x0 (FALSE) = Boot Ack disable When this bit set, DWC_mshc checks for boot acknowledge start pattern of 0-1-0 during boot operation. This bit is applicable for both mandatory and alternate boot mode.
[7]	**VALIDATE_BOOT**	**Validate Mandatory Boot Enable bit** 0x1 (TRUE) = Validate Mandatory boot enable bit 0x0 (FALSE) = Ignore Mandatory boot Enable bit This bit is used to validate the MAN_BOOT_EN bit.
[6:1]	**RSVD_6_1**	These bits (RSVD _6_1) of the BOOT_CTRL_R register are reserved and always return 0.
[0]	**MAN_BOOT_EN**	**Mandatory Boot Enable** This bit is used to initiate the mandatory boot operation. The application sets this bit along with VALIDATE_BOOT bit. Writing 0 is ignored. The DWC_mshc clears this bit after the boot transfer is completed or terminated. 0x1 (MAN_BOOT_EN) = Mandatory boot enable 0x0 (MAN_BOOT_DIS) = Mandatory boot disable Testable = read only

AT CTRL R Register (AT_CTRL_R)

Register	Offset	R/W	Description	Reset Value
AT_CTRL_R	SDHx_BA+0x540	R/W	Tuning and Auto-tuning control register	0x0300_0005

31	30	29	28	27	26	25	24
Reserved				SWIN_TH_VAL			
23	22	21	20	19	18	17	16
Reserved			POST_CHANGE_DLY		PRE_CHANGE_DLY		TUNE_CLK_STOP_EN
15	14	13	12	11	10	9	8
RSDV3				WIN_EDGE_SEL			
7	6	5	4	3	2	1	0
RSDV2			SW_TUNE_EN	RPT_TUNE_ERR	SWIN_TH_EN	CI_SEL	AT_EN

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:24]	**SWIN_TH_VAL**	**Sampling window threshold value setting** The maximum value that can be set here depends on the length of delayline used for tuning. A delayLine with 128 taps can use values from 0x0 to 0x7F. This field is valid only when SWIN_TH_EN is '1'. Should be programmed only when SAMPLE_CLK_SEL is '0' 0x0 = Threshold values is 0x1, windows of length 1 tap and above can be selected as sampling window. 0x1 = Threshold values is 0x2, windows of length 2 taps and above can be selected as sampling window. 0x2 = Threshold values is 0x1, windows of length 3 taps and above can be selected as sampling window. ..... 0x7F = Threshold values is 0x1, windows of length 127 taps and above can be selected as sampling window.
[23:21]	**Reserved**	Reserved.
[20:19]	**POST_CHANGE_DLY**	**Time taken for phase switching and stable clock output** Specifies the maximum time (in terms of cclk cycles) that the delay line can take to switch its output phase after a change in tuning_cclk_sel or autotuning_cclk_sel. 0x0 (LATENCY_LT_1) = Less than 1-cycle latency 0x1 (LATENCY_LT_2) = Less than 2-cycle latency 0x2 (LATENCY_LT_3) = Less than 3-cycle latency 0x3 (LATENCY_LT_4) = Less than 4-cycle latency
[18:17]	**PRE_CHANGE_DLY**	**Maximum Latency specification between cclk_tx and cclk_rx** 0x0 (LATENCY_LT_1) = Less than 1-cycle latency 0x1 (LATENCY_LT_2) = Less than 2-cycle latency 0x2 (LATENCY_LT_3) = Less than 3-cycle latency 0x3 (LATENCY_LT_4) = Less than 4-cycle latency
[16]	**TUNE_CLK_STOP_EN**	**Clock stopping control for Tuning and auto-tuning circuit** When enabled, clock gate control output of DWC_mshc (clk2card_on) is pulled low before

		changing phase select codes on tuning_cclk_sel and autotuning_cclk_sel. This effectively stops the Device/Card clock, cclk_rx and also drift_cclk_rx. Changing phase code when clocks are stopped ensures glitch free phase switching. Set this bit to 0 if the PHY or delayline can guarantee glitch free switching. 0x1 (ENABLE_CLK_STOPPING) = Clocks stopped during phase code change 0x0 (DISABLE_CLK_STOPPING) = Clocks not stopped. PHY ensures glitch free phase switching.
[15:12]	**RSDV3**	These bits (RSVD3) of the AT_CTRL_R register are reserved and always return 0.
[11:8]	**WIN_EDGE_SEL**	This field sets the phase for Left and Right edges for drift monitoring [Left edge offset + Right edge offset] must not be less than total taps of delayLine. 0x0 = User selection disabled. Tuning calculated edges are used. 0x1 = Right edge Phase is center + 2 stages, Left edge Phase is center - 2 stages. 0x2 = Right edge Phase is center + 3 stages, Left edge Phase is center - 3 stages. ... 0xF = Right edge Phase is center + 16 stages, Left edge Phase is center - 16 stages.
[7:5]	**RSDV2**	These bits (RSVD2) of the AT_CTRL_R register are reserved and always return 0.
[4]	**SW_TUNE_EN**	**Enable software-managed tuning flow** 0x1 (SW_TUNING_ENABLE) = Software-managed tuning enabled. AT_STAT_R.CENTER_PH_CODE Field is now writable. 0x0 (SW_TUNING_DISABLE) = Software-managed tuning disabled.
[3]	**RPT_TUNE_ERR**	Framing errors are not generated when executing tuning. This debug bit allows users to report these errors. 0x1 (DEBUG_ERRORS) = Debug mode for reporting framing errors 0x0 (ERRORS_DISABLED) = Default mode where as per SD-HCI no errors are reported.
[2]	**SWIN_TH_EN**	**Sampling window Threshold enable** Selects the tuning mode Field should be programmed only when SAMPLE_CLK_SEL is '0' 0x1 (THRESHOLD_MODE) = Tuning engine selects the first complete sampling window that meets the threshold set by SWIN_TH_VAL field 0x0 (LARGEST_WIN_MODE) = Tuning engine sweeps all taps and settles at the largest window
[1]	**CI_SEL**	Select the interval when the corrected center phase select code can be driven on tuning_cclk_sel output. 0x0 (WHEN_IN_BLK_GAP) = Driven in block gap interval 0x1 (WHEN_IN_IDLE) = Driven at the end of the transfer
[0]	**AT_EN**	Setting this bit enables Auto tuning engine This bit is enabled by default when core is configured with mode3 retuning support Clear this bit to 0 when core is configured to have Mode3 re-tuning but SW wishes to disable mode3 re-tuning This field should be programmed only when CLK_CTRL_R.SD_CLK_EN is 0. 0x1 (AT_ENABLE) = AutoTuning is enabled 0x0 (AT_DISABLE) = AutoTuning is disabled

AT STAT R Register (AT_STAT_R)

Register	Offset	R/W	Description	Reset Value
AT_STAT_R	SDHx_BA+0x544	R/W	Tuning and Auto-tuning status register	0x0000_0006

31	30	29	28	27	26	25	24
RSDV1							
23	22	21	20	19	18	17	16
L_EDGE_PH_CODE							
15	14	13	12	11	10	9	8
R_EDGE_PH_CODE							
7	6	5	4	3	2	1	0
CENTER_PH_CODE							

Bits	Description	
[31:24]	**RSDV1**	These bits of the AT_STAT_R register are reserved and always return 0.
[23:16]	**L_EDGE_PH_CODE**	**Left Edge Phase code** Reading this field returns the phase code value used by Auto-tuning engine to sample data on Left edge of sampling window.
[15:8]	**R_EDGE_PH_CODE**	**Right Edge Phase code** Reading this field returns the phase code value used by Auto-tuning engine to sample data on Right edge of sampling window.
[7:0]	**CENTER_PH_CODE**	**Centered Phase code** Reading this field returns the current value on tuning_cclk_sel output Setting AT_CTRL_R.SW_TUNE_EN enables software to write to this field and its contents are reflected on tuning_cclk_sel. Testable = read only

CQCAP Register (CQCAP)

Register	Offset	R/W	Description	Reset Value
CQCAP	SDHx_BA+0x184	R	Command Queuing Capabilities register	0x0000_30C8

31	30	29	28	27	26	25	24
CQCCAP_RSVD3			CRYPTO_SUPPORT	CQCCAP_RSVD2			
23	22	21	20	19	18	17	16
CQCCAP_RSVD2							
15	14	13	12	11	10	9	8
ITCFMUL				CQCCAP_RSVD1		ITCFVAL	
7	6	5	4	3	2	1	0
ITCFVAL							

Bits	Description	
[31:29]	**CQCCAP_RSVD3**	These bits [31:29] of the CQCAP register are reserved and always return 0.
[28]	**CRYPTO_SUPPORT**	**Crypto Support** 0x0 (FALSE) = Crypto not Supported 0x1 (TRUE) = Crypto Supported This bit indicates whether the Host Controller supports cryptographic operations.
[27:16]	**CQCCAP_RSVD2**	These bits [27:16] of the CQCAP register are reserved and always return 0.
[15:12]	**ITCFMUL**	**Internal Timer Clock Frequency Multiplier** This field indicates the frequency of the clock used for interrupt coalescing timer and for determining the SQS polling period. See ITCFVAL definition for details. Values 0x5 to 0xF are reserved.
[11:10]	**CQCCAP_RSVD1**	These bits of the CQCAP register are reserved and always return 0.
[9:0]	**ITCFVAL**	**Internal Timer Clock Frequency Value** 0x0 (CLK_1 kHz) = 1 kHz clock 0x1 (CLK_10 kHz) = 10 kHz clock 0x2 (CLK_100 kHz) = 100 kHz clock 0x3 (CLK_1 MHz) = 1 MHz clock 0x4 (CLK_10 MHz) = 10 MHz clock This field scales the frequency of the timer clock provided by ITCFMUL. The Final clock frequency of actual timer clock is calculated as ITCFVAL* ITCFMUL.

6.36 NAND Flash Interface (NFI)

6.36.1 Overview

The NAND Flash Interface of this chip has DMA unit and NAND Flash unit. The DMA unit provides a DMA (Direct Memory Access) function for NFI to exchange data between system memory (e.g. SDRAM) and shared buffer (128 bytes), and the NAND Flash unit control the interface of NAND Flash. The interface controller can support NAND-type Flash and the NFI is cooperated with DMAC to provide a fast data transfer between system memory and cards.

6.36.2 Features

- Supports single DMA channel and address in non-word boundary
- Supports hardware Scatter-Gather function
- Supports 128 bytes shared buffer for data exchange between system memory and Flash device (Separated into two 64 bytes ping-pong FIFO)
- Supports SLC and MLC NAND type Flash
- Adjustable NAND page sizes. (2048B+spare area, 4096B+spare area and 8192B+spare area)
- Supports up to 8-bit/12-bit /24-bit hardware ECC calculation circuit to protect data communication
- Supports programmable NAND timing cycle
- Supports EDO mode

6.36.3 Block Diagram

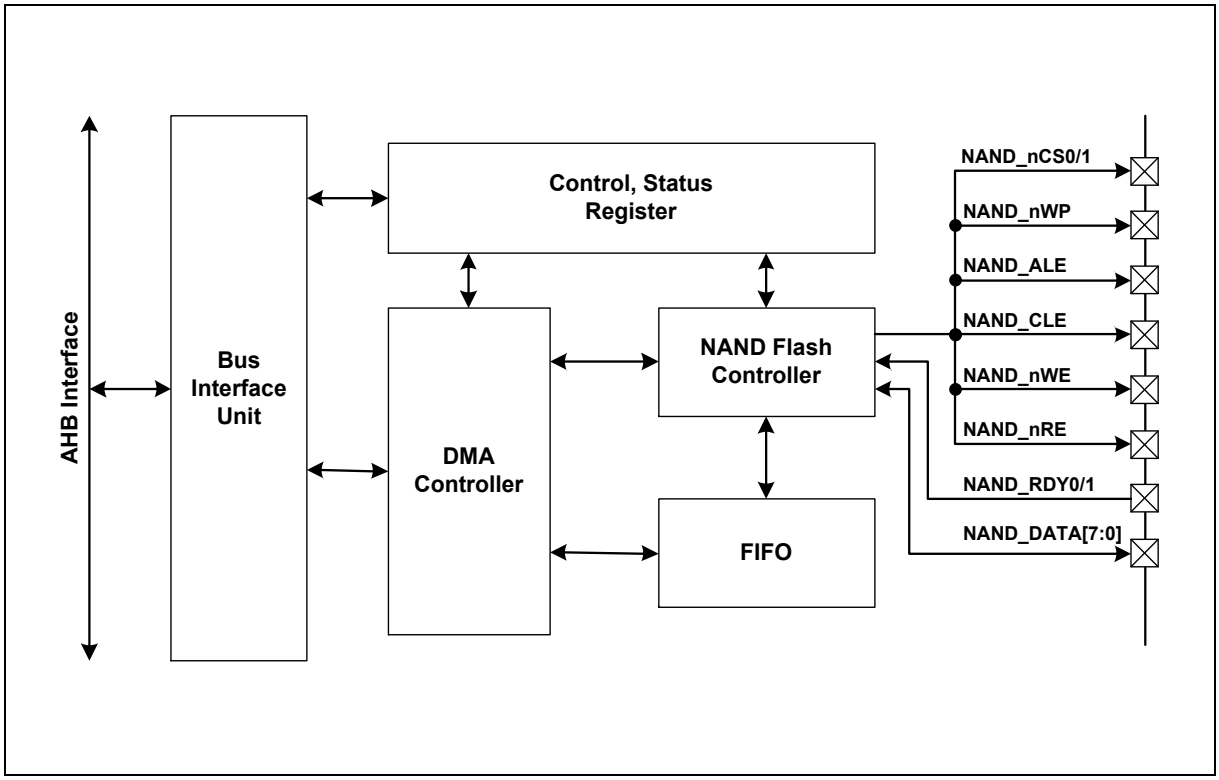


Figure 6.36-1 NFI Block Diagram

6.36.4 Basic Configuration

6.36.4.1 NAND Flash Basic Configuration

- Clock Source Configuration
  - Set NANDEN (CLK_SYSCLK0[18]) high to enable clock for NAND Flash controller operation.
- Reset Configuration
  - Reset NAND Flash controller in NANDRST (SYS_IPRST0[26]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
NAND	NAND_DATA0	PA.0	MFP6
	NAND_DATA1	PA.1	MFP6
	NAND_DATA2	PA.2	MFP6
	NAND_DATA3	PA.3	MFP6
	NAND_DATA4	PA.4	MFP6
	NAND_DATA5	PA.5	MFP6
	NAND_DATA6	PA.6	MFP6
	NAND_DATA7	PA.7	MFP6
	NAND_RDY0	PA.8	MFP6
	NAND_RDY1	PG.11	MFP6
	NAND_nRE	PA.9	MFP6
	NAND_nWE	PA.10	MFP6
	NAND_CLE	PA.11	MFP6
	NAND_ALE	PA.12	MFP6
	NAND_nCS0	PA.13	MFP6
	NAND_nCS1	PG.14	MFP6
	NAND_nWP	PA.14	MFP6

Table 6.36-1 NAND Flash Pin Configuration

6.36.5 Functional Description

6.36.5.1 DMA Controller (DMAC)

The DMAC provides a DMA (Direct Memory Access) function for NFI controller to exchange data between system memory (e.g. SDRAM) and shared buffer (128 bytes). Arbitration of DMA request between NFI is done by DMAC's bus master. User only simply fills in the starting address and enables DMAC, and then lets DMAC to handle the data transfer automatically.

There is a 128 bytes shared buffer inside DMAC, separate into two 64 bytes ping-pong FIFO (total 128 bytes). It can provide multi-block transfers using ping-pong mechanism for NFI. When NFI is not busy, these shared buffers can be accessed directly by software.

6.36.5.2 NAND Flash Controller (NFC)

The NAND Flash Interface supports NAND-type Flash. NFI is cooperation with DMAC to provide a fast data transfer between system memory and cards. There is a single 128 bytes buffer embedded in DMAC for temporary data storage (separated into two 64 bytes ping-pong FIFO). Since DMAC only has a single channel, this means only one interface can be active at one time.

6.36.5.3 NAND Flash

The NFI provides an interface for NAND-type Flash access. It supports 2048 bytes/page, 4096 bytes/page and 8192 bytes/page NAND. This NAND-type Flash controller provides all required signals for NAND Flash, including NAND_RDY0/1, NAND_nCS0/1, NAND_CLE, NAND_ALE, NAND_nWE, NAND_nRE and data pins NAND_DATA0 to NAND_DATA7.

The NAND Flash controller provides direct command port, address port and data port to control NAND Flash signals manually. When command port written, NAND Flash controller generates appropriate signal to NAND. When address port written without setting EOA (NFI_NANDADDR[31])] high, NAND Flash controller generate an address cycle to NAND, but do not clear ALE until the last address cycle is written through address port with setting EOA (NFI_NANDADDR[31]) high. By using this method, address cycle can be generated by software arbitrarily.

For example, if user wants to write 4 address cycles to NAND, it is necessary to write 3 addresses to the address port without setting EOA (NFI_NANDADDR[31])] high, and then write the last one address to the address port with setting EOA (NFI_NANDADDR[31])] high.

The NAND Flash controller also provides a status and an interrupt flag of NAND_RDY0/1 pin. The interrupt flag will be set only when rising edge encountered on NAND_RDY0/1 pin.

The NFI supports three different page sizes, 2048 bytes/page, 4096 bytes/page and 8192 bytes/page. Use PSIZE (NFI_NANDCTL[17:16]) to select the NAND Flash type. Using DMA function for data transfer can increase the performance. For different model of NAND, it is necessary to adjust the timing parameter at NFI_NANDTMCTL to meet specification of NAND Flash device. Adjust timing parameter can also improve data transfer performance.

NAND Flash controller equips a BCH algorithm for error recovery. The BCH algorithm can correct up to 8 bits errors, 12 bits errors or 24 bits errors. By reading ECCFLDIF (NFI_NANDINTSTS[2]) to check the error occurrence while by reading NFI_NANDECCES0, NFI_NANDECCES1, NFI_NANDECCES2 and NFI_NANDECCES3 to know how many errors and if those errors are correctable or not. If those errors are correctable, please read NFI_NANDECCEAx and NFI_NANDECCEDx to correct the errors manually.

For 2K/4K/8K Page size NAND Flash with BCH algorithm, T can be t8, t12 or t24. Based on the page size and T setting, NFI generate different size of parity data. The number of byte for parity data in different page size and T setting listed in the Table 6.36-2 shown below. The data arrangement of redundant area is as Figure 6.36-2, Figure 6.36-3 and Figure 6.36-4 shown below.

It is recommended to choose appropriate T based on NAND Flash page size and redundant area size.

BCH Algorithm		Parity (Byte) 2048 Page Size	Parity (Byte) 4096 Page Size	Parity (Byte) 8192 Page Size
BCH T8		60	120	240
BCH T12		92	184	368
BCH T24		90	180	360

Table 6.36-2 Number of Parity (Byte) for Each BCH Algorithm

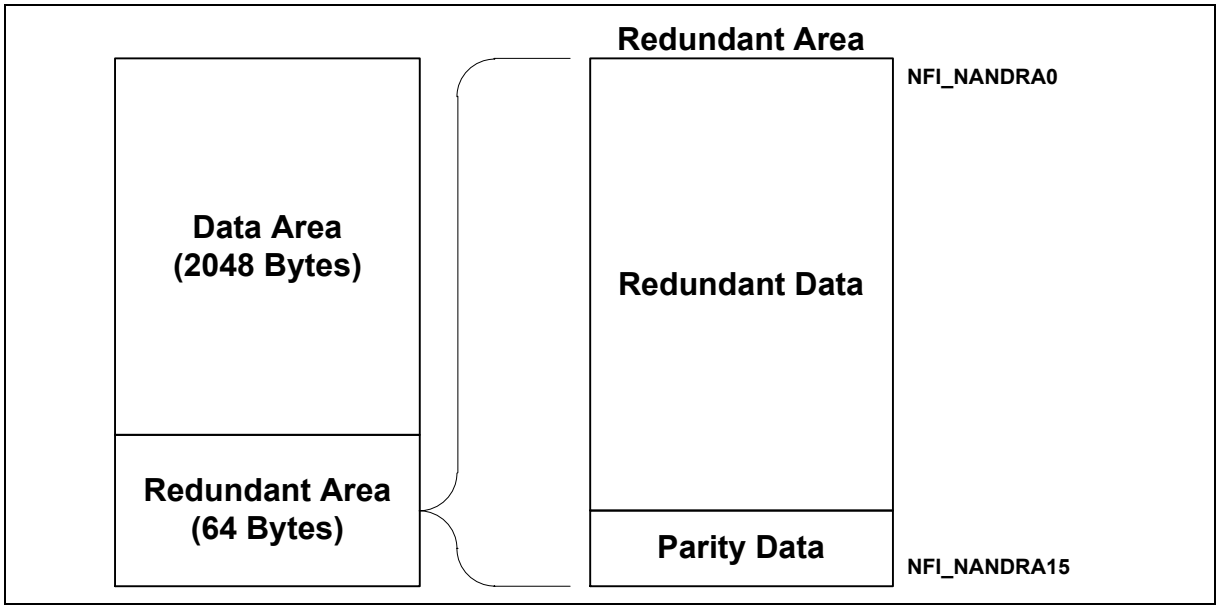


Figure 6.36-2 Data Arrangement for 2 KB Page Size NAND Flash

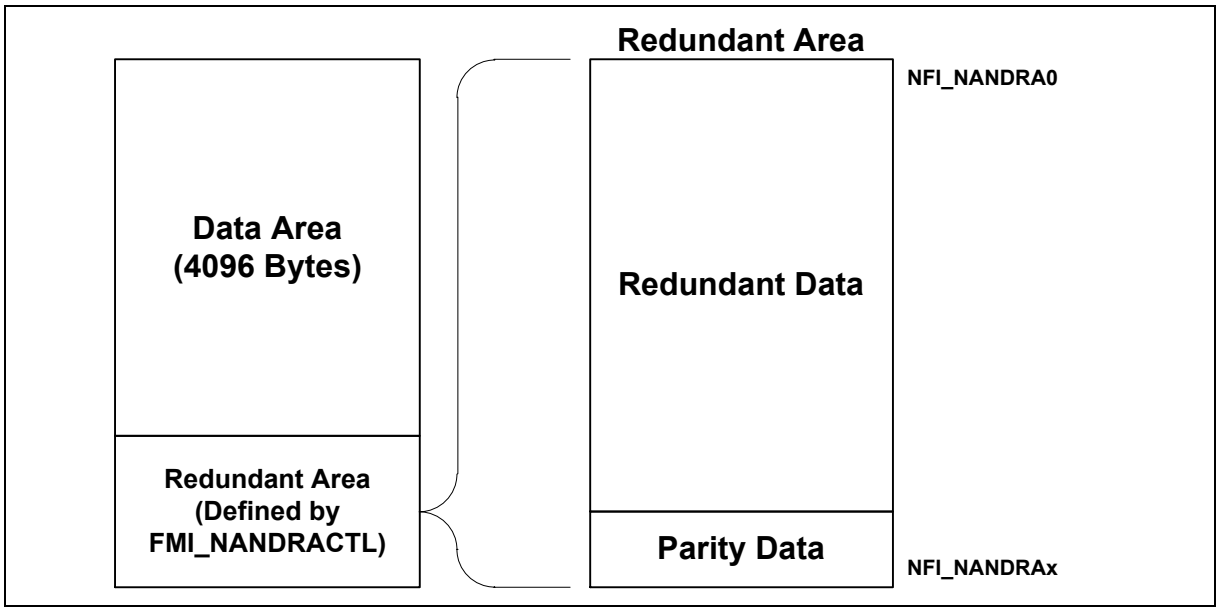


Figure 6.36-3 Data Arrangement for 4 KB Page Size NAND Flash

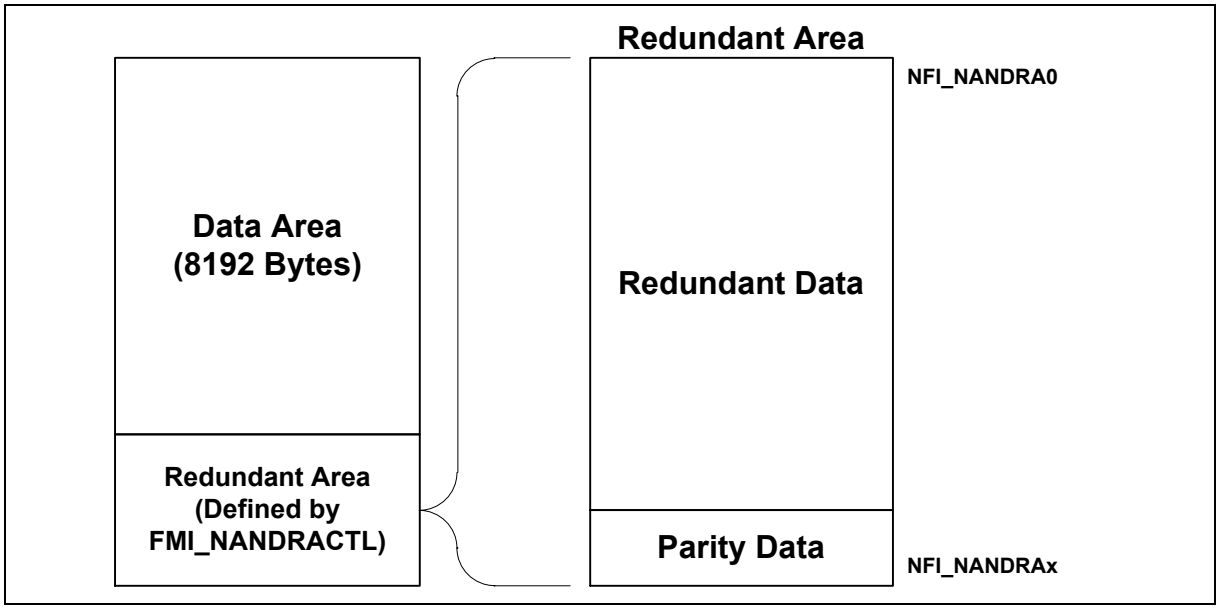


Figure 6.36-4 Data Arrangement for 8 KB Page Size NAND Flash

6.36.5.4 EDO Mode

The NAND Flash controller can support Extend Data Output (EDO) mode that allows the processor to access NAND Flash memory faster, but must match the NAND Flash EDO design used. Before entering EDO mode, set CALESH (NFI_NANDTMCTL[22:16]), HIWID (NFI_NANDTMCTL[15:8]) and LOWID (NFI_NANDTMCTL[7:0]) properly.

The NAND Flash CLS/ALE setup and hold time can be determinate by CALESH (NFI_NANDTMCTL[22:16]). There are different CLS/ALE setup and hold time according to different NAND. Figure 6.36-5 shows NAND Flash EDO mode timing; the low width and high width of nRE can be determinate by LOWID (NFI_NANDTMCTL[7:0]) and HIWID (NFI_NANDTMCTL[15:8]) separately. Set low width of nRE (tRP) to be smaller than tREA, then enter EDO mode. Software can use EDOD (NFI_NANDTMCTL[27:24]) to delay sampling point when entering EDO mode. The delay time can be adjusted every 0.5*HCLK and the maximum delay time is 8*HCLK. The data can be latched on rising edge of sample point.

After setting CALESH (NFI_NANDTMCTL[22:16]), HIWID (NFI_NANDTMCTL[15:8]) and LOWID (NFI_NANDTMCTL[7:0]), then set EDOEN (NFI_NANDTMCTL[31]) high to enable EDO function and check if EDOF (NFI_NANDINTSTS[4]) is asserted.

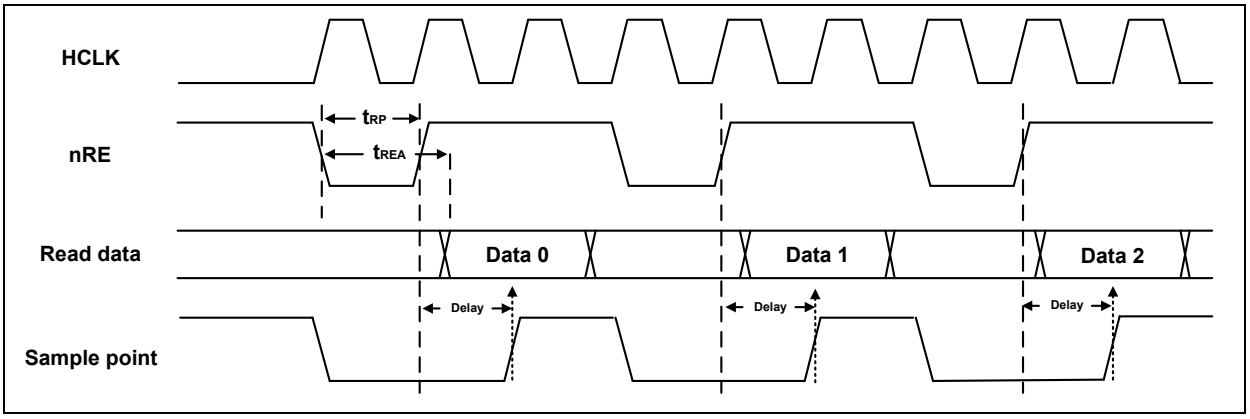


Figure 6.36-5 NAND Flash Read Path Timing Diagram (EDO Mode)

6.36.5.5 Programming Flow

Here is a simple example programming flow without DMA Scatter-Gather enable.

1. Set DMACEN (NFI_DMACTL[0]) to enable DMA Controller.
2. Fill in the corresponding starting address in NFI_DMASA for NFI.
3. Set NANDEN (NFI_GCTL[3]) to enable IP to start DMA transfer and set DMAIE (NFI_NANDINTEN [0]) to enable DMA read or write data complete interrupt.
4. Wait IP to be finished, DMAIF (NFI_NANDINTSTS[0]) will be asserted.

Here is a simple example programming flow with DMA Scatter-Gather enable.

1. Set DMACEN (NFI_DMACTL[0]) to enable DMA and SGEN (NFI_DMACTL[3]) to enable Scatter-Gather function.
2. Fill in the corresponding starting address of Physical Address Descriptor (PAD) table in NFI_DMASA for NFI.
3. When bit-0 of NFI_DMASA is 1, the PAD will fetch in out of order, otherwise, it is fetched in order from PAD. The first time of writing bit-0 with 1 or not is not available for this function. The bits will be available in PAD table.
4. Set NANDEN (NFI_GCTL[3]) to enable IP to start DMA transfer and set DMAIE (NFI_NANDINTEN [0]) to enable DMA read or write data complete interrupt.
5. Wait IP to be finished, DMAIF (NFI_NANDINTSTS[0]) will be asserted.

6.36.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
NFI Base Address: **NFI_BA = 0x401A_0000**				
NFI_BUFFERn **n = 0, 1..31**	NFI_BA+0x000+0x4*n	R/W	NFI Embedded Buffer Word n	0x0000_0000
NFI_DMACTL	NFI_BA+0x400	R/W	NFI DMA Control and Status Register	0x0000_0000
NFI_DMASA	NFI_BA+0x408	R/W	NFI DMA Transfer Starting Address Register	0x0000_0000
NFI_DMABCNT	NFI_BA+0x40C	R	NFI DMA Transfer Byte Count Register	0x0000_0000
NFI_DMAINTEN	NFI_BA+0x410	R/W	NFI DMA Interrupt Enable Control Register	0x0000_0001
NFI_DMAINTSTS	NFI_BA+0x414	R/W	NFI DMA Interrupt Status Register	0x0000_0000
NFI_GCTL	NFI_BA+0x800	R/W	NFI Global Control and Status Register	0x0000_0000
NFI_GINTEN	NFI_BA+0x804	R/W	NFI Global Interrupt Control Register	0x0000_0001
NFI_GINTSTS	NFI_BA+0x808	R/W	NFI Global Interrupt Status Register	0x0000_0000
NFI_NANDCTL	NFI_BA+0x8A0	R/W	NAND Flash Control Register	0x0291_0090
NFI_NANDTMCTL	NFI_BA+0x8A4	R/W	NAND Flash Timing Control Register	0x0501_0309
NFI_NANDINTEN	NFI_BA+0x8A8	R/W	NAND Flash Interrupt Enable Register	0x0000_0000
NFI_NANDINTSTS	NFI_BA+0x8AC	R/W	NAND Flash Interrupt Status Register	0x00XX_0000
NFI_NANDCMD	NFI_BA+0x8B0	W	NAND Flash Command Port Register	0xXXXX_XXXX
NFI_NANDADDR	NFI_BA+0x8B4	W	NAND Flash Address Port Register	0xXXXX_XXXX
NFI_NANDDATA	NFI_BA+0x8B8	R/W	NAND Flash Data Port Register	0xXXXX_XXXX
NFI_NANDRACTL	NFI_BA+0x8BC	R/W	NAND Flash Redundant Area Control Register	0x0000_0000
NFI_NANDECTL	NFI_BA+0x8C0	R/W	NAND Flash Extend Control Register	0x0000_0000
NFI_NANDECCES0	NFI_BA+0x8D0	R	NAND Flash ECC Error Status 0 Register	0x0000_0000
NFI_NANDECCES1	NFI_BA+0x8D4	R	NAND Flash ECC Error Status 1 Register	0x0000_0000
NFI_NANDECCES2	NFI_BA+0x8D8	R	NAND Flash ECC Error Status 2 Register	0x0000_0000
NFI_NANDECCES3	NFI_BA+0x8DC	R	NAND Flash ECC Error Status 3 Register	0x0000_0000
NFI_NANDECCEA0	NFI_BA+0x900	R	NAND Flash ECC Error Byte Address 0 Register	0x0000_0000
NFI_NANDECCEA1	NFI_BA+0x904	R	NAND Flash ECC Error Byte Address 1 Register	0x0000_0000
NFI_NANDECCEA2	NFI_BA+0x908	R	NAND Flash ECC Error Byte Address 2 Register	0x0000_0000
NFI_NANDECCEA3	NFI_BA+0x90C	R	NAND Flash ECC Error Byte Address 3 Register	0x0000_0000
NFI_NANDECCEA4	NFI_BA+0x910	R	NAND Flash ECC Error Byte Address 4 Register	0x0000_0000
NFI_NANDECCEA5	NFI_BA+0x914	R	NAND Flash ECC Error Byte Address 5 Register	0x0000_0000
NFI_NANDECCEA6	NFI_BA+0x918	R	NAND Flash ECC Error Byte Address 6 Register	0x0000_0000
NFI_NANDECCEA7	NFI_BA+0x91C	R	NAND Flash ECC Error Byte Address 7 Register	0x0000_0000

NFI_NANDECCEA8	NFI_BA+0x920	R	NAND Flash ECC Error Byte Address 8 Register	0x0000_0000
NFI_NANDECCEA9	NFI_BA+0x924	R	NAND Flash ECC Error Byte Address 9 Register	0x0000_0000
NFI_NANDECCEA10	NFI_BA+0x928	R	NAND Flash ECC Error Byte Address 10 Register	0x0000_0000
NFI_NANDECCEA11	NFI_BA+0x92C	R	NAND Flash ECC Error Byte Address 11 Register	0x0000_0000
NFI_NANDECCED0	NFI_BA+0x960	R	NAND Flash ECC Error Data Register 0	0x8080_8080
NFI_NANDECCED1	NFI_BA+0x964	R	NAND Flash ECC Error Data Register 1	0x8080_8080
NFI_NANDECCED2	NFI_BA+0x968	R	NAND Flash ECC Error Data Register 2	0x8080_8080
NFI_NANDECCED3	NFI_BA+0x96C	R	NAND Flash ECC Error Data Register 3	0x8080_8080
NFI_NANDECCED4	NFI_BA+0x970	R	NAND Flash ECC Error Data Register 4	0x8080_8080
NFI_NANDECCED5	NFI_BA+0x974	R	NAND Flash ECC Error Data Register 5	0x8080_8080
NFI_NANDRAn **n = 0, 1..117**	NFI_BA+0xA00+0x4*n	R/W	NAND Flash Redundant Area Word n	Undefined

6.36.7 Register Description

NFI Embedded Buffer Word n (NFI_BUFFERn)

Register	Offset	R/W	Description	Reset Value
NFI_BUFFERn **n = 0, 1..31**	NFI_BA+0x000+0x4*n	R/W	NFI Embedded Buffer Word n	0x0000_0000

31	30	29	28	27	26	25	24
Data							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Data							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Data							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Data							

Bits	Description	
[31:0]	**Data**	**NFI Embedded Buffer Word n** This field indicates a 32-bit data of NAND Flash controller embedded buffer.

NFI DMA Control and Status Register (NFI_DMACTL)

Register	Offset	R/W	Description	Reset Value
NFI_DMACTL	NFI_BA+0x400	R/W	NFI DMA Control and Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						DMABUSY	Reserved
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				SGEN	Reserved	DMARST	DMACEN

Bits	Description	
[31:10]	**Reserved**	Reserved.
[9]	**DMABUSY**	**NFI DMA Transfer in Progress** This bit indicates if NFI is granted and doing DMA transfer or not. 0 = NFI DMA transfer is not in progress. 1 = NFI DMA transfer is in progress.
[8:4]	**Reserved**	Reserved.
[3]	**SGEN**	**Scatter-gather Function for NFI Enable Bit** 0 = Scatter-gather function Disabled (DMA will treat the starting address in DMASA as starting pointer of a single block memory). 1 = Scatter-gather function Enabled (DMA will treat the starting address in DMASA as a starting address of Physical Address Descriptor (PAD) table. The format of these Pads' will be described later).
[2]	**Reserved**	Reserved.
[1]	**DMARST**	**Software Engine Reset** 0 = No effect. 1 = Reset internal state machine and pointers. The contents of control register will not be cleared. This bit will be auto cleared after a few clock cycles. **Note:** The software reset DMA related registers.
[0]	**DMACEN**	**DMA Controller Engine Enable Bit** 0 = DMA Controller Disabled. 1 = DMA Controller Enabled. **Note 1:** If this bit is cleared, DMA will ignore all requests from NAND Flash and force bus master into IDLE state. **Note 2:** If target abort occurred, DMACEN will be cleared.

NFI DMA Transfer Starting Address Register (NFI_DMASA)

Register	Offset	R/W	Description	Reset Value
NFI_DMASA	NFI_BA+0x408	R/W	NFI DMA Transfer Starting Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DMASA							
23	22	21	20	19	18	17	16
DMASA							
15	14	13	12	11	10	9	8
DMASA							
7	6	5	4	3	2	1	0
DMASA							ORDER

Bits	Description	
[31:1]	**DMASA**	**DMA Transfer Starting Address** This field pads 0 as least significant bit indicates a 32-bit starting address of system memory (SRAM) for DMA to retrieve or fill in data. If DMA is not in normal mode, this field will be interpreted as a starting address of Physical Address Descriptor (PAD) table. **Note:** Starting address of the SRAM must be word aligned, for example, 0x0000_0000, 0x0000_0004.
[0]	**ORDER**	**Determined to the PAD Table Fetching Is in Order or Out of Order** 0 = PAD table is fetched in order. 1 = PAD table is fetched out of order. **Note:** The bit 0 is valid in scatter-gather mode when SGEN (NFI_DMACTL[3]) = 1.

NFI DMA Transfer Byte Count Register (NFI_DMABCNT)

Register	Offset	R/W	Description	Reset Value
NFI_DMABCNT	NFI_BA+0x40C	R	NFI DMA Transfer Byte Count Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				BCNT			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
BCNT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BCNT							

Bits	Description	
[31:20]	**Reserved**	Reserved.
[19:0]	**BCNT**	**DMA Transfer Byte Count (Read Only)** This field indicates the remained byte count of DMA transfer. The value of this field is valid only when NFI is busy; otherwise, it is zero.

NFI DMA Interrupt Enable Control Register (NFI_DMAINTEN)

Register	Offset	R/W	Description	Reset Value
NFI_DMAINTEN	NFI_BA+0x410	R/W	NFI DMA Interrupt Enable Control Register	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						WEOTIEN	ABORTIEN

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**WEOTIEN**	**Wrong EOT Encountered Interrupt Enable Bit** 0 = Interrupt generation Disabled when wrong EOT (end of transfer) is encountered. 1 = Interrupt generation Enabled when wrong EOT (end of transfer) is encountered.
[0]	**ABORTIEN**	**DMA Read/Write Target Abort Interrupt Enable Bit** 0 = Target abort interrupt generation Disabled during DMA transfer. 1 = Target abort interrupt generation Enabled during DMA transfer.

NFI DMA Interrupt Status Register (NFI_DMAINTSTS)

Register	Offset	R/W	Description	Reset Value
NFI_DMAINTSTS	NFI_BA+0x414	R/W	NFI DMA Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						WEOTIF	ABORTIF

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**WEOTIF**	**Wrong EOT Encountered Interrupt Flag (Read Only)** When DMA Scatter-Gather function is enabled, and EOT of the descriptor is encountered before DMA transfer finished (that means the total sector count of all PAD is less than the sector count of NFI), this bit will be set. 0 = No EOT encountered before DMA transfer finished. 1 = EOT encountered before DMA transfer finished. **Note:** This bit is read only, but can be cleared by writing '1' to it.
[0]	**ABORTIF**	**DMA Read/Write Target Abort Interrupt Flag (Read Only)** 0 = No bus ERROR response received. 1 = Bus ERROR response received. **Note 1:** This bit is read only, but can be cleared by writing '1' to it. **Note 2:** When DMA's bus master received ERROR response, it means that target abort happened. DMA will stop transfer and respond this event by set ABORTIF high. Then, NFI go to IDLE state. When target abort occurred or WEOTIF is set, it is necessary to reset NFI's DMA and then transfer those data again.

NFI Global Control and Status Register (NFI_GCTL)

Register	Offset	R/W	Description	Reset Value
NFI_GCTL	NFI_BA+0x800	R/W	NFI Global Control and Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				NANDEN	Reserved		GCTLRST

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3]	**NANDEN**	**NAND Flash Functionality Enable Bit** 0 = NAND Flash functionality of NFI Disabled. 1 = NAND Flash functionality of NFI Enabled.
[2:1]	**Reserved**	Reserved.
[0]	**GCTLRST**	**Software Engine Reset** 0 = No effect. 1 = Reset all NFI engines. **Note:** The contents of control register will not be cleared. This bit will auto cleared after a few clock cycles.

NFI Global Interrupt Control Register (NFI_GINTEN)

Register	Offset	R/W	Description	Reset Value
NFI_GINTEN	NFI_BA+0x804	R/W	NFI Global Interrupt Control Register	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							DTAIEN

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**DTAIEN**	**DMA READ/WRITE Target Abort Interrupt Enable Bit** 0 = DMA READ/WRITE target abort interrupt generation Disabled. 1 = DMA READ/WRITE target abort interrupt generation Enabled.

NFI Global Interrupt Status Register (NFI_GINTSTS)

Register	Offset	R/W	Description	Reset Value
NFI_GINTSTS	NFI_BA+0x808	R/W	NFI Global Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							DTAIF

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**DTAIF**	**DMA READ/WRITE Target Abort Interrupt Flag (Read Only)** This bit indicates DMA received an ERROR response from internal AHB bus during DMA read/write operation. When Target Abort occurred, please reset all engines. 0 = No bus ERROR response received. 1 = Bus ERROR response received. **Note:** This bit is read only, but can be cleared by writing '1' to it.

NAND Flash Control Register (NFI_NANDCTL)

Register	Offset	R/W	Description	Reset Value
NFI_NANDCTL	NFI_BA+0x8A0	R/W	NAND Flash Control Register	0x0291_0090

31	30	29	28	27	26	25	24
Reserved					CS1	CS0	Reserved
23	22	21	20	19	18	17	16
ECCEN	BCHTSEL					PSIZE	
15	14	13	12	11	10	9	8
Reserved						SRAMINT	PROT3BEN
7	6	5	4	3	2	1	0
ECCCHK	Reserved		REDUNAUTO WEN	REDUNREN	DWREN	DRDEN	SWRST

Bits	Description	
[31:27]	**Reserved**	Reserved.
[26]	**CS1**	**NAND Flash Chip Select 1 Enable Bit** 0 = Chip select 1 Enabled. 1 = Chip select 1 Disabled.
[25]	**CS0**	**NAND Flash Chip Select 0 Enable Bit** 0 = Chip select 0 Enabled. 1 = Chip select 0 Disabled.
[24]	**Reserved**	Reserved.
[23]	**ECCEN**	**ECC Algorithm Enable Bit** This field is used to select the ECC algorithm for data protecting. The BCH algorithm can correct 8 or 12 or 24 bits. 0 = BCH code encode/decode Disabled. 1 = BCH code encode/decode Enabled. **Note 1:** If disabling ECCEN and when reading data from NAND, the NAND controller will ignore its ECC check result. When writing data to NAND, the NAND controller will write out 0xFF to every parity field. **Note 2:** The ECC algorithm only protects data area and hardware ECC parity code by default. By setting PROT3BEN (NFI_NANDCTL[8]) high, the first 3 bytes of redundant data are also protected by ECC algorithm.
[22:18]	**BCHTSEL**	**BCH Correct Bit Selection** This field is used to select BCH correct bits for data protecting. For BCH algorithm, T can be 8 or 12 or 24 for choosing (correct 8 or 12 or 24 bits). 00001 = Using BCH T24 to encode/decode (T24).(.1024 Bytes per block) 00100 = Using BCH T8 to encode/decode (T8). 01000 = Using BCH T12 to encode/decode (T12).

[17:16]	**PSIZE**	**Page Size of NAND** This bit indicates the page size of NAND. There are four page sizes for choose, 2048 bytes/page, 4096 bytes/page and 8192 bytes/page. Before setting PSIZE register, user must set BCHTSEL register at first. 01 = Page size is 2048 bytes/page. 10 = Page size is 4096 bytes/page. 11 = Page size is 8192 bytes/page.
[15:10]	**Reserved**	Reserved.
[9]	**SRAMINT**	**SRAM Initial** 0 = No effect. 1 = Reset the internal NFI_NANDRA0~NFI_NANDRA1 to 0xFFFF_FFFF. **Note:** The contents of control register will not be cleared. This bit will be auto cleared after a few clock cycles.
[8]	**PROT3BEN**	**Protect_3Byte Software Data Enable Bit** The ECC algorithm only protects data area and hardware ECC parity code. User can choose to protect software redundant data first 3 bytes by setting this bit high. 0 = Software redundant data is not protected by ECC algorithm. 1 = Software redundant data first 3 bytes protected by ECC algorithm.
[7]	**ECCCHK**	**None Used Field ECC Check After Read Page Data** 0 = ECC check Disabled. The NAND controller will always check ECC result for each field, no matter it is used or not. 1 = ECC check Enabled. The NAND controller will check 1's count for byte 2, 3 of redundant data of the ECC in each field. If the count value is greater than 8, the NAND controller will treat this field as none used field; otherwise, it is used. If that field is none used field, the NAND controller will ignore its ECC check result.
[6:5]	**Reserved**	Reserved.
[4]	**REDUNAUTOWEN**	**Redundant Area Auto Write Enable Bit** This field is used to auto write redundant data out to NAND Flash. The redundant data area is dependent on NFI_NANDRACTL register. 0 = Auto write redundant data out to NAND Flash Disabled. 1 = Auto write redundant data out to NAND Flash Enabled.
[3]	**REDUNREN**	**Redundant Area Read Enable Bit** This bit enables NAND controller to transfer redundant data from NAND Flash into NFI_NANDRA, the data size is dependent on NFI_NANDRACTL register. 0 = No effect. 1 = Read redundant data transfer Enabled. **Note:** When transfer completed, this bit will be cleared automatically.
[2]	**DWREN**	**DMA Write Data Enable Bit** This bit enables NAND controller to transfer data (1 page) from DMAC's embedded frame buffer into NAND Flash or NAND type Flash. 0 = No effect. 1 = DMA write data transfer Enabled. **Note:** When DMA transfer completed, this bit will be cleared automatically.
[1]	**DRDEN**	**DMA Read Data Enable Bit** This bit enables NAND controller to transfer data (1 page) from NAND Flash or NAND type Flash into DMAC's embedded frame buffer. 0 = No effect. 1 = DMA read data transfer Enabled. **Note:** When DMA transfer completed, this bit will be cleared automatically.

[0]	**SWRST**	**Software Engine Reset** 0 = No effect. 1 = Reset the internal state machine and counters (include DWREN (NFI_NANDCTL[2]) and DRDEN (NFI_NANDCTL[1])). **Note:** The contents of control register will not be cleared. This bit will be auto cleared after a few clock cycles.

NAND Flash Timing Control Register (NFI_NANDTMCTL)

Register	Offset	R/W	Description	Reset Value
NFI_NANDTMCTL	NFI_BA+0x8A4	R/W	NAND Flash Timing Control Register	0x0501_0309

31	30	29	28	27	26	25	24
EDOEN	Reserved			EDOD			
23	22	21	20	19	18	17	16
Reserved	CALESH						
15	14	13	12	11	10	9	8
HIWID							
7	6	5	4	3	2	1	0
LOWID							

Bits	Description	
[31]	**EDOEN**	**EDO Mode Enable Bit** This bit specifies NAND Flash to EDO mode. Before trigger this bit, software should fill EDOD(NFI_NANDTMCTL[27:24]), CALESH(NFI_NANDTMCTL[22:16]), HIWID(NFI_NANDTMCTL[15:8]) and LOWID(NFI_NANDTMCTL[7:0]). 0 = EDO Mode related function Disabled. 1 = EDO Mode related function Enabled. **Note:** Only NAND Flash entering EDO mode should set this bit.
[30:28]	**Reserved**	Reserved.
[27:24]	**EDOD**	**EDO Mode Delay Time** Control this field to delay sampling point when NAND Flash enters EDO mode only. The EDO mode delay time can be calculated using following equation: Delay time = (EDOD+1)*(TAHB/2).
[23]	**Reserved**	Reserved.
[22:16]	**CALESH**	**CLE/ALE Setup/Hold Time** This field controls the CLE/ALE setup/hold time to –WE. The setup/hold time can be calculated using following equation: tCLS = (CALESH+1)*TAHB. tCLH = ((CALESH*2)+2)*TAHB. tALS = (CALESH+1)*TAHB. tALH = ((CALESH*2)+2)*TAHB.
[15:8]	**HIWID**	**Read/Write Enable Signal High Pulse Width** This field controls the high pulse width of signals –RE and –WE while HARDWARE mode page access is enabled. The pulse width is a multiple of period of AHB bus clock. (The actual width time will be [clock period*(HIWID+1)])
[7:0]	**LOWID**	**Read/Write Enable Signal Low Pulse Width** This field controls the low pulse width of signals –RE and –WE while HARDWARE mode page access is enabled. The pulse width is a multiple of period of AHB bus clock. (The actual width time will be [clock period*(LOWID+1)])

NOTE1: The reset value calculated based on 180 MHz AHB Clock.

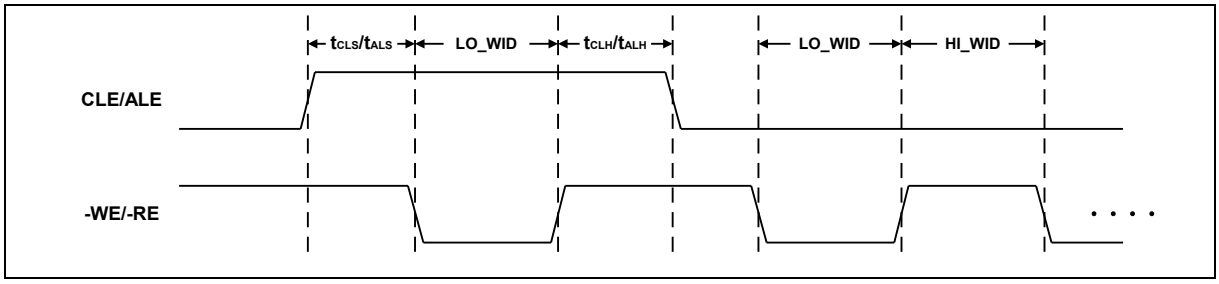


Figure 6.36-6 Timing Controlled by NFI_NANDTMCTL Register

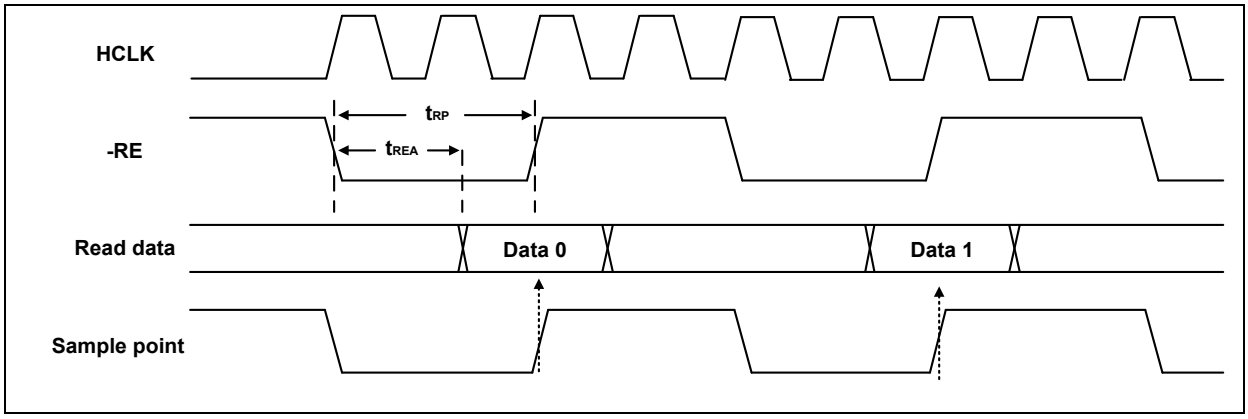


Figure 6.36-7 NAND Flash Read Path Timing Diagram (Non-EDO mode)

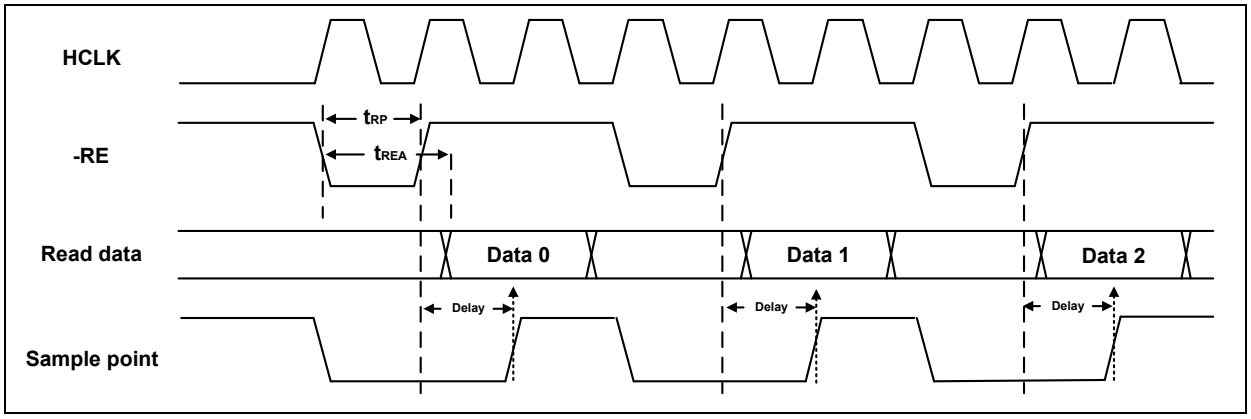


Figure 6.36-8 NAND Flash Read Path Timing Diagram (EDO Mode)

NAND Flash Interrupt Enable Register (NFI_NANDINTEN)

Register	Offset	R/W	Description	Reset Value
NFI_NANDINTEN	NFI_BA+0x8A8	R/W	NAND Flash Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				RB1IE	RB0IE	Reserved	
7	6	5	4	3	2	1	0
Reserved					ECCFLDIE	Reserved	DMAIE

Bits	Description	
[31:12]	**Reserved**	Reserved.
[11]	**RB1IE**	**Ready/-Busy 1 Rising Edge Detect Interrupt Enable Bit** 0 = R/-B rising edge detect interrupt generation Disabled. 1 = R/-B rising edge detect interrupt generation Enabled.
[10]	**RB0IE**	**Ready/-Busy 0 Rising Edge Detect Interrupt Enable Bit** 0 = R/-B rising edge detect interrupt generation Disabled. 1 = R/-B rising edge detect interrupt generation Enabled.
[9:3]	**Reserved**	Reserved.
[2]	**ECCFLDIE**	**ECC Field Check Error Interrupt Enable Bit** This bit can check the ECC error on each field (512bytes) of data transfer. Enable this bit to detect error and do error correction. 0 = ECC field check error Disabled. 1 = ECC field check error Enabled.
[1]	**Reserved**	Reserved.
[0]	**DMAIE**	**DMA Read/Write Data Complete Interrupt Enable Bit** 0 = DMA read/write data complete interrupt generation Disabled. 1 = DMA read/write data complete interrupt generation Enabled.

NAND Flash Interrupt Status Register (NFI_NANDINTSTS)

Register	Offset	R/W	Description	Reset Value
NFI_NANDINTSTS	NFI_BA+0x8AC	R/W	NAND Flash Interrupt Status Register	0x00XX_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				RB1Status	RB0Status	Reserved	
15	14	13	12	11	10	9	8
Reserved				RB1IF	RB0IF	Reserved	
7	6	5	4	3	2	1	0
Reserved			EDOF	Reserved	ECCFLDIF	Reserved	DMAIF

Bits	Description	
[31:20]	**Reserved**	Reserved.
[19]	**RB1Status**	**Ready/-Busy 1 Pin Status (Read Only)** This bit reflects the Ready/-Busy pin status of NAND Flash.
[18]	**RB0Status**	**Ready/-Busy 0 Pin Status (Read Only)** This bit reflects the Ready/-Busy pin status of NAND Flash.
[17:12]	**Reserved**	Reserved.
[11]	**RB1IF**	**Ready/-Busy 1 Rising Edge Detect Interrupt Flag (Read Only)** 0 = R/-B rising edge is not detected. 1 = R/-B rising edge is detected. **Note:** This bit is read only, but can be cleared by writing '1' to it.
[10]	**RB0IF**	**Ready/-Busy 0 Rising Edge Detect Interrupt Flag (Read Only)** 0 = R/-B rising edge is not detected. 1 = R/-B rising edge is detected. **Note:** This bit is read only, but can be cleared by writing '1' to it.
[9:5]	**Reserved**	Reserved.
[4]	**EDOF**	**EDO Mode Entrance Flag (Read Only)** 0 = NAND Flash did not enter EDO mode. 1 = NAND Flash entered EDO mode. **Note:** This bit is read only, but can be cleared by writing '1' to it.
[3]	**Reserved**	Reserved.
[2]	**ECCFLDIF**	**ECC Field Check Error Interrupt Flag (Read Only)** This bit can check the ECC error on each field (512bytes) of data transfer. Read this bit to check if the error occurred. 0 = No occurrence of ECC error. 1 = Occurrence of ECC error. **Note:** This bit is read only, but can be cleared by writing '1' to it.

[1]	**Reserved**	Reserved.
[0]	**DMAIF**	**DMA Read/Write Data Complete Interrupt Flag (Read Only)** 0 = DMA read/write transfer is not finished yet. 1 = DMA read/write transfer is done. **Note:** This bit is read only, but can be cleared by writing '1' to it.

NAND Flash Command Port Register (NFI_NANDCMD)

Register	Offset	R/W	Description	Reset Value
NFI_NANDCMD	NFI_BA+0x8B0	W	NAND Flash Command Port Register	0xXXXX_XXXX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
COMMAND							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**COMMAND**	**NAND Flash Command Port** When CPU writes to this port, NFI will send a command to NAND Flash.

NAND Flash Address Port Register (NFI_NANDADDR)

Register	Offset	R/W	Description	Reset Value
NFI_NANDADDR	NFI_BA+0x8B4	W	NAND Flash Address Port Register	0xXXXX_XXXX

31	30	29	28	27	26	25	24
EOA	Reserved						
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
ADDRESS							

Bits	Description	
[31]	**EOA**	**End of Address** Write this bit to indicate if this address is the last one or not. By writing address port with this bit low, the NAND Flash controller will set ALE pin to active (HIGH). After the last address is written (with this bit set high), the NAND Flash controller will set ALE pin to inactive (LOW). 0 = Not the last address cycle. 1 = The last one address cycle.
[30:8]	**Reserved**	Reserved.
[7:0]	**ADDRESS**	**NAND Flash Address Port** By writing this port, NAND Flash control will send an address to NAND Flash.

NAND Flash Data Port Register (NFI_NANDDATA)

Register	Offset	R/W	Description	Reset Value
NFI_NANDDATA	NFI_BA+0x8B8	R/W	NAND Flash Data Port Register	0xXXXX_XXXX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATA							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**DATA**	**NAND Flash Data Port** CPU can access NAND's memory array through this data port. When CPU WRITE, the lower 8-bit data from CPU will appear on the data bus of NAND controller. When CPU READ, NAND controller will get 8-bit data from data bus.

NAND Flash Redundant Area Control Register (NFI_NANDRACTL)

Register	Offset	R/W	Description	Reset Value
NFI_NANDRACTL	NFI_BA+0x8BC	R/W	NAND Flash Redundant Area Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MECC							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
MECC							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							RA128EN
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RA128EN							

Bits	Description	
[31:16]	**MECC**	**Mask ECC During Write Page Data** These 16 bits registers indicate NAND controller to write out ECC parity or just 0xFF for each field (every 512 bytes) the real parity data will be write out to NFI_NANDRAx. 0x00 = Do not mask the ECC parity for each field. 0x01 = Mask ECC parity and write out FF to NAND ECC parity for 512 Bytes page size or 2K/4K/8K page size first 512 field. 0x02 = Mask ECC parity and write out FF to NAND ECC parity for 512 Bytes page size or 2K/4K/8K page size second 512 field. Others = Mask ECC parity and write out FF to NAND ECC parity for 512 Bytes page size or 2K/4K/8K page size each 512 field.
[15:9]	**Reserved**	Reserved.
[8:0]	**RA128EN**	**Redundant Area 128 Byte Enable Bit** These bits indicate NAND Flash extended redundant area. If PSIZE (NFI_NANDCTL[17:16]) = 2'b01, this field will be set 0x40 (64 bytes) automatically. If PSIZE (NFI_NANDCTL[17:16]) = 2'b10, this field will be set 0x80 (128 bytes) automatically. If PSIZE (NFI_NANDCTL[17:16]) = 2'b11, this field will be set 0x100 (256 bytes) automatically. **Note:** The REA128EN must be 4 byte aligned, so bit1 and bit0 can't be filled 1 to it. The maximum redundant area of the controller is 472 bytes.

NAND Flash Extend Control Register (NFI_NANDECTL)

Register	Offset	R/W	Description	Reset Value
NFI_NANDECTL	NFI_BA+0x8C0	R/W	NAND Flash Extend Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							WP

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**WP**	**NAND Flash Write Protect Control** Set this bit low (low active) to make NAND_nWP functional pin low to prevent the write to NAND Flash device. 0 = NAND Flash is write-protected and is not writeable. 1 = NAND Flash is not write-protected and is writeable.

NAND Flash ECC Error Status 0 Register (NFI_NANDECCES0)

Register	Offset	R/W	Description	Reset Value
NFI_NANDECCES0	NFI_BA+0x8D0	R	NAND Flash ECC Error Status 0 Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	F4ECNT					F4STAT	
23	22	21	20	19	18	17	16
Reserved	F3ECNT					F3STAT	
15	14	13	12	11	10	9	8
Reserved	F2ECNT					F2STAT	
7	6	5	4	3	2	1	0
Reserved	F1ECNT					F1STAT	

Bits	Description	
[31]	**Reserved**	Reserved.
[30:26]	**F4ECNT**	**Error Count of ECC Field 4** This field contains the error counts after ECC correct calculation of Field 4. For this ECC core (BCH algorithm), only when F4STAT equals to 0x01, the value in this field is meaningful. F4ECNT means how many errors depending on which ECC is used.
[25:24]	**F4STAT**	**ECC Status of Field 4** This field contains the ECC correction status (BCH algorithm) of ECC-field 4. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error. 11 = Reserved.
[23]	**Reserved**	Reserved.
[22:18]	**F3ECNT**	**Error Count of ECC Field 3** This field contains the error counts after ECC correct calculation of Field 3. For this ECC core (BCH algorithm), only when F3STAT equals to 0x01, the value in this field is meaningful. F3ECNT means how many errors depending on which ECC is used.
[17:16]	**F3STAT**	**ECC Status of Field 3** This field contains the ECC correction status (BCH algorithm) of ECC-field 3. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error. 11 = Reserved.
[15]	**Reserved**	Reserved.
[14:10]	**F2ECNT**	**Error Count of ECC Field 2** This field contains the error counts after ECC correct calculation of Field 2. For this ECC core (BCH algorithm), only when F2STAT equals to 0x01, the value in this field is meaningful. F2ECNT means how many errors depending on which ECC is used.

[9:8]	**F2STAT**	**ECC Status of Field 2** This field contains the ECC correction status (BCH algorithm) of ECC-field 2. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error. 11 = Reserved.
[7]	**Reserved**	Reserved.
[6:2]	**F1ECNT**	**Error Count of ECC Field 1** This field contains the error counts after ECC correct calculation of Field 1. For this ECC core (BCH algorithm), only when F1STAT equals to 0x01, the value in this field is meaningful. F1ECNT means how many errors depending on which ECC is used.
[1:0]	**F1STAT**	**ECC Status of Field 1** This field contains the ECC correction status (BCH algorithm) of ECC-field 1. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error. 11 = Reserved.

NAND Flash ECC Error Status 1 Register (NFI_NANDECCES1)

Register	Offset	R/W	Description	Reset Value
NFI_NANDECCES1	NFI_BA+0x8D4	R	NAND Flash ECC Error Status 1 Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	F8ECNT					F8STAT	
23	22	21	20	19	18	17	16
Reserved	F7ECNT					F7STAT	
15	14	13	12	11	10	9	8
Reserved	F6ECNT					F6STAT	
7	6	5	4	3	2	1	0
Reserved	F5ECNT					F5STAT	

Bits	Description	
[31]	**Reserved**	Reserved.
[30:26]	**F8ECNT**	**Error Count of ECC Field 8** This field contains the error counts after ECC correct calculation of Field 8. For this ECC core (BCH algorithm), only when F8STAT equals to 0x01, the value in this field is meaningful. F8ECNT means how many errors depending on which ECC is used.
[25:24]	**F8STAT**	**ECC Status of Field 8** This field contains the ECC correction status (BCH algorithm) of ECC-field 8. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error. 11 = Reserved.
[23]	**Reserved**	Reserved.
[22:18]	**F7ECNT**	**Error Count of ECC Field 7** This field contains the error counts after ECC correct calculation of Field 7. For this ECC core (BCH algorithm), only when F7STAT equals to 0x01, the value in this field is meaningful. F7ECNT means how many errors depending on which ECC is used.
[17:16]	**F7STAT**	**ECC Status of Field 7** This field contains the ECC correction status (BCH algorithm) of ECC-field 7. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error. 11 = Reserved.
[15]	**Reserved**	Reserved.
[14:10]	**F6ECNT**	**Error Count of ECC Field 6** This field contains the error counts after ECC correct calculation of Field 6. For this ECC core (BCH algorithm), only when F6STAT equals to 0x01, the value in this field is meaningful. F6ECNT means how many errors depending on which ECC is used.

[9:8]	**F6STAT**	**ECC Status of Field 6** This field contains the ECC correction status (BCH algorithm) of ECC-field 6. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error. 11 = Reserved.
[7]	**Reserved**	Reserved.
[6:2]	**F5ECNT**	**Error Count of ECC Field 5** This field contains the error counts after ECC correct calculation of Field 5. For this ECC core (BCH algorithm), only when F5STAT equals to 0x01, the value in this field is meaningful. F5ECNT means how many errors depending on which ECC is used.
[1:0]	**F5STAT**	**ECC Status of Field 5** This field contains the ECC correction status (BCH algorithm) of ECC-field5. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error. 11 = Reserved.

NAND Flash ECC Error Status 2 Register (NFI_NANDECCES2)

Register	Offset	R/W	Description	Reset Value
NFI_NANDECCES2	NFI_BA+0x8D8	R	NAND Flash ECC Error Status 2 Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	F12ECNT					F12STAT	
23	22	21	20	19	18	17	16
Reserved	F11ECNT					F11STAT	
15	14	13	12	11	10	9	8
Reserved	F10ECNT					F10STAT	
7	6	5	4	3	2	1	0
Reserved	F9ECNT					F9STAT	

Bits	Description	
[31]	**Reserved**	Reserved.
[30:26]	**F12ECNT**	**Error Count of ECC Field 12** This field contains the error counts after ECC correct calculation of Field 12. For this ECC core (BCH algorithm), only when F12STAT equals to 0x01, the value in this field is meaningful. F12ECNT means how many errors depending on which ECC is used.
[25:24]	**F12STAT**	**ECC Status of Field 12** This field contains the ECC correction status (BCH algorithm) of ECC-field 12. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error. 11 = Reserved.
[23]	**Reserved**	Reserved.
[22:18]	**F11ECNT**	**Error Count of ECC Field 11** This field contains the error counts after ECC correct calculation of Field 11. For this ECC core (BCH algorithm), only when F11STAT equals to 0x01, the value in this field is meaningful. F11ECNT means how many errors depending on which ECC is used.
[17:16]	**F11STAT**	**ECC Status of Field 11** This field contains the ECC correction status (BCH algorithm) of ECC-field 11. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error. 11 = Reserved.
[15]	**Reserved**	Reserved.
[14:10]	**F10ECNT**	**Error Count of ECC Field 10** This field contains the error counts after ECC correct calculation of Field 10. For this ECC core (BCH algorithm), only when F10STAT equals to 0x01, the value in this field is meaningful. F10ECNT means how many errors depending on which ECC is used.

[9:8]	**F10STAT**	**ECC Status of Field 10** This field contains the ECC correction status (BCH algorithm) of ECC-field 10. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error. 11 = Reserved.
[7]	**Reserved**	Reserved.
[6:2]	**F9ECNT**	**Error Count of ECC Field 9** This field contains the error counts after ECC correct calculation of Field 9. For this ECC core (BCH algorithm), only when F9STAT equals to 0x01, the value in this field is meaningful. F9ECNT means how many errors depending on which ECC is used.
[1:0]	**F9STAT**	**ECC Status of Field 9** This field contains the ECC correction status (BCH algorithm) of ECC-field 9. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error. 11 = Reserved.

NAND Flash ECC Error Status 3 Register (NFI_NANDECCES3)

Register	Offset	R/W	Description	Reset Value
NFI_NANDECCES3	NFI_BA+0x8DC	R	NAND Flash ECC Error Status 3 Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	F16ECNT					F16STAT	
23	22	21	20	19	18	17	16
Reserved	F15ECNT					F15STAT	
15	14	13	12	11	10	9	8
Reserved	F14ECNT					F14STAT	
7	6	5	4	3	2	1	0
Reserved	F13ECNT					F13STAT	

Bits	Description	
[31]	**Reserved**	Reserved.
[30:26]	**F16ECNT**	**Error Count of ECC Field 16** This field contains the error counts after ECC correct calculation of Field 16. For this ECC core (BCH algorithm), only when F16STAT equals to 0x01, the value in this field is meaningful. F16ECNT means how many errors depending on which ECC is used.
[25:24]	**F16STAT**	**ECC Status of Field 16** This field contains the ECC correction status (BCH algorithm) of ECC-field 16. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error. 11 = Reserved.
[23]	**Reserved**	Reserved.
[22:18]	**F15ECNT**	**Error Count of ECC Field 15** This field contains the error counts after ECC correct calculation of Field 15. For this ECC core (BCH algorithm), only when F15STAT equals to 0x01, the value in this field is meaningful. F15ECNT means how many errors depending on which ECC is used.
[17:16]	**F15STAT**	**ECC Status of Field 15** This field contains the ECC correction status (BCH algorithm) of ECC-field 15. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error. 11 = Reserved.
[15]	**Reserved**	Reserved.
[14:10]	**F14ECNT**	**Error Count of ECC Field 14** This field contains the error counts after ECC correct calculation of Field 14. For this ECC core (BCH algorithm), only when F14STAT equals to 0x01, the value in this field is meaningful. F14ECNT means how many errors depending on which ECC is used.

[9:8]	**F14STAT**	**ECC Status of Field 14** This field contains the ECC correction status (BCH algorithm) of ECC-field 14. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error. 11 = Reserved.
[7]	**Reserved**	Reserved.
[6:2]	**F13ECNT**	**Error Count of ECC Field 13** This field contains the error counts after ECC correct calculation of Field 13. For this ECC core (BCH algorithm), only when F13STAT equals to 0x01, the value in this field is meaningful. F13ECNT means how many errors depending on which ECC is used.
[1:0]	**F13STAT**	**ECC Status of Field 13** This field contains the ECC correction status (BCH algorithm) of ECC-field 13. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error. 11 = Reserved.

NAND Flash ECC Error Byte Address 0 Register (NFI_NANDECCEA0)

Register	Offset	R/W	Description	Reset Value
NFI_NANDECCEA0	NFI_BA+0x900	R	NAND Flash ECC Error Byte Address 0 Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					ERRADDR1		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ERRADDR1							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					ERRADDR0		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ERRADDR0							

Bits	Description	
[31:27]	**Reserved**	Reserved.
[26:16]	**ERRADDR1**	**ECC Error Address First Field of Error 1** This field contains an 11-bit ECC error address 1 of first field. If it is a correctable error, please read the error data, ERRDATA1 (NFI_NANDECCED0[15:8]), to correct this error.
[15:11]	**Reserved**	Reserved.
[10:0]	**ERRADDR0**	**ECC Error Address First Field of Error 0** This field contains an 11-bit ECC error address 0 of first field. If it is a correctable error, please read the error data, ERRDATA0 (NFI_NANDECCED0[7:0]), to correct this error.

NAND Flash ECC Error Byte Address 1 Register (NFI_NANDECCEA1)

Register	Offset	R/W	Description	Reset Value
NFI_NANDECCEA1	NFI_BA+0x904	R	NAND Flash ECC Error Byte Address 1 Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					ERRADDR3		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ERRADDR3							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					ERRADDR2		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ERRADDR2							

Bits	Description	
[31:27]	**Reserved**	Reserved.
[26:16]	**ERRADDR3**	**ECC Error Address First Field of Error 3** This field contains an 11-bit ECC error address 3 of first field. If it is a correctable error, please read the error data, ERRDATA3 (NFI_NANDECCED0[31:24]), to correct this error.
[15:11]	**Reserved**	Reserved.
[10:0]	**ERRADDR2**	**ECC Error Address First Field of Error 2** This field contains an 11-bit ECC error address 2 of first field. If it is a correctable error, please read the error data, ERRDATA2 (NFI_NANDECCED0[23:16]), to correct this error.

NAND Flash ECC Error Byte Address 2 Register (NFI_NANDECCEA2)

Register	Offset	R/W	Description	Reset Value
NFI_NANDECCEA2	NFI_BA+0x908	R	NAND Flash ECC Error Byte Address 2 Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					ERRADDR5		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ERRADDR5							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					ERRADDR4		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ERRADDR4							

Bits	Description	
[31:27]	**Reserved**	Reserved.
[26:16]	**ERRADDR5**	**ECC Error Address First Field of Error 5** This field contains an 11-bit ECC error address 5 of first field. If it is a correctable error, please read the error data, ERRDATA5 (NFI_NANDECCED1[15:8]), to correct this error.
[15:11]	**Reserved**	Reserved.
[10:0]	**ERRADDR4**	**ECC Error Address First Field of Error 4** This field contains an 11-bit ECC error address 4 of first field. If it is a correctable error, please read the error data, ERRDATA4 (NFI_NANDECCED1[7:0]), to correct this error.

NAND Flash ECC Error Byte Address 3 Register (NFI_NANDECCEA3)

Register	Offset	R/W	Description	Reset Value
NFI_NANDECCEA3	NFI_BA+0x90C	R	NAND Flash ECC Error Byte Address 3 Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					ERRADD7		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ERRADDR7							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					ERRADDR6		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ERRADDR6							

Bits	Description	
[31:27]	**Reserved**	Reserved.
[26:16]	**ERRADDR7**	**ECC Error Address First Field of Error 7** This field contains an 11-bit ECC error address 7 of first field. If it is a correctable error, please read the error data, ERRDATA7 (NFI_NANDECCED1[31:24]), to correct this error.
[15:11]	**Reserved**	Reserved.
[10:0]	**ERRADDR6**	**ECC Error Address First Field of Error 6** This field contains an 11-bit ECC error address 6 of first field. If it is a correctable error, please read the error data, ERRDATA6 (NFI_NANDECCED1[23:16]), to correct this error.

NAND Flash ECC Error Byte Address 4 Register (NFI_NANDECCEA4)

Register	Offset	R/W	Description	Reset Value
NFI_NANDECCEA4	NFI_BA+0x910	R	NAND Flash ECC Error Byte Address 4 Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					ERRADDR9		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ERRADDR9							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					ERRADDR8		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ERRADDR8							

Bits	Description	
[31:27]	**Reserved**	Reserved.
[26:16]	**ERRADDR9**	**ECC Error Address First Field of Error 9** This field contains an 11-bit ECC error address 9 of first field. If it is a correctable error, please read the error data, ERRDATA9 (NFI_NANDECCED2[15:8]), to correct this error.
[15:11]	**Reserved**	Reserved.
[10:0]	**ERRADDR8**	**ECC Error Address First Field of Error 8** This field contains an 11-bit ECC error address 8 of first field. If it is a correctable error, please read the error data, ERRDATA8 (NFI_NANDECCED2[7:0]), to correct this error.

NAND Flash ECC Error Byte Address 5 Register (NFI_NANDECCEA5)

Register	Offset	R/W	Description	Reset Value
NFI_NANDECCEA5	NFI_BA+0x914	R	NAND Flash ECC Error Byte Address 5 Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					ERRADDR11		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ERRADDR11							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					ERRADDR10		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ERRADDR10							

Bits	Description	
[31:27]	**Reserved**	Reserved.
[26:16]	**ERRADDR11**	**ECC Error Address First Field of Error 11** This field contains an 11-bit ECC error address 11 of first field. If it is a correctable error, please read the error data, ERRDATA11 (NFI_NANDECCED2[31:24]), to correct this error.
[15:11]	**Reserved**	Reserved.
[10:0]	**ERRADDR10**	**ECC Error Address First Field of Error 10** This field contains an 11-bit ECC error address 10 of first field. If it is a correctable error, please read the error data, ERRDATA10 (NFI_NANDECCED2[23:16]), to correct this error.

NAND Flash ECC Error Byte Address 6 Register (NFI_NANDECCEA6)

Register	Offset	R/W	Description	Reset Value
NFI_NANDECCEA6	NFI_BA+0x918	R	NAND Flash ECC Error Byte Address 6 Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					**ERRADDR13**		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ERRADDR13							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					**ERRADDR12**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ERRADDR12							

Bits	Description	
[31:27]	**Reserved**	Reserved.
[26:16]	**ERRADDR13**	**ECC Error Address First Field of Error 13** This field contains an 11-bit ECC error address 13 of first field. If it is a correctable error, please read the error data, ERRDATA13 (NFI_NANDECCED3[15:8]), to correct this error.
[15:11]	**Reserved**	Reserved.
[10:0]	**ERRADDR12**	**ECC Error Address First Field of Error 12** This field contains an 11-bit ECC error address 12 of first field. If it is a correctable error, please read the error data, ERRDATA12 (NFI_NANDECCED3[7:0]), to correct this error.

NAND Flash ECC Error Byte Address 7 Register (NFI_NANDECCEA7)

Register	Offset	R/W	Description	Reset Value
NFI_NANDECCEA7	NFI_BA+0x91C	R	NAND Flash ECC Error Byte Address 7 Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					**ERRADDR15**		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ERRADDR15							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					**ERRADDR14**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ERRADDR14							

Bits	Description	
[31:27]	**Reserved**	Reserved.
[26:16]	**ERRADDR15**	**ECC Error Address First Field of Error 15** This field contains an 11-bit ECC error address 15 of first field. If it is a correctable error, please read the error data, ERRDATA15 (NFI_NANDECCED3[31:24]), to correct this error.
[15:11]	**Reserved**	Reserved.
[10:0]	**ERRADDR14**	**ECC Error Address First Field of Error 14** This field contains an 11-bit ECC error address 14 of first field. If it is a correctable error, please read the error data, ERRDATA14 (NFI_NANDECCED3[23:16]), to correct this error.

NAND Flash ECC Error Byte Address 8 Register (NFI_NANDECCEA8)

Register	Offset	R/W	Description	Reset Value
NFI_NANDECCEA8	NFI_BA+0x920	R	NAND Flash ECC Error Byte Address 8 Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					**ERRADDR17**		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ERRADDR17							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					**ERRADDR16**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ERRADDR16							

Bits	Description	
[31:27]	**Reserved**	Reserved.
[26:16]	**ERRADDR17**	**ECC Error Address First Field of Error 17** This field contains an 11-bit ECC error address 17 of first field. If it is a correctable error, please read the error data, ERRDATA17 (NFI_NANDECCED4[15:8]), to correct this error.
[15:11]	**Reserved**	Reserved.
[10:0]	**ERRADDR16**	**ECC Error Address First Field of Error 16** This field contains an 11-bit ECC error address 16 of first field. If it is a correctable error, please read the error data, ERRDATA16 (NFI_NANDECCED4[7:0]), to correct this error.

NAND Flash ECC Error Byte Address 9 Register (NFI_NANDECCEA9)

Register	Offset	R/W	Description	Reset Value
NFI_NANDECCEA9	NFI_BA+0x924	R	NAND Flash ECC Error Byte Address 9 Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					**ERRADDR19**		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ERRADDR19							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					**ERRADDR18**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ERRADDR18							

Bits	Description	
[31:27]	**Reserved**	Reserved.
[26:16]	**ERRADDR19**	**ECC Error Address First Field of Error 19** This field contains an 11-bit ECC error address 19 of first field. If it is a correctable error, please read the error data, ERRDATA19 (NFI_NANDECCED4[31:24]), to correct this error.
[15:11]	**Reserved**	Reserved.
[10:0]	**ERRADDR18**	**ECC Error Address First Field of Error 18** This field contains an 11-bit ECC error address 18 of first field. If it is a correctable error, please read the error data, ERRDATA18 (NFI_NANDECCED4[23:16]), to correct this error.

NAND Flash ECC Error Byte Address 10 Register (NFI_NANDECCEA10)

Register	Offset	R/W	Description	Reset Value
NFI_NANDECCEA10	NFI_BA+0x928	R	NAND Flash ECC Error Byte Address 10 Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					ERRADDR21		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ERRADDR21							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					ERRADDR20		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ERRADDR20							

Bits	Description	
[31:27]	**Reserved**	Reserved.
[26:16]	**ERRADDR21**	**ECC Error Address First Field of Error 21** This field contains an 11-bit ECC error address 21 of first field. If it is a correctable error, please read the error data, ERRDATA21 (NFI_NANDECCED5[15:8]), to correct this error.
[15:11]	**Reserved**	Reserved.
[10:0]	**ERRADDR20**	**ECC Error Address First Field of Error 20** This field contains an 11-bit ECC error address 20 of first field. If it is a correctable error, please read the error data, ERRDATA20 (NFI_NANDECCED5[7:0]), to correct this error.

NAND Flash ECC Error Byte Address 11 Register (NFI_NANDECCEA11)

Register	Offset	R/W	Description	Reset Value
NFI_NANDECCEA11	NFI_BA+0x92C	R	NAND Flash ECC Error Byte Address 11 Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					ERRADDR23		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ERRADDR23							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					ERRADDR22		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ERRADDR22							

Bits	Description	
[31:27]	**Reserved**	Reserved.
[26:16]	**ERRADDR23**	**ECC Error Address First Field of Error 23** This field contains an 11-bit ECC error address 23 of first field. If it is a correctable error, please read the error data, ERRDATA23 (NFI_NANDECCED5[31:24]), to correct this error.
[15:11]	**Reserved**	Reserved.
[10:0]	**ERRADDR22**	**ECC Error Address First Field of Error 22** This field contains an 11-bit ECC error address 22 of first field. If it is a correctable error, please read the error data, ERRDATA22 (NFI_NANDECCED5[23:16]), to correct this error.

NAND Flash ECC Error Data Register 0 (NFI_NANDECCED0)

Register	Offset	R/W	Description	Reset Value
NFI_NANDECCED0	NFI_BA+0x960	R	NAND Flash ECC Error Data Register 0	0x8080_8080

31	30	29	28	27	26	25	24
ERRDATA3							
23	22	21	20	19	18	17	16
ERRDATA2							
15	14	13	12	11	10	9	8
ERRDATA1							
7	6	5	4	3	2	1	0
ERRDATA0							

Bits	Description	
[31:24]	**ERRDATA3**	**ECC Error Data of First Field 3** This field contains an 8-bit BCH ECC error data 3 of first field. If it is a correctable error, please read out the error data in this field and doing bitwise XOR with received data locating at address ERRADDR3 (NFI_NANDECCEA1[26:16]), and then the result will be the correct data.
[23:16]	**ERRDATA2**	**ECC Error Data of First Field 2** This field contains an 8-bit BCH ECC error data 2 of first field. If it is a correctable error, please read out the error data in this field and doing bitwise XOR with received data locating at address ERRADDR2 (NFI_NANDECCEA1[10:0]), and then the result will be the correct data.
[15:8]	**ERRDATA1**	**ECC Error Data of First Field 1** This field contains an 8-bit BCH ECC error data 1 of first field. If it is a correctable error, please read out the error data in this field and doing bitwise XOR with received data locating at address ERRADDR1 (NFI_NANDECCEA0[26:16]), and then the result will be the correct data.
[7:0]	**ERRDATA0**	**ECC Error Data of First Field 0** This field contains an 8-bit BCH ECC error data 0 of first field. If it is a correctable error, please read out the error data in this field and doing bitwise XOR with received data locating at address ERRADDR0 (NFI_ANNDECCEA0[10:0]), and then the result will be the correct data.

NAND Flash ECC Error Data Register 1 (NFI_NANDECCED1)

Register	Offset	R/W	Description	Reset Value
NFI_NANDECCED1	NFI_BA+0x964	R	NAND Flash ECC Error Data Register 1	0x8080_8080

31	30	29	28	27	26	25	24
ERRDATA7							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ERRDATA6							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ERRDATA5							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ERRDATA4							

Bits	Description	
[31:24]	**ERRDATA7**	**ECC Error Data of First Field 7** This field contains an 8-bit BCH ECC error data 7 of first field. If it is a correctable error, please read out the error data in this field and doing bitwise XOR with received data locating at address ERRADDR7 (NFI_NANDECCEA3[26:16]), and then the result will be the correct data.
[23:16]	**ERRDATA6**	**ECC Error Data of First Field 6** This field contains an 8-bit BCH ECC error data 6 of first field. If it is a correctable error, please read out the error data in this field and doing bitwise XOR with received data locating at address ERRADDR6 (NFI_NANDECCEA3[10:0]), and then the result will be the correct data.
[15:8]	**ERRDATA5**	**ECC Error Data of First Field 5** This field contains an 8-bit BCH ECC error data 5 of first field. If it is a correctable error, please read out the error data in this field and doing bitwise XOR with received data locating at address ERRADDR5 (NFI_NANDECCEA2[26:16]), and then the result will be the correct data.
[7:0]	**ERRDATA4**	**ECC Error Data of First Field 4** This field contains an 8-bit BCH ECC error data 4 of first field. If it is a correctable error, please read out the error data in this field and doing bitwise XOR with received data locating at address ERRADDR4 (NFI_NANDECCEA2[10:0]), and then the result will be the correct data.

NAND Flash ECC Error Data Register 2 (NFI_NANDECCED2)

Register	Offset	R/W	Description	Reset Value
NFI_NANDECCED2	NFI_BA+0x968	R	NAND Flash ECC Error Data Register 2	0x8080_8080

31	30	29	28	27	26	25	24
ERRDATA11							
23	22	21	20	19	18	17	16
ERRDATA10							
15	14	13	12	11	10	9	8
ERRDATA9							
7	6	5	4	3	2	1	0
ERRDATA8							

Bits	Description	
[31:24]	**ERRDATA11**	**ECC Error Data of First Field 11** This field contains an 8-bit BCH ECC error data 11 of first field. If it is a correctable error, please read out the error data in this field and doing bitwise XOR with received data locating at address ERRADDR11 (NFI_NANDECCEA5[26:16]), and then the result will be the correct data.
[23:16]	**ERRDATA10**	**ECC Error Data of First Field 10** This field contains an 8-bit BCH ECC error data 10 of first field. If it is a correctable error, please read out the error data in this field and doing bitwise XOR with received data locating at address ERRADDR10 (NFI_NANDECCEA5[10:0]), and then the result will be the correct data.
[15:8]	**ERRDATA9**	**ECC Error Data of First Field 9** This field contains an 8-bit BCH ECC error data 9 of first field. If it is a correctable error, please read out the error data in this field and doing bitwise XOR with received data locating at address ERRADDR9 (NFI_NANDECCEA4[26:16]), and then the result will be the correct data.
[7:0]	**ERRDATA8**	**ECC Error Data of First Field 8** This field contains an 8-bit BCH ECC error data 8 of first field. If it is a correctable error, please read out the error data in this field and doing bitwise XOR with received data locating at address ERRADDR8 (NFI_NANDECCEA4[10:0]), and then the result will be the correct data.

NAND Flash ECC Error Data Register 3 (NFI_NANDECCED3)

Register	Offset	R/W	Description	Reset Value
NFI_NANDECCED3	NFI_BA+0x96C	R	NAND Flash ECC Error Data Register 3	0x8080_8080

31	30	29	28	27	26	25	24
ERRDATA15							
23	22	21	20	19	18	17	16
ERRDATA14							
15	14	13	12	11	10	9	8
ERRDATA13							
7	6	5	4	3	2	1	0
ERRDATA12							

Bits	Description	
[31:24]	**ERRDATA15**	**ECC Error Data of First Field 15** This field contains an 8-bit BCH ECC error data 15 of first field. If it is a correctable error, please read out the error data in this field and doing bitwise XOR with received data locating at address ERRADDR15 (NFI_NANDECCEA7[26:16]), and then the result will be the correct data.
[23:16]	**ERRDATA14**	**ECC Error Data of First Field 14** This field contains an 8-bit BCH ECC error data 14 of first field. If it is a correctable error, please read out the error data in this field and doing bitwise XOR with received data locating at address ERRADDR14 (NFI_NANDECCEA7[10:0]), and then the result will be the correct data.
[15:8]	**ERRDATA13**	**ECC Error Data of First Field 13** This field contains an 8-bit BCH ECC error data 13 of first field. If it is a correctable error, please read out the error data in this field and doing bitwise XOR with received data locating at address ERRADDR13 (NFI_NANDECCEA6[26:16]), and then the result will be the correct data.
[7:0]	**ERRDATA12**	**ECC Error Data of First Field 12** This field contains an 8-bit BCH ECC error data 12 of first field. If it is a correctable error, please read out the error data in this field and doing bitwise XOR with received data locating at address ERRADDR12 (NFI_NANDECCEA6[10:0]), and then the result will be the correct data.

NAND Flash ECC Error Data Register 4 (NFI_NANDECCED4)

Register	Offset	R/W	Description	Reset Value
NFI_NANDECCED4	NFI_BA+0x970	R	NAND Flash ECC Error Data Register 4	0x8080_8080

31	30	29	28	27	26	25	24
ERRDATA19							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ERRDATA18							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ERRDATA17							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ERRDATA16							

Bits	Description	
[31:24]	**ERRDATA19**	**ECC Error Data of First Field 19** This field contains an 8-bit BCH ECC error data 19 of first field. If it is a correctable error, please read out the error data in this field and doing bitwise XOR with received data locating at address ERRADDR19 (NFI_NANDECCEA9[26:16]), and then the result will be the correct data.
[23:16]	**ERRDATA18**	**ECC Error Data of First Field 18** This field contains an 8-bit BCH ECC error data 18 of first field. If it is a correctable error, please read out the error data in this field and doing bitwise XOR with received data locating at address ERRADDR18 (NFI_NANDECCEA9[10:0]), and then the result will be the correct data.
[15:8]	**ERRDATA17**	**ECC Error Data of First Field 17** This field contains an 8-bit BCH ECC error data 17 of first field. If it is a correctable error, please read out the error data in this field and doing bitwise XOR with received data locating at address ERRADDR17 (NFI_NANDECCEA8[26:16]), and then the result will be the correct data.
[7:0]	**ERRDATA16**	**ECC Error Data of First Field 16** This field contains an 8-bit BCH ECC error data 16 of first field. If it is a correctable error, please read out the error data in this field and doing bitwise XOR with received data locating at address ERRADDR16 (NFI_NANDECCEA8[10:0]), and then the result will be the correct data.

NAND Flash ECC Error Data Register 5 (NFI_NANDECCED5)

Register	Offset	R/W	Description	Reset Value
NFI_NANDECCED5	NFI_BA+0x974	R	NAND Flash ECC Error Data Register 5	0x8080_8080

31	30	29	28	27	26	25	24
ERRDATA23							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ERRDATA22							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ERRDATA21							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ERRDATA20							

Bits	Description	
[31:24]	**ERRDATA23**	**ECC Error Data of First Field 23** This field contains an 8-bit BCH ECC error data 23 of first field. If it is a correctable error, please read out the error data in this field and doing bitwise XOR with received data locating at address ERRADDR23 (NFI_NANDECCEA11[26:16]), and then the result will be the correct data.
[23:16]	**ERRDATA22**	**ECC Error Data of First Field 22** This field contains an 8-bit BCH ECC error data 22 of first field. If it is a correctable error, please read out the error data in this field and doing bitwise XOR with received data locating at address ERRADDR22 (NFI_NANDECCEA11[10:0]), and then the result will be the correct data.
[15:8]	**ERRDATA21**	**ECC Error Data of First Field 21** This field contains an 8-bit BCH ECC error data 21 of first field. If it is a correctable error, please read out the error data in this field and doing bitwise XOR with received data locating at address ERRADDR21 (NFI_NANDECCEA10[26:16]), and then the result will be the correct data.
[7:0]	**ERRDATA20**	**ECC Error Data of First Field 20** This field contains an 8-bit BCH ECC error data 20 of first field. If it is a correctable error, please read out the error data in this field and doing bitwise XOR with received data locating at address ERRADDR20 (NFI_NANDECCEA10[10:0]), and then the result will be the correct data.

NAND Flash Redundant Area Word n (NFI_NANDRAn)

Register	Offset	R/W	Description	Reset Value
NFI_NANDRAn **n = 0, 1..117**	NFI_BA+0xA00+0x4*n	R/W	NAND Flash Redundant Area Word n	Undefined

31	30	29	28	27	26	25	24
Data							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Data							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Data							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Data							

Bits	Description	
[31:0]	**Data**	**NAND Flash Redundant Area Word n** This field indicates a 32-bit data of redundant area.

6.37 Cryptographic Accelerator (CRYPTO)

6.37.1 Overview

The Crypto (Cryptographic Accelerator) includes a secure pseudo random number generator (PRNG) core and supports AES, SHA/HMAC, RSA, and ECC algorithms.

The PRNG core supports 128, 163, 192, 224, 233, 255, 256, 283, 384, 409, 512, 521 and 571 bits random number generation. (283~571 bits are only generated for Key Store).

The AES accelerator is an implementation fully compliant with the AES (Advance Encryption Standard) encryption and decryption algorithm. The AES accelerator supports ECB, CBC, CFB, OFB, CTR, CBC-CS1, CBC-CS2, CBC-CS3, CCM and GCM mode.

The SHA accelerator is an implementation fully compliant with the MD5, SM3, SHA-160, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/t, SHA3 and corresponding HMAC (Keyed-Hash Message Authentication Code) algorithms, except for SHAKE128 and SHAKE256.

The ECC accelerator is an implementation fully compliant with elliptic curve cryptography by using polynomial basis in binary field and prime filed.

The RSA accelerator is an implementation fully compliant with RSA cryptography, CRT decryption algorithm, side-channel attack countermeasures and CRT fault injection attack countermeasure algorithm.

The Crypto can get key from Key Store and/or put the key to Key Store determined by the function of each accelerator.

The Crypto supports one technique to improve power analysis protection ability.

6.37.2 Features

- PRNG
  - Supports 128, 163, 192, 224, 233, 255, 256, 283, 384, 409, 512, 521 and 571 bits random number generation (283~571 bits are only generated for Key Store).
  - Can take the true random number seed from TRNG
  - Can take the true random number from TRNG (only for Key Store)
- AES
  - Supports FIPS NIST 197
  - Supports SP800-38A and addendum
  - Supports 128, 192, and 256 bits key
  - Supports both encryption and decryption
  - Supports ECB, CBC, CFB, OFB, CTR, CBC-CS1, CBC-CS2 and CBC-CS3 modes
  - Supports CCM mode, GCM mode and GMAC function
  - Supports SM4 block cipher algorithm
  - Supports key expander for key expansion in FIPS NIST 197
  - Supports three techniques to improve side-channel attack protection ability
  - Supports five techniques to improve advance side-channel attack protection ability for differential fault analysis (DFA), collision-correlation attack (CCA), and template based attack (TA)
- SHA
  - Supports FIPS NIST 180, 180-2, 180-4

- Supports MD5
- Supports SHA-160, SHA-224, SHA-256, SHA-384, SHA-512 and SHA-512/t
- Supports SHA3-224, SHA3-256, SHA3-384, SHA3-512, SHAKE128 and SHAKE256
- Supports SM3 Cryptographic Hash Algorithm

- HMAC
  - Supports FIPS NIST 180, 180-2, 180-4
  - Supports HMAC-MD5
  - Supports HMAC-SHA-160, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512
  - Supports HMAC-SHA3-224, HMAC-SHA3-256, HMAC-SHA3-384, and HMAC-SHA3-512
- ECC
  - Supports both prime field GF(p) and binary filed GF(2^m)
  - Supports NIST P-192, P-224, P-256, P-384, and P-521
  - Supports NIST B-163, B-233, B-283, B-409, and B-571
  - Supports NIST K-163, K-233, K-283, K-409, and K-571
  - Supports Curve25519
  - Supports Public Key Cryptographic Algorithm SM2 Based on Elliptic Curves
  - Supports point multiplication, addition and doubling operations in GF(p) and GF(2^m)
  - Supports modulus division, multiplication, addition and subtraction operations in GF(p)
  - Supports three techniques to improve side-channel attack protection ability
  - Supports three techniques to improve advance side-channel attack protection ability for differential fault attack (DFA), differential power analysis (DPA) and correlation power analysis (CPA)
- RSA
  - Supports both encryption and decryption with 1024, 2048, 3072 and 4096 bits
  - Supports CRT decryption with 2048, 3072 and 4096 bits
  - Supports three techniques to improve side-channel attack protection ability
  - Supports one technique to improve CRT fault injection attack protection ability

6.38 Key Store (KS)

6.38.1 Overview

The Key Store (KS) is a key management device with 4 Kbytes SRAM and OTP for key storage. The Key Store is capable of providing a crypto engine to access or store the key while encryption, decryption and generation. The Key Store supports revoke key operation if key is unused. The Key Store is able to protect the key by data scrambling, data remanence prevention and silent access.

6.38.2 Features

- Supports programming interface for key management
- Supports key size required for Cryptography from 128 bits to 4096 bits
- Supports 32 SRAM keys by 4 Kbytes SRAM
- Supports 9 OTP keys of which key 0~2 are 128-bits and key 3~8 are 256-bits
- Supports crypto engine access or store key in Key Store directly
- Supports ECDH operation with ECC and PRNG engine
- Supports to store middle data for RSA CRT and SCAP mode
- Supports revoke operation
- Supports erase key in SRAM and revoke key in OTP while tamper detected
- Supports integrity checking
- Supports data scrambling
- Supports data remanence prevention at SRAM
- Supports silent access for side-channel protection at SRAM

6.39 Camera Capture Interface Controller (CCAP)

6.39.1 Overview

The camera capture interface controller (CCAP) is designed to capture image data from a sensor. After capturing or fetching image data, it will process the image data, and then the embedded DMA controller will move the data from the internal FIFO to system memory with AHB bus.

6.39.2 Features

- CCIR601 and CCIR656 and 4-bit interfaces supported for connection to CMOS image sensor
- 8-bit YUV422 and 8-bit RGB565 color format supported for data-in from CMOS sensor
- 8-bit/4-bit monochrome format supported for data-in from CMOS sensor
- 8-bit/10-bit raw Bayer format supported for data-in from CMOS sensor
- YUV422, RGB565, RGB555 and Y-only color supported for packet data output.
- Single interrupt source to interrupt controller from maskable interrupt source: Address Match, Bus Master Transfer Error, Video Frame End
- Embedded DMA controller supported to transfer data from internal FIFO to system memory through AHB bus
- Supports YUV422 and YUV420 planar data output
- CROP function supported to crop input image to the required size for digital application.
- Frame rate scaling-down supported
- Image scaling-down supported
- Bit luma output with 8-bit threshold setting supported.

6.39.3 Block Diagram

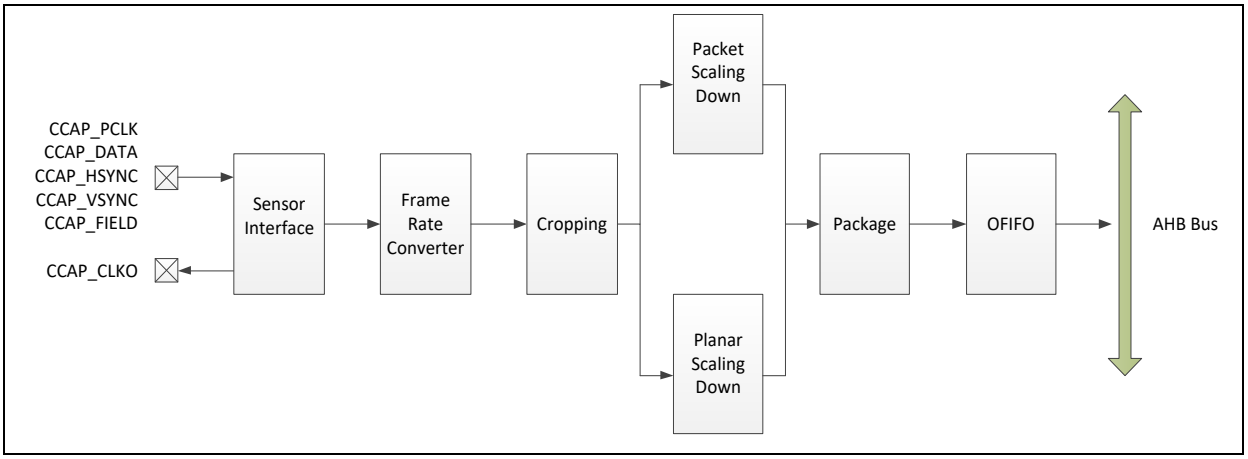


Figure 6.39-1 Camera Capture Interface Controller Functional Block Diagram

6.39.4 Basic Configuration

6.39.4.1 CCAP Pin Description

Name	BITs	I/O Type	Pin Description
CCAP_DATA	10	Input	Sensor pixel data input
CCAP_HSYNC	1	Input	Sensor horizontal synchronization input
CCAP_VSYNC	1	Input	Sensor vertical synchronization input
CCAP_PIXCLK	1	Input	Sensor pixel clock input (from sensor)
CCAP_SCLK	1	Output	Sensor pixel clock output (to sensor)

- The CCAP_DATA are synchronous with CCAP_PIXCLK. Set the register PCLKP (CCAP_PAR[8]) to change the polarity of the rising/falling edge.
- The CCAP_HSYNC signal indicates the start/end of a line. Set the HSP (CCAP_PAR[9]) to change its polarity.
- The CCAP_VSYNC signal indicates the start/end of a frame. Set the VSP (CCAP_PAR[10]) to change its polarity.
- The CCAP_SCLK signal indicates the pixel clock to the sensor.

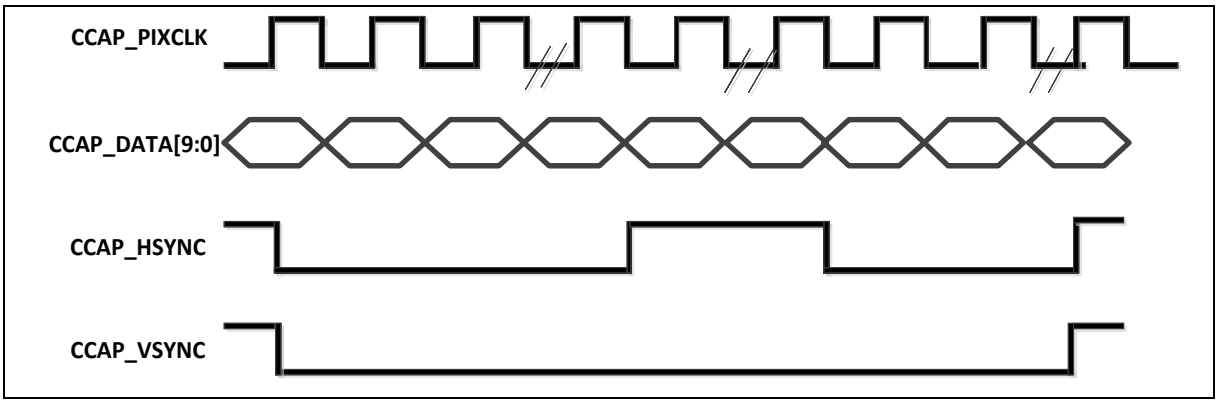


Figure 6.39-2 CCAP Basic Pin Diagram

6.39.4.2 CCAP Pin Configurations

Group	Pin Name	GPIO	MFP
CCAP0	CCAP0_SCLK (to Sensor)	PK.9	MFP6
	CCAP0_DATA0	PM.2	MFP6
	CCAP0_DATA1	PM.3	MFP6
	CCAP0_DATA2	PM.4	MFP6
	CCAP0_DATA3	PM.5	MFP6
	CCAP0_DATA4	PM.6	MFP6
	CCAP0_DATA5	PM.7	MFP6
	CCAP0_DATA6	PM.8	MFP6
	CCAP0_DATA7	PM.9	MFP6

	CCAP0_DATA8	PM.10	MFP6
	CCAP0_DATA9	PM.11	MFP6
	CCAP0_HSYNC	PK.11	MFP6
	CCAP0_PIXCLK (from Sensor)	PK.10	MFP6
	CCAP0_VSYNC	PM.0	MFP6
CCAP1	CCAP1_SCLK (to Sensor)	PN.10	MFP6
		PE.8	MFP7
	CCAP1_DATA0	PN.0	MFP6
		PE.0	MFP7
	CCAP1_DATA1	PN.1	MFP6
		PE.1	MFP7
	CCAP1_DATA2	PN.2	MFP6
		PE.2	MFP7
	CCAP1_DATA3	PN.3	MFP6
		PE.3	MFP7
	CCAP1_DATA4	PN.4	MFP6
		PE.4	MFP7
	CCAP1_DATA5	PN.5	MFP6
		PE.5	MFP7
	CCAP1_DATA6	PN.6	MFP6
		PE.6	MFP7
	CCAP1_DATA7	PN.7	MFP6
		PE.7	MFP7
	CCAP1_DATA8	PE.12	MFP6
	CCAP1_DATA9	PE.13	MFP6
	CCAP1_HSYNC	PN.12	MFP6
		PE.10	MFP7
		PB.9	MFP7
	CCAP1_PIXCLK (from Sensor)	PN.11	MFP6
		PE.9	MFP7
	CCAP1_VSYNC	PN.13	MFP6
		PE.11	MFP7
		PB.10	MFP7

6.39.5 Functional Description

6.39.5.1 Image Capture Flow

Configure these settings to capture image data from a sensor to system memory by CCAP function.

- Configure the GPIO to set the MFP.
- Configure the clocks to enable the clock of the CCAP.
- Configure the related control register according to input CMOS sensor spec
- Configure the interrupt related control register to enable or mask the interrupt.
- Configure the embedded DMA related control register to let DMA move the data to the system memory
- Configure the control register about CCAP feature (Cropping, One shutter mode, Luminance 8-bit to 1-bit conversion, frame rate scaling down, packet vertical/horizontal scaling down, color control)
- Configure the control register to enable the capture function and packet/planar output

6.39.5.2 Supported Sensor Type and Output Format

The CCAP supports two different sensors: Monochrome/raw Bayer CMOS sensor and Color CMOS sensor. User can set the MONO(CCAP_CTL [7]) according to the input sensor .

The CCAP supports two different sensor input formats (YCbCr422, RGB565). User can set the INFMT(CCAP_PAR[0]) according to the input sensor format.

The CCAP supports two different sensor input types (CCIR601, CCIR656). User can set the SENTYPE(CCAP_PAR[1]) according to the input sensor type.

The CCAP supports different sensor input data orders. User can set the INDATORD(CCAP_PAR [3:2]) according to the input sensor order.

INFMT(CCAP_PAR[0])	INDATORD(CCAP_PAR[3:2])	Sensor Input Data Order
0 (YCbCr)	0	Byte 0 1 2 3 : Y0 U0 Y1 V0
0 (YCbCr)	1	Byte 0 1 2 3 : Y0 V0 Y1 U0
0 (YCbCr)	2	Byte 0 1 2 3 : U0 Y0 V0 Y1
0 (YCbCr)	3	Byte 0 1 2 3 : V0 Y0 U0 Y1
1 (RGB565)	0	Byte 0 1 :{R[4:0],G[5:3]} {G[2:0],B[4:0]}
1 (RGB565)	1	Byte 0 1 :{B[4:0],G[5:3]} {G[2:0],R[4:0]}
1 (RGB565)	2	Byte 0 1 :{G[2:0],B[4:0]} {R[4:0],G[5:3]}
1 (RGB565)	3	Byte 0 1 :{G[2:0],R[4:0]} {B[4:0],G[5:3]}

The CCAP supports different sensor signal polarities (VSYNC,HSYNC,PCLKP). User can set the VSP(CCAP_PAR [10]), HSP(CCAP_PAR[9]) , PLKP(CCAP_PAR[8]) according to the input sensor polarity.

For Monochrome/raw Bayer CMOS sensor, the CCAP supports three different sensor data I/O interface and data sequence. User can set the BAYER_10(CCAP_CTL [21]), MY8_MY4(CCAP_CTL[18]) and MY4_SWAP(CCAP_CTL[17]) according to the input sensor.

The CCAP supports four different output formats to system memory (YCbCr422, only output Y,RGB555,RGB565). User can set the OUTFMT(CCAP_PAR [5:4]) according to the user need.

User can set the UPDATE(CCAP_CTL [24]) register .If this value is set to 1,it will update these register

when new frame starts.

6.39.5.3 CCAP Interrupt Source Enable and Mask

The CCAP supports three interrupt sources:

1. Address Match Interrupt: When Current Packet System Memory Address Register (CCAP_CURADDRP) is equal to the Compare Memory Address Register (CCAP_CMPADDR) , it will trigger this interrupt.
2. Bus Master Transfer Error Interrupt: When CCAP issues a request to the AHB bus, and get an error response from bus. It will trigger this interrupt.
3. Video Frame End Interrupt: When the end of frame capture, it will trigger this interrupt.

User can set the Camera Capture Interface Interrupt Register (CCAP_INT) to enable or mask these interrupts.

6.39.5.4 Data Transfer with the Embedded DMA Controller

The CCAP has an embedded DMA controller to transfer the data from CCAP internal FIFO to outside system memory. User can set the CCAP_PKTBA0 control register to decide the system memory base address, and set the PKTSTRIDE to decide the pixel stride width between two lines. User also can read the CCAP_CURADDRP to know the status about which address will be written by CCAP.

User can set the threshold value by PKTFTH (CCAP_FIFOTH [28:24]).When the number of the words internal FIFO received are greater than the PKTFTH value, it will trigger the embedded DMA to move data to system memory. User can read OVF(CCAP_FIFOTH [31]) to know whether the internal FIFO is overflow or not.

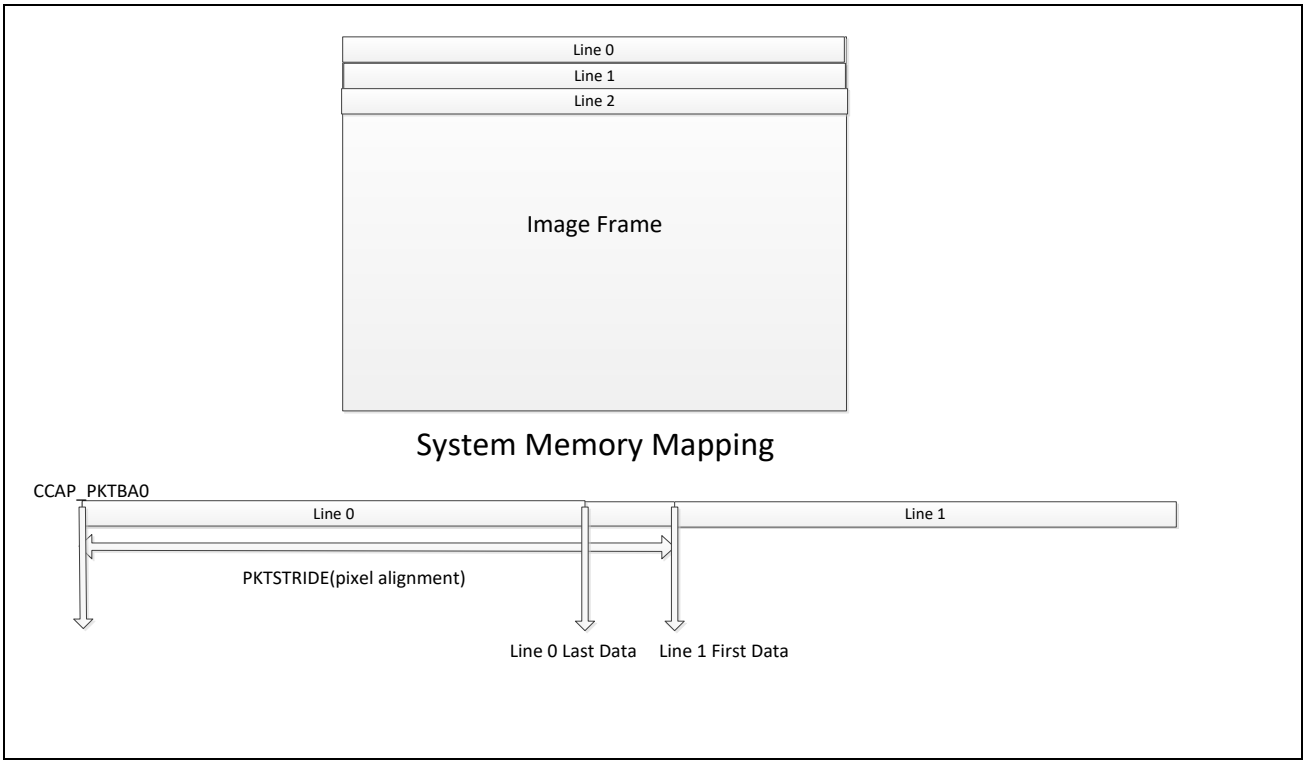


Figure 6.39-3 System Memory Mapping Block

6.39.5.5 Cropping Feature

The capture interface can select a window from the received image. The size of the window is specified by the number of pixel clocks (horizontal dimension) and the number of lines (vertical dimension). The

start (left upper corner) coordinates can be specified by the CCAP_CWSP register. The size (vertical dimension in number of lines and horizontal dimension in number of pixel clocks) can be specified by the CCAP_CWS register.

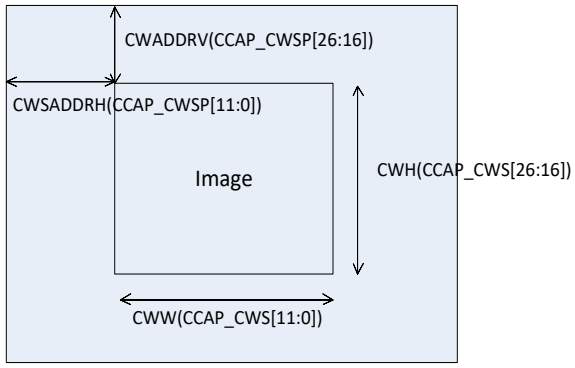


Figure 6.39-4 Image Start and Size of the Window after Cropping Block

6.39.5.6 One Shutter Mode (Single Frame)

In one shutter mode, a single frame is captured. When the SHUTTER (CCAP_CTL [16]) bit is set, the Image Capture interface automatically disables the capture interface after a frame is captured.

6.39.5.7 8-bit/10-bit Raw Bayer format feature

In 10-bit raw Bayer format mode, user needs to set BAYER_10(CCAP_CTL[21]) and MONO (CCAP_CTL[7]) to 1. The output format OUTFMT(CCAP_PAR [5:4]) needs to be set to 1.

In 8-bit raw Bayer format mode, user need to set BAYER_10(CCAP_CTL [21]) to 0, MY8_MY4(CCAP_CTL[18]) to 1, and MONO (CCAP_CTL[7]) to 1. The output format OUTFMT(CCAP_PAR [5:4]) needs to be set to 1.

- 10-bit raw Bayer output format

31:26	25:16	15:10	9:0
0	D1[9:0]	0	D0[9:0]

- 8-bit raw Bayer output format

31:24	23:16	15:8	7:0
D3[7:0]	D2[7:0]	D1[7:0]	D0[7:0]

P.S. Raw Bayer format does not support the features of cropping and scaling down. User needs to set CWH(CCAP_CWS [26:16]) and CWW(CCAP_CWS[11:0]) as the original picture size.

6.39.5.8 Luminance 8-bit to 1-bit Conversion Feature

When Luma_Y_one (CCAP_CTL [19]) bit is set, it will convert Y-8-bit to Y-1-bit according to the LUMA_Y1_THRESH (CCAP_LUMA_Y1_THD[7:0]). When the Y-8-bit value is greater than LUMA_Y1_THRESH, it will output 1. Otherwise, it will output 0.

6.39.5.9 Frame Rate Scaling Down Feature

The CCAP can set the FRM(CCAP_FRCTL [5:0]) and FRN(CCAP_FRCTL[13:8]) to support frame rate scaling down. The output image frame rate will be equal to input image format *(FRN/FRM)

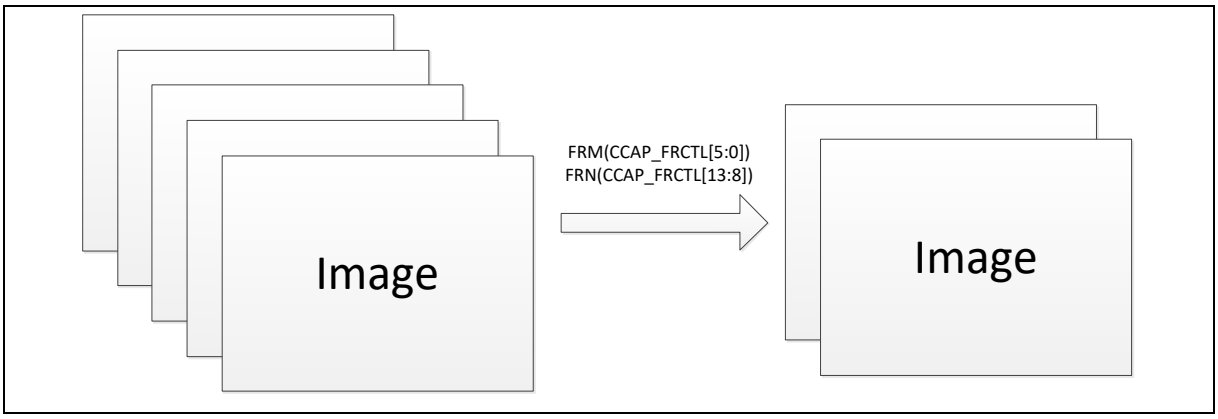


Figure 6.39-5 Frame Rate Scaling Down Block

6.39.5.10 Packet Vertical/Horizontal Scaling Down Feature

The CCAP can set the CCAP_PKTSM and CCAP_PKTSL control register to scale down the vertical/horizontal of input image frame.

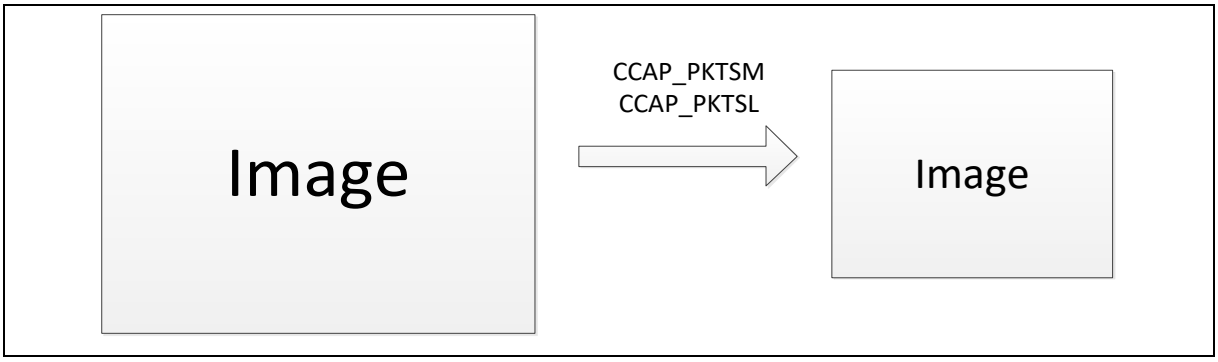


Figure 6.39-6 Frame Packet Scaling Down Block

6.39.5.11 Enable the Capture Interface

User can set the CCAPEN(CCAP_CTL [0]) to enable the capture interface, and set the PKTEN (CCAP_CTL[6]) , PLNEN (CCAP_CTL[5]) to enable the packet/planar output.

6.39.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write.

Register	Offset	R/W	Description	Reset Value
CCAP Base Address: **CCAPx_BA = 0x4024_0000 + (0x0001_0000*x)** **x=0,1**				
CCAPx_CTL	CCAPx_BA+0x00	R/W	Camera Capture Interface Control Register	0x0000_0040
CCAPx_PAR	CCAPx_BA+0x04	R/W	Camera Capture Interface Parameter Register	0x0000_0000
CCAPx_INT	CCAPx_BA+0x08	R/W	Camera Capture Interface Interrupt Register	0x0000_0000
CCAPx_POSTERIZE	CCAPx_BA+0x0C	R/W	YUV Component Posterizing Factor Register	0x0000_0000
CCAPx_SEPIA	CCAPx_BA+0x1C	R/W	Sepia Effect Control Register	0x0000_0000
CCAPx_CWSP	CCAPx_BA+0x20	R/W	Cropping Window Starting Address Register	0x0000_0000
CCAPx_CWS	CCAPx_BA+0x24	R/W	Cropping Window Size Register	0x0000_0000
CCAPx_PKTSL	CCAPx_BA+0x28	R/W	Packet Scaling Vertical/Horizontal Factor Register (LSB)	0x0000_0000
CCAPx_PLNSL	CCAPx_BA+0x2C	R/W	Planar Scaling Vertical/Horizontal Factor Register (LSB)	0x0000_0000
CCAPx_FRCTL	CCAPx_BA+0x30	R/W	Scaling Frame Rate Factor Register	0x0000_0000
CCAPx_STRIDE	CCAPx_BA+0x34	R/W	Frame Output Pixel Stride Width Register	0x0000_0000
CCAPx_FIFOTH	CCAPx_BA+0x3C	R/W	FIFO Threshold Register	0x070D_0507
CCAPx_CMPADDR	CCAPx_BA+0x40	R/W	Compare Memory Base Address Register	0xFFFF_FFFC
CCAPx_LUMA_Y1_THD	CCAPx_BA+0x44	R/W	Luminance Y8 to Y1 Threshold Value Register	0x0000_0080
CCAPx_PKTSM	CCAPx_BA+0x48	R/W	Packet Scaling Vertical/Horizontal Factor Register (MSB)	0x0000_0000
CCAPx_PLNSM	CCAPx_BA+0x4C	R/W	Planar Scaling Vertical/Horizontal Factor Register (MSB)	0x0000_0000
CCAPx_CURADDRP	CCAPx_BA+0x50	R	Current Packet System Memory Address Register	0x0000_0000
CCAPx_CURADDRY	CCAPx_BA+0x54	R	Current Planar Y System Memory Address Register	0x0000_0000
CCAPx_CURADDRU	CCAPx_BA+0x58	R	Current Planar U System Memory Address Register	0x0000_0000
CCAPx_CURADDRV	CCAPx_BA+0x5C	R	Current Planar V System Memory Address Register	0x0000_0000
CCAPx_PKTBA0	CCAPx_BA+0x60	R/W	System Memory Packet Base Address 0 Register	0x0000_0000
CCAPx_YBA	CCAPx_BA+0x80	R/W	System Memory Planar Y Base Address Register	0x0000_0000
CCAPx_UBA	CCAPx_BA+0x84	R/W	System Memory Planar U Base Address Register	0x0000_0000
CCAPx_VBA	CCAPx_BA+0x88	R/W	System Memory Planar V Base Address Register	0x0000_0000

6.39.7 Register Description

Camera Capture Interface Control Register (CCAPx_CTL)

Register	Offset	R/W	Description	Reset Value
CCAPx_CTL	CCAPx_BA+0x00	R/W	Camera Capture Interface Control Register	0x0000_0040

31	30	29	28	27	26	25	24
Reserved							VPRST
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		BAYER10	UPDATE	LumaYOne	MY8MY4	MY4SWAP	SHUTTER
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MONO	PKTEN	PLNEN	Reserved				CCAPEN

Bits	Description	
[31:25]	**Reserved**	Reserved.
[24]	**VPRST**	**Capture Interface Reset** 0 = Capture interface reset Disabled. 1 = Capture interface reset Enabled.
[23:22]	**Reserved**	Reserved.
[21]	**BAYER10**	**Bayer Format for 10 Bit** 0 = depends on the MY8_MY4 register. 1 = 10 bit Bayer format.
[20]	**UPDATE**	**Update Register at end Frame** 0 = Update register at new frame Disabled. 1 = Update register at new frame Enabled (Auto clear to 0 when register updated).
[19]	**LumaYOne**	**Color/Monochrome CMOS Sensor Luminance 8-bit Y to 1-bit Y Conversion** 0 = Color/Monochrome CMOS sensor Luma-Y-One bit Disabled. 1 = Color/Monochrome CMOS sensor Luma-Y-One bit Enabled. **Note:** Color CMOS sensor U/V components are ignored when the Luma_Y_One is enabled.
[18]	**MY8MY4**	**Monochrome CMOS Sensor Data I/O Interface** 0 = Monochrome CMOS sensor is by the 4-bit data I/O interface. 1 = Monochrome/raw Bayer CMOS sensor is by the 8-bit data I/O interface.
[17]	**MY4SWAP**	**Monochrome CMOS Sensor 4-bit Data Nibble Swap** 0 = The 4-bit data input sequence: 1st Pixel is for 1st Nibble (1st pixel @MSB). 1 = The 4-bit data input sequence: 1st Pixel is for 2nd Nibble (1st pixel @LSB).
[16]	**SHUTTER**	**Camera Capture Interface Automatically Disables the Capture Interface after a Frame Had Been Captured** 0 = Shutter Disabled.

		1 = Shutter Enabled.
[15:8]	**Reserved**	Reserved.
[7]	**MONO**	**Monochrome/Raw Bayer CMOS Sensor Select** 0 = Color CMOS Sensor. 1 = Monochrome/raw Bayer CMOS Sensor. The U/V components are ignored when the MONO is enabled.
[6]	**PKTEN**	**Packet Output Enable Bit** 0 = Packet output Disabled. 1 = Packet output Enabled.
[5]	**PLNEN**	**Planar Output Enable Bit** 0 = Planar output Disabled. 1 = Planar output Enabled.
[4:1]	**Reserved**	Reserved.
[0]	**CCAPEN**	**Camera Capture Interface Enable Bit** 0 = Camera Capture Interface Disabled. 1 = Camera Capture Interface Enabled.

Camera Capture Interface Parameter Register (CCAPx_PAR)

Register	Offset	R/W	Description	Reset Value
CCAPx_PAR	CCAPx_BA+0x04	R/W	Camera Capture Interface Parameter Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved					FBB	Reserved	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			COLORCTL		VSP	HSP	PCLKP
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	RANGE	OUTFMT		INDATORD		SENTYPE	INFMT

Bits	Description	
[31:19]	**Reserved**	Reserved.
[18]	**FBB**	**Field by Blank** Field by Blank means blanking pixel data(0x80108010) have to transfer to system memory or not (**Only in Ccir-656** mode) 0 = Field by blank Disabled (blank pixel data will transfer to system memory). 1 = Field by blank Enabled (only active data will transfer to system memory).
[12:11]	**COLORCTL**	**Special COLORCTL Processing** 00 = Normal Color. 01 = Sepia effect, corresponding U,V component value is set at register CAP_SEPIA. 10 = Negative picture. 11 = Posterize image, the Y, U, V components posterizing factor are set at register CAP_POSTERIZE.
[10]	**VSP**	**Sensor Vsync Polarity** 0 = Sync Low. 1 = Sync High.
[9]	**HSP**	**Sensor Hsync Polarity** 0 = Sync Low. 1 = Sync High.
[8]	**PCLKP**	**Sensor Pixel Clock Polarity** 0 = Input video data and signals are latched by falling edge of Pixel Clock. 1 = Input video data and signals are latched by rising edge of Pixel Clock.
[7]	**PLNFMT**	**Planar Output YUV Format** 0 = YUV422. 1 = YUV420.
[6]	**RANGE**	**Scale Input YUV CCIR601 Color Range to Full Range** 0 = Default. 1 = Scale to full range.

[5:4]	**OUTFMT**	**Image Data Format Output to System Memory** 00 = YCbCr422. 01 = Only output Y. (Select this format when CCAPx_CTL “Luma_Y_One” or “MONO” enabled) 10 = RGB555. 11 = RGB565.
[3:2]	**INDATORD**	**Sensor Input Data Order** If INFMT (CAP_PAR [0]) = 0 (YCbCr). 00 = Sensor input data (Byte 0 1 2 3) is Y0 U0 Y1 V0. 01 = Sensor input data (Byte 0 1 2 3) is Y0 V0 Y1 U0. 10 = Sensor input data (Byte 0 1 2 3) is U0 Y0 V0 Y1. 11 = Sensor input data (Byte 0 1 2 3) is V0 Y0 U0 Y1. If INFMT (CAP_PAR [0]) = 1 (RGB565). 00 = Sensor input data (Byte 0) is {R[4:0],G[5:3]},. Sensor input data (Byte 1) is {G[2:0],B[4:0]}. 01 = Sensor input data (Byte 0) is {B[4:0],G[5:3]},. Sensor input data (Byte 1) is {G[2:0], R[4:0]}. 10 = Sensor input data (Byte 0) is {G[2:0],B[4:0]},. Sensor input data (Byte 1) is {R[4:0], G[5:3]}. 11 = Sensor input data (Byte 0) is {G[2:0],R[4:0]},. Sensor input data (Byte 1) is {B[4:0], G[5:3]}.
[1]	**SENTYPE**	**Sensor Input Type** 0 = CCIR601. 1 = CCIR656, Vsync and Hsync embedded in the data signal.
[0]	**INFMT**	**Sensor Input Data Format** 0 = YCbCr422. 1 = RGB565.

Camera Capture Interface Interrupt Register (CCAPx_INT)

Register	Offset	R/W	Description	Reset Value
CCAPx_INT	CCAPx_BA+0x08	R/W	Camera Capture Interface Interrupt Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				ADDRMIEN	Reserved	MEIEN	VIEN
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				ADDRMINTF	Reserved	MEINTF	VINTF

Bits	Description	
[31:20]	**Reserved**	Reserved.
[19]	**ADDRMIEN**	**Address Match Interrupt Enable Bit** 0 = Address match interrupt Disabled. 1 = Address match interrupt Enabled.
[18]	**Reserved**	Reserved.
[17]	**MEIEN**	**Bus Master Transfer Error Interrupt Enable Bit** 0 = Bus Master Transfer error interrupt Disabled. 1 = Bus Master Transfer error interrupt Enabled.
[16]	**VIEN**	**Video Frame End Interrupt Enable Bit** 0 = Video frame end interrupt Disabled. 1 = Video frame end interrupt Enabled.
[15:4]	**Reserved**	Reserved.
[3]	**ADDRMINTF**	**Memory Address Match Interrupt** 0 = Memory Address Match Interrupt did not occur. 1 = Memory Address Match Interrupt occurred. **Note:** This bit is cleared by writing 1 to it.
[2]	**Reserved**	Reserved.
[1]	**MEINTF**	**Bus Master Transfer Error Interrupt** 0 = Transfer Error did not occur. 1 = Transfer Error occurred. **Note:** This bit is cleared by writing 1 to it.
[0]	**VINTF**	**Video Frame End Interrupt** 0 = Did not receive a frame completely. 1 = Received a frame completely. **Note:** This bit is cleared by writing 1 to it.

YUV Component Posterizing Factor Register (CCAPx_POSTERIZE)

Register	Offset	R/W	Description	Reset Value
CCAPx_POSTERIZE	CCAPx_BA+0x0C	R/W	YUV Component Posterizing Factor Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
YCOMP							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
UCOMP							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VCOMP							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**YCOMP**	**Y Component Posterizing Factor** Specify the 8-bit value of YCOMP for Y_Posterizing_Factor[7:0]. Final_Y_Out[7:0] = Original_Y[7:0] & Y_Posterizing_Factor[7:0].
[15:8]	**UCOMP**	**U Component Posterizing Factor** Specify the 8-bit value of UCOMP for U_Posterizing_Factor[7:0]. Final_U_Out[7:0] = Original_U[7:0] & U_Posterizing_Factor[7:0].
[7:0]	**VCOMP**	**V Component Posterizing Factor** Specify the 8-bit value of VCOMP for V_Posterizing_Factor[7:0]. Final_V_Out[7:0] = Original_V[7:0] & V_Posterizing_Factor[7:0].

Sepia Effect Control Register (CCAPx_SEPIA)

Register	Offset	R/W	Description	Reset Value
CCAPx_SEPIA	CCAPx_BA+0x1C	R/W	Sepia Effect Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
UCOMP							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VCOMP							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:8]	**UCOMP**	**U Component Value for Sepia Color Effect** Define the constant U component while the “Sepia” color effect is turned on.
[7:0]	**VCOMP**	**V Component Value for Sepia Color Effect** Define the constant V component while the “Sepia” color effect is turned on.

Cropping Window Starting Address Register (CCAPx_CWSP)

Register	Offset	R/W	Description	Reset Value
CCAPx_CWSP	CCAPx_BA+0x20	R/W	Cropping Window Starting Address Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					CWSADDRV		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CWSADDRV							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				CWSADDRH			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CWSADDRH							

Bits	Description	
[31:27]	**Reserved**	Reserved.
[26:16]	**CWSADDRV**	**Cropping Window Vertical Starting Address** Specify the value of the cropping window vertical start address.
[15:12]	**Reserved**	Reserved.
[11:0]	**CWSADDRH**	**Cropping Window Horizontal Starting Address** Specify the value of the cropping window horizontal start address.

Cropping Window Size Register (CCAPx_CWS)

Register	Offset	R/W	Description	Reset Value
CCAPx_CWS	CCAPx_BA+0x24	R/W	Cropping Window Size Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					CWH		
23	22	21	20	19	18	17	16
CWH							
15	14	13	12	11	10	9	8
Reserved				CWW			
7	6	5	4	3	2	1	0
CWW							

Bits	Description	
[31:27]	**Reserved**	Reserved.
[26:16]	**CWH**	**Cropping Window Height** Specify the size of the cropping window height.
[15:12]	**Reserved**	Reserved.
[11:0]	**CWW**	**Cropping Window Width** Specify the size of the cropping window width.

Packet Scaling Vertical/Horizontal Factor Register (LSB) (CCAPx_PKTSL)

Register	Offset	R/W	Description	Reset Value
CCAPx_PKTSL	CCAPx_BA+0x28	R/W	Packet Scaling Vertical/Horizontal Factor Register (LSB)	0x0000_0000

31	30	29	28	27	26	25	24
PKTSVNL							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PKTSVML							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PKTSHNL							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PKTSHML							

Bits	Description	
[31:24]	**PKTSVNL**	**Packet Scaling Vertical Factor n** Specify the lower 8-bit of numerator part (N) of the vertical scaling factor. The lower 8-bit will be cascaded with higher 8-bit (PKDSVNH) to form a 16-bit numerator of vertical factor.
[23:16]	**PKTSVML**	**Packet Scaling Vertical Factor m** Specify the lower 8-bit of denominator part (M) of the vertical scaling factor. The lower 8-bit will be cascaded with higher 8-bit (PKDSVMH) to form a 16-bit denominator (M) of vertical factor. The output image width will be equal to the image height * N/M. **Note:** The value of N must be equal to or less than M.
[15:8]	**PKTSHNL**	**Packet Scaling Horizontal Factor n** Specify the lower 8-bit of numerator part (N) of the horizontal scaling factor. The lower 8-bit will be cascaded with higher 8-bit (PKDSHNH) to form a 16-bit numerator of horizontal factor.
[7:0]	**PKTSHML**	**Packet Scaling Horizontal Factor M** Specifies the lower 8-bit of denominator part (M) of the horizontal scaling factor. The lower 8-bit will be cascaded with higher 8-bit (PKDSHMH) to form a 16-bit denominator (M) of vertical factor. The output image width will be equal to the image width * N/M. **Note:** The value of N must be equal to or less than M.

Planar Scaling Vertical/Horizontal Factor Register (LSB) (CCAPx_PLNSL)

Register	Offset	R/W	Description	Reset Value
CCAPx_PLNSL	CCAPx_BA+0x2C	R/W	Planar Scaling Vertical/Horizontal Factor Register (LSB)	0x0000_0000

31	30	29	28	27	26	25	24
PLNSVNL							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PLNSVML							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PLNSHNL							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PLNSHNL							

Bits	Description	
[31:24]	**PLNSVNL**	**Planar Scaling Vertical Factor n** Specify the lower 8-bit of numerator part (N) of the vertical scaling factor. The lower 8-bit will be cascaded with higher 8-bit (PNDSVNH) to form a 16-bit numerator of vertical factor.
[23:16]	**PLNSVML**	**Planar Scaling Vertical Factor M** Specify the lower 8-bit of denominator part (M) of the vertical scaling factor. The lower 8-bit will be cascaded with higher 8-bit (PNDSVMH) to form a 16-bit denominator (M) of vertical factor. The output image width will be equal to the image height * N/M. **Note:** The value of N must be equal to or less than M.
[15:8]	**PLNSHNL**	**Planar Scaling Horizontal Factor n** Specify the lower 8-bit of numerator part (N) of the horizontal scaling factor. The lower 8-bit will be cascaded with higher 8-bit (PNDSHNH) to form a 16-bit numerator of horizontal factor.
[7:0]	**PLNSHML**	**Planar Scaling Horizontal Factor M** Specify the lower 8-bit of denominator part (M) of the horizontal scaling factor. The lower 8-bit will be cascaded with higher 8-bit (PNDSHMH) to form a 16-bit denominator (M) of vertical factor. The output image width will be equal to the image width * N/M. **Note:** The value of N must be equal to or less than M.

Scaling Frame Rate Factor Register (CCAPx_FRCTL)

Register	Offset	R/W	Description	Reset Value
CCAPx_FRCTL	CCAPx_BA+0x30	R/W	Scaling Frame Rate Factor Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	28	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		FRN					
7	6	5	4	3	2	1	0
Reserved		FRM					

Bits	Description	
[31:14]	**Reserved**	Reserved.
[13:8]	**FRN**	**Scaling Frame Rate Factor n** Specify the denominator part (N) of the frame rate scaling factor.
[7:6]	**Reserved**	Reserved.
[5:0]	**FRM**	**Scaling Frame Rate Factor M** Specify the denominator part (M) of the frame rate scaling factor. The output image frame rate will be equal to input image frame rate * (N/M). **Note:** The value of N must be equal to or less than M.

Output Frame Pixel Stride Width (CCAPx_STRIDE)

Register	Offset	R/W	Description	Reset Value
CCAPx_STRIDE	CCAPx_BA+0x34	R/W	Frame Output Pixel Stride Width Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	28	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		PKTSTRIDE					
7	6	5	4	3	2	1	0
PKTSTRIDE							

Bits	Description	
[31:14]	**Reserved**	Reserved.
[13:0]	**PKTSTRIDE**	**Packet Frame Output Pixel Stride Width** The output pixel stride size of the packet pipe. It is a 32-pixel aligned stride width for the Luma-Y-One bit format or a 4-pixel aligned stride with for the Luma-Y-Eight bit format when color or monochrome CMOS sensors used. This means that every new captured line address is word alignment when color or monochrome CMOS sensors used.

FIFO Threshold Register (CCAPx_FIFOTH)

Register	Offset	R/W	Description	Reset Value
CCAPx_FIFOTH	CCAPx_BA+0x3C	R/W	FIFO Threshold Register	0x070D_0507

31	30	29	28	27	26	25	24
OVF	Reserved		PKTFTH				
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			PLNYFTH				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				PLNUFTH			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				PLNVFTH			

Bits	Description	
[31]	**OVF**	**FIFO Overflow Flag** Indicate the FIFO overflow flag.
[30:29]	**Reserved**	Reserved.
[28:24]	**PKTFTH**	**Packet FIFO Threshold** Specify the 5-bit value of the packet FIFO threshold.
[23:21]	**Reserved**	Reserved.
[20:16]	**PLNYFTH**	**Planar Y FIFO Threshold** Specify the 5-bit value of the planar Y FIFO threshold.
[15:12]	**Reserved**	Reserved.
[11:8]	**PLNUFTH**	**Planar U FIFO Threshold** Specify the 4-bit value of the planar U FIFO threshold.
[7:4]	**Reserved**	Reserved.
[3:0]	**PLNVFTH**	**Planar V FIFO Threshold** Specify the 4-bit value of the planar V FIFO threshold.

Compare Memory Address Register (CCAPx_CMPADDR)

Register	Offset	R/W	Description	Reset Value
CCAPx_CMPADDR	CCAPx_BA+0x40	R/W	Compare Memory Base Address Register	0xFFFF_FFFC

31	30	29	28	27	26	25	24
CMPADDR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CMPADDR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CMPADDR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CMPADDR							

Bits	Description	
[31:0]	**CMPADDR**	**Compare Memory Base Address** It is a word alignment address, that is, the address is aligned by ignoring the 2 LSB bits [1:0].

Luminance Y8 to Y1 Threshold Value Register (CCAPx_LUMA_Y1_THD)

Register	Offset	R/W	Description	Reset Value
CCAPx_LUMA_Y1_THD	CCAPx_BA+0x44	R/W	Luminance Y8 to Y1 Threshold Value Register	0x0000_0080

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
LUMAY1THRESH							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**LUMAY1THRESH**	**Luminance Y8 to Y1 Threshold Value** Specify the 8-bit threshold value for the luminance Y bit-8 to the luminance Y 1-bit conversion.

Packet Scaling Vertical/Horizontal Factor Register (MSB) (CCAPx_PKTSM)

Register	Offset	R/W	Description	Reset Value
CCAPx_PKTSM	CCAPx_BA+0x48	R/W	Packet Scaling Vertical/Horizontal Factor Register (MSB)	0x0000_0000

31	30	29	28	27	26	25	24
PKTSVNH							
23	22	21	20	19	18	17	16
PKTSVMH							
15	14	13	12	11	10	9	8
PKTSHNH							
7	6	5	4	3	2	1	0
PKTSHMH							

Bits	Description	
[31:24]	**PKTSVNH**	**Packet Scaling Vertical Factor n** Specify the higher 8-bit of numerator part (N) of the vertical scaling factor. Please refer to the register "CCAPx_PKTSL" to check the cooperation between these two registers.
[23:16]	**PKTSVMH**	**Packet Scaling Vertical Factor m** Specify the lower 8-bit of denominator part (M) of the vertical scaling factor. Please refer to the register "CCAPx_PKTSL" to check the cooperation between these two registers.
[15:8]	**PKTSHNH**	**Packet Scaling Horizontal Factor n** Specify the lower 8-bit of numerator part (N) of the horizontal scaling factor. Please refer to the register "CCAPx_PKTSL" for the detailed operation.
[7:0]	**PKTSHMH**	**Packet Scaling Horizontal Factor M** Specify the lower 8-bit of denominator part (M) of the horizontal scaling factor. Please refer to the register "CCAPx_PKTSL" for the detailed operation.

Planar Scaling Vertical/Horizontal Factor Register (MSB) (CCAPx_PLNSM)

Register	Offset	R/W	Description	Reset Value
CCAPx_PLNSM	CCAPx_BA+0x4C	R/W	Planar Scaling Vertical/Horizontal Factor Register (MSB)	0x0000_0000

31	30	29	28	27	26	25	24
PLNSVNH							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PLNSVMH							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PLNSHNH							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PLNSHMH							

Bits	Description	
[31:24]	**PLNSVNH**	**Planar Scaling Vertical Factor n** Specifies the higher 8-bit of numerator part (N) of the vertical scaling factor. For detailed programming, please refer to the register "CAP_PLNSL".
[23:16]	**PLNSVMH**	**Planar Scaling Vertical Factor m** Specifies the lower 8-bit of denominator part (M) of the vertical scaling factor. For detailed programming, please refer to the register "CAP_PLNSL".
[15:8]	**PLNSHNH**	**Planar Scaling Horizontal Factor n** Specifies the higher 8-bit of numerator part (N) of the horizontal scaling factor. For detailed programming, please refer to the register "CAP_PLNSL".
[7:0]	**PLNSHMH**	**Planar Scaling Horizontal Factor m** Specifies the higher 8-bit of denominator part (M) of the horizontal scaling factor For detailed programming, please refer to the register "CAP_PLNSL".

Current Packet System Memory Address Register (CCAPx_CURADDRP)

Register	Offset	R/W	Description	Reset Value
CCAPx_CURADDRP	CCAPx_BA+0x50	R	Current Packet System Memory Address Register	0x0000_0000

31	30	29	28	27	26	25	24
CURADDR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CURADDR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CURADDR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CURADDR							

Bits	Description	
[31:0]	**CURADDR**	**Current Packet Output Memory Address** Specify the 32-bit value of the current packet output memory address.

Current Planar Y System Memory Address Register (CCAPx_CURADDRY)

Register	Offset	R/W	Description	Reset Value
CCAPx_CURADDRY	CCAPx_BA+0x54	R	Current Planar Y System Memory Address Register	0x0000_0000

31	30	29	28	27	26	25	24
CURADDR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CURADDR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CURADDR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CURADDR							

Bits	Description	
[31:0]	**CURADDR**	**Current Planar Y Output Memory Address** Specify the 32-bit value of the current planar Y output memory address.

Current Planar U System Memory Address Register (CCAPx_CURADDRU)

Register	Offset	R/W	Description	Reset Value
CCAPx_CURADDRU	CCAPx_BA+0x58	R	Current Planar U System Memory Address Register	0x0000_0000

31	30	29	28	27	26	25	24
CURADDR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CURADDR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CURADDR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CURADDR							

Bits	Description	
[31:0]	**CURADDR**	**Current Planar U Output Memory Address** Specify the 32-bit value of the current planar U output memory address.

Current Planar V System Memory Address Register (CCAPx_CURADDRV)

Register	Offset	R/W	Description	Reset Value
CCAPx_CURADDRV	CCAPx_BA+0x5C	R	Current Planar V System Memory Address Register	0x0000_0000

31	30	29	28	27	26	25	24
CURADDR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CURADDR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CURADDR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CURADDR							

Bits	Description	
[31:0]	**CURADDR**	**Current Planar V Output Memory Address** Specify the 32-bit value of the current planar V output memory address.

System Memory Packet Base Address 0 Register (CCAPx_PKTBA0)

Register	Offset	R/W	Description	Reset Value
CCAPx_PKTBA0	CCAPx_BA+0x60	R/W	System Memory Packet Base Address 0 Register	0x0000_0000

31	30	29	28	27	26	25	24
BASEADDR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
BASEADDR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
BASEADDR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BASEADDR							

Bits	Description	
[31:0]	**BASEADDR**	**System Memory Packet Base Address 0** It is a word alignment address, that is, the address is aligned by ignoring the 2 LSB bits [1:0].

System Memory Planar Y Base Address Register (CCAPx_YBA)

Register	Offset	R/W	Description	Reset Value
CCAPx_YBA	CCAPx_BA+0x80	R/W	System Memory Planar Y Base Address Register	0x0000_0000

31	30	29	28	27	26	25	24
BASEADDR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
BASEADDR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
BASEADDR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BASEADDR							

Bits	Description	
[31:0]	**BASEADDR**	**System Memory Planar Y Base Address** It is a word alignment address, that is, the address is aligned by ignoring the 2 LSB bits [1:0].

System Memory Planar U Base Address Register (CCAPx_UBA)

Register	Offset	R/W	Description	Reset Value
CCAPx_UBA	CCAPx_BA+0x84	R/W	System Memory Planar U Base Address Register	0x0000_0000

31	30	29	28	27	26	25	24
BASEADDR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
BASEADDR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
BASEADDR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BASEADDR							

Bits	Description	
[31:0]	**BASEADDR**	**System Memory Planar U Base Address** It is a word alignment address, that is, the address is aligned by ignoring the 2 LSB bits [1:0].

System Memory Planar V Base Address Register (CCAPx_VBA)

Register	Offset	R/W	Description	Reset Value
CCAPx_VBA	CCAPx_BA+0x88	R/W	System Memory Planar V Base Address Register	0x0000_0000

31	30	29	28	27	26	25	24
BASEADDR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
BASEADDR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
BASEADDR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BASEADDR							

Bits	Description	
[31:0]	**BASEADDR**	**System Memory Planar V Base Address** It is a word alignment address, that is, the address is aligned by ignoring the 2 LSB bits [1:0].

6.40 LCD Display Controller (DISP)

6.40.1 Overview

The Vivante DCUltraLite (DCUltraL) display controller (DISP) IP defines a high-performance optimized-area display core that can be used for reading rendered images from the frame buffer to the display. In addition to providing hardware cursor patterns, the display controller performs format conversions, dithering and gamma corrections. This controller includes support for parallel pixel output and is easily adapted to external serialization logic, for example HDMI. This document contains copyright confidential and proprietary material disclosed with permission of Vivante Corporation. Vivante has authorized re-distribution of this material restricted to those NDA partners and licensees of Nuvoton who are engineering MPU product which includes the discussed Vivante IP. All rights reserved.

6.40.2 Features

- Video Timing Generation
  - HSYNC, VSYNC, DE signals
  - Programmable timers
- Sync Interface
  - Parallel Pixel Output with 24-bit Data, HSync, VSync, Data enable
  - DPI 24-bit, 18-bit (2 configs) and 16-bit support (3 configs)
  - Easily adaptable to external serialization logic, e.g. HDMI
- MPU Interface
  - Support i80/m68 with 8, 9, 16, 18-bit system interface
  - Support VSYNC interface (output)
  - Support TE signal (input)
  - Support LCD chip register configuration by MPU interface
- Display
  - Single display
  - Maximum display size: 1080p
  - Sync and blank signals
  - Gamma and dither tables
- Input Formats
  - ARGB2101010,A/XRGB8888,A/XRGB1555,RGB565,A/XRGB4444
  - Index1/2/4/8
  - YUV422 packed and semi planar (YUY2, UYVY, NV16)
  - YUV420 semi-planar (YUY2(P010), NV12 and YUV420 semi-planar 10-bit)
- Format Conversion
  - Pixel inputs accepted from multiple RGB and YUV formats
  - Color Space Conversion BT.2020 and BT.709
  - Pixel output is 24 bit RGB in multiple formats
- Output Formats
  - DPI_D16CFG1/DPI_D16CFG2/DPI_D16CFG3/DPI_D18CFG1/DPI_D18CFG2/

DPI_D24

- Hardware Cursor
  - Supports ARGB888 and Mask cursor formats
- Supports OSD function
- Color
  - A separate Look Up Table for Dither
  - A separate Look Up Table for Gamma Correction
  - Overlay with coordinate generator
  - Alpha Blending: 8 Porter Duff Blending modes

6.40.3 Block Diagram

The main functional blocks of the DCUltraL are described here. A block diagram of the complete graphics pipeline is given in the figure below.

Host Interface

Allows the core to communicate with external memory and the CPU through the AXI and AHB buses. In this block, data crosses clock domain boundaries.

Front End and Pixel Pipeline

Single Display pipeline supports linear frame buffers for RGB and YUV inputs.

Cursor

Provides hardware cursor functionality.

Dither

Provides an indexable look-up-table (LUT).

Gamma Correction

Provides a gamma correction SRAM.

Display Interface

Supports Parallel Pixel Output with 24-bit Data, HSync, VSync, Data enable. Easily adapted to external serialization logic, e.g. HDMI.

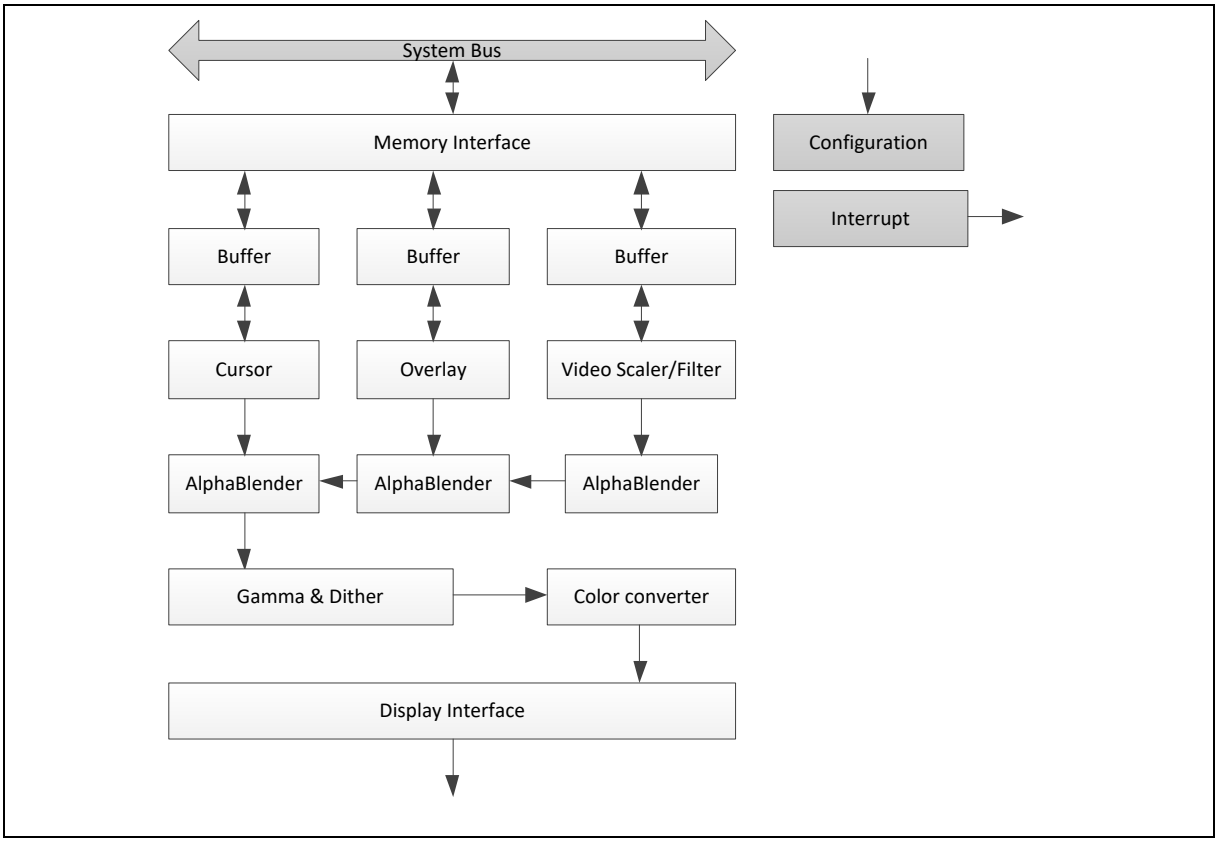


Figure 6.40-1 Display Controller Block Diagram

6.40.4 Basic Configuration

- Clock Source Configuration
  - Select DISP engine clock source in DCUSEL (CLK_CLKSEL0[24])
  - Enable DISP engine clock in DCUEN (CLK_SYSCLK0[26])
- DDR AXI Configuration
  - Enable DDR AXI port2 in port_en (PCTRL_2[0])
- Reset Configuration
  - Reset DISP engine in DISPCRST (SYS_IPRST0[9])
- Pin Configuration

Group	Pin Name	GPIO	MFP
LCD Display	LCM_VSYNC	PG.8	MFP6
	LCM_HSYNC	PG.9	MFP6
	LCM_CLK	PG.10	MFP6
	LCM_DEN	PK.4	MFP6
	LCM_DATA0	PI.8	MFP6
	LCM_DATA1	PI.9	MFP6
	LCM_DATA2	PI.10	MFP6

	LCM_DATA3	PI.11	MFP6
	LCM_DATA4	PI.12	MFP6
	LCM_DATA5	PI.13	MFP6
	LCM_DATA6	PI.14	MFP6
	LCM_DATA7	PI.15	MFP6
	LCM_DATA8	PH.0	MFP6
	LCM_DATA9	PH.1	MFP6
	LCM_DATA10	PH.2	MFP6
	LCM_DATA11	PH.3	MFP6
	LCM_DATA12	PH.4	MFP6
	LCM_DATA13	PH.5	MFP6
	LCM_DATA14	PH.6	MFP6
	LCM_DATA15	PH.7	MFP6
	LCM_DATA16	PC.12	MFP6
	LCM_DATA17	PC.13	MFP6
	LCM_DATA18	PC.14	MFP6
	LCM_DATA19	PC.15	MFP6
	LCM_DATA20	PH.12	MFP6
	LCM_DATA21	PH.13	MFP6
	LCM_DATA22	PH.14	MFP6
	LCM_DATA23	PH.15	MFP6
MPU Display	LCM_MPU_RD/EN	PG.8	MFP6
	LCM_MPU_WR/RW	PG.9	MFP6
	LCM_MPU_RS	PK.4	MFP6
	LCM_MPU_D0	PI.8	MFP6
	LCM_MPU_D1	PI.9	MFP6
	LCM_MPU_D2	PI.10	MFP6
	LCM_MPU_D3	PI.11	MFP6
	LCM_MPU_D4	PI.12	MFP6
	LCM_MPU_D5	PI.13	MFP6
	LCM_MPU_D6	PI.14	MFP6
	LCM_MPU_D7	PI.15	MFP6
	LCM_MPU_D8	PH.0	MFP6
	LCM_MPU_D9	PH.1	MFP6
	LCM_MPU_D10	PH.2	MFP6

	LCM_MPU_D11	PH.3	MFP6
	LCM_MPU_D12	PH.4	MFP6
	LCM_MPU_D13	PH.5	MFP6
	LCM_MPU_D14	PH.6	MFP6
	LCM_MPU_D15	PH.7	MFP6
	LCM_MPU_D16	PC.12	MFP6
	LCM_MPU_D17	PC.13	MFP6
	LCM_MPU_CS	PC.14	MFP6
	LCM_MPU_TE	PC.15	MFP7
	LCM_MPU_VSYNC	PC.15	MFP8

6.40.5 Functional Description

6.40.5.1 Video Timing Generation

Video timing generation (VTG) includes a timer corresponding to the pixel clock. Through the timer, VTG generates digital RGB output, including HSYNC, VSYNC, and DE (DataEnable which shows the “Active display”), as illustrated in the figure below.

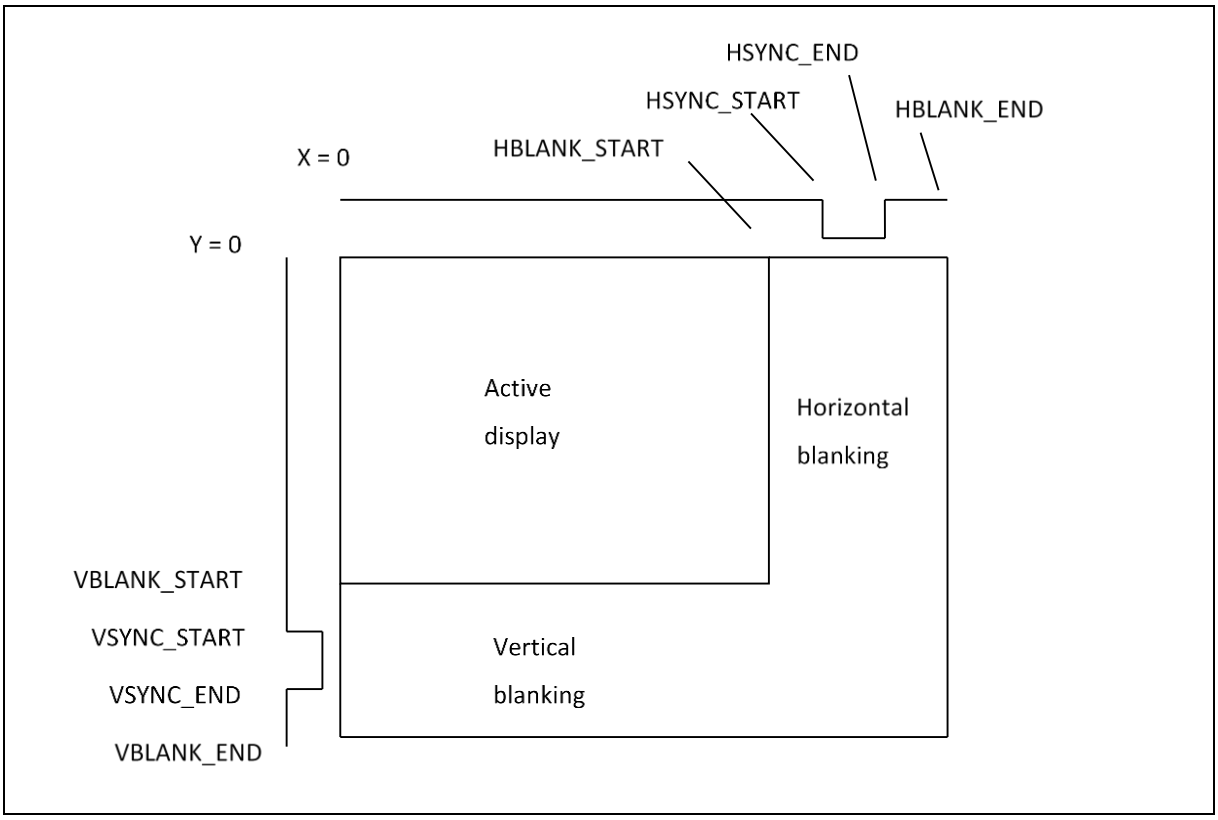


Figure 6.40-2 Timing Diagram

6.40.5.2 Hardware Cursor

Hardware provides a 32x32 pixel cursor

Cursor Format

The Vivante Display Controller supports a 32x32 pixel cursor. Two modes are supported: A8R8G8B8 and MASKED.

- A8R8G8B8 mode provides a full color cursor and one cursor pixel occupies 4 Bytes.

```
{redCursor,greenCursor,BlueCursor} = {red, green, blue} // read from memory,
alphacursor = alpha[7:3]
```

- MASKED cursor mode uses 2 bits for one cursor pixel. Thus 32x32 cursor pixels have 32 cursor color rows and each cursor color row is 64 bits.

```
cursorColorRow_H[31:0] = colorRow[63:32]
cursorColorRow_L[31:0] = colorRow[31:0]
xorCursor = cursorColorRow_H[cursorXPos[4:0]]
andCursor = cursorColorRow_L[cursorXPos[4:0]]
//  andCursor   xorCursor      cursor color
//  0           0          background register color
//  0           1          foreground register color
//  1           0          frame buffer pixel color
//  1           1          invert frame buffer pixel color
```

Note that if the cursor format is masked, cursor alpha = 5'b1_1111. A8R8G8B8 cursor data is read from memory. For the mask, read the 2-bit masked cursor data to get the cursor value based on these 2-bit combinations. Once the cursor pixel value is determined, the cursor is blended with the frame buffer pixel value.

Cursor Points

The illustrations below show two points important for tracking a cursor's position:

- Top-left point

  which is used as the base address for the cursor.

- Hot spot

  which is used in a search for a corresponding screen coordinate when a selection is made, such as when the "Enter" key or left mouse button is pushed.

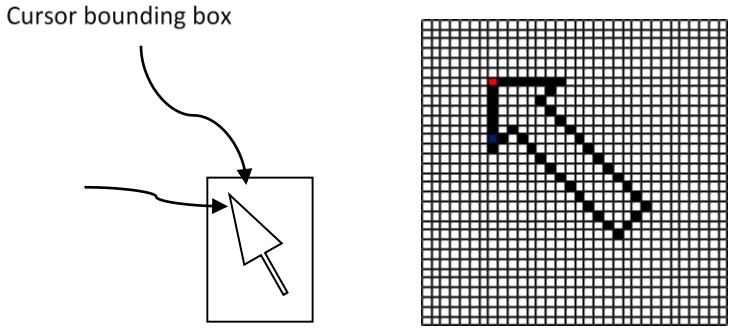


Figure 6.40-3 Cursor

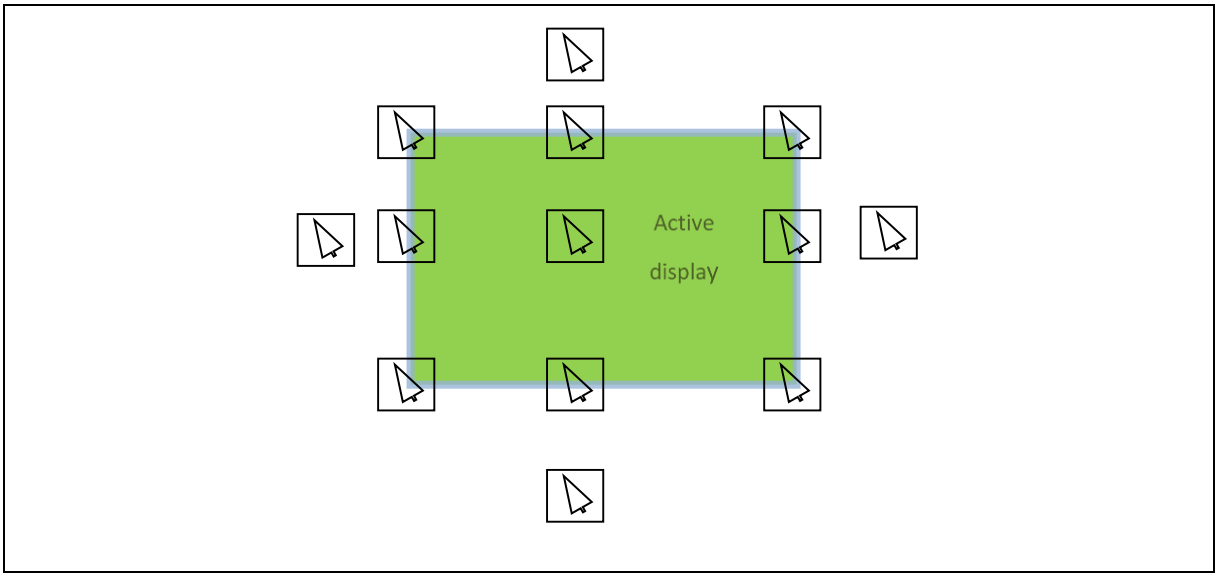


Figure 6.40-4 Examples of Various Cursor Position in Relation to a Display Window

6.40.5.3 Swizzle

Swizzle is used to switch the position of components in pixel data. The same color format may be stored in different order in memory. To accommodate this possibility, the Vivante Display Controller will first swizzle an input pixel. Since the input pixel may have been input with four different format orders: ARGB, RGBA, ABGR or BGRA, this initial swizzle is just to convert the format into ARGB. For YUV formats, the U and V pixels may be placed in different positions, so UV-swizzle is used to convert them to a common format.

Note that the Vivante Display Controller assumes the initial input of ARGB data is in ARGB order, which means A is in the upper bits while B is in the lowest bits. If the user input is not in ARGB order, swizzle is needed to reshape the data, e.g., if the input format is in BGRA order, the swizzle should be configured to handle BGRA format.

For YUV formats, especially for UV mixed formats, the Display Controller assumes that for each UV pair, U is in the lower bits while V is in the upper bits. If the user input is not in the pre-mentioned order, UV-swizzle should be configured to reshape the data.

Bit No.	31							24	23							16	15							8	7							0
32-bit Formats	**Pixel = 31:0**																															
A8R8G8B8	A	A	A	A	A	A	A	A	R	R	R	R	R	R	R	R	G	G	G	G	G	G	G	G	B	B	B	B	B	B	B	B
X8R8G8B8	X	X	X	X	X	X	X	X	R	R	R	R	R	R	R	R	G	G	G	G	G	G	G	G	B	B	B	B	B	B	B	B
A2B10G10R10	A	A	B	B	B	B	B	B	B	B	B	B	G	G	G	G	G	G	G	G	G	G	R	R	R	R	B	R	R	R	R	R
16-bit Formats	**Second Pixel = 31:16**																**First Pixel = 15:0**															
A1R5G5B5	A	R	R	R	R	R	G	G	G	G	G	B	B	B	B	B	A	R	R	R	R	R	G	G	G	G	G	B	B	B	B	B
X1R5G5B5	X	R	R	R	R	R	G	G	G	G	G	B	B	B	B	B	X	R	R	R	R	R	G	G	G	G	G	B	B	B	B	B
A4R4G4B4	A	A	A	A	R	R	R	R	G	G	G	G	B	B	B	B	A	A	A	A	R	R	R	R	G	G	G	G	B	B	B	B
X4R4G4B4	X	X	X	X	R	R	R	R	G	G	G	G	B	B	B	B	X	X	X	X	R	R	R	R	G	G	G	G	B	B	B	B

R5G6B5	R	R	R	R	R	G	G	G	G	G	G	B	B	B	B	B	R	R	R	R	R	G	G	G	G	G	G	B	B	B	B	B

Table 6.40-1 Expected Input Organization for RGB Data

6.40.5.4 ColorKey for Overlay

Vivante DCUltraL Display Controllers provide for overlay and transparency effects using Color Keys to blend pixels in images. A chosen color in a source image can be selected and filtered out so that it can be replaced with a different designated color in the destination target image.

The filter mechanism is controlled using the dcregOverlayColorKey/KeyHigh0~15 registers. Each pixel color in the overlay or video layer is compared against the range specified in these registers. If the pixel color is neither smaller than colorKey nor larger than colorKeyHigh, the Display Controller will regard it as a special color and replace it with another color. If the source color was on the overlay layer, it is replaced with the corresponding pixel on the video layer. If the source color was on the video layer, it will be replaced with the configured background color.

ColorKey Features and Constraints

- ColorKey is only supported for ARGB format. It is not available for Index or YUV formats.
- ColorKey is done before alpha blend.
  - If ColorKey is enabled for overlay, the special color pixel will be replaced with the corresponding video pixel color.
  - For the video layer the special color pixel will be replaced with the configured background color.
- The comparison between pixel color and color is done by comparing the component value of the pixel and the ColorKey value, e. g. R,G,B ,A of the color.
- All of the components must match the ColorKey condition if the pixel is to be regarded as special.
- The ColorKey and ColorKeyHigh values are both configured in the 32 bit registers with the format ARGB8888, which means A value is written from 24 to 31 bits, R value is written from 16 to 23 bits, G value is written from 8 to 15 bits and B value is written from 0 to 7 bits. The detailed description of the register value is shown with the figure below.

Bit No.	**31**								**24**	**23**							**16**	**15**							**8**	**7**							**0**
ColorKey Format	**Pixel = 31:0**																																
A8R8G8B8	A	A	A	A	A	A	A	A		R	R	R	R	R	R	R	R	G	G	G	G	G	G	G	G	B	B	B	B	B	B	B	B

Table 6.40-2 ColorKey Organization

6.40.5.5 Dither

In an image where color intensities change slowly, there may be noticeable jumps in intensity levels between pixels. Dithering can be used to diffuse the intensity across neighboring pixels. In dithering mode pixels are scanned in order, and errors in calculating a pixel's intensity are distributed (i.e., diffused) to neighboring pixels to keep the overall intensity of the image closer to the input intensity.

Dither Look Up Table

Vivante's display controller dither implementation requires a look up table of 4x4 entries of 4 bits. These 64 bit tables are located in four 32-bit registers dcregDisplayDitherTableLow0 and

dcregDisplayDitherTableHigh0. The dither look-up table needs to be indexed by the two least significant bits x[1:0] and the two least significant bits y[1:0] of the displays.

Enabling Dithering

Dithering can be enabled using the following method:

1. Decide which bit of pixel color to enhance.

   This can be done by setting the register Display Dither Configuration (dcregDisplayDitherConfig*, for display0 offset 0x1410). For example, setting RedSize=6 (values may vary from 4 to 8) means it is the 6th bit starting from the MSB that we want to enhance, in other words, it is the RedColor[2]bit from RedColor[7:0]. GreenSize and BlueSize function in the same way.

2. Create the look-up table.

   By setting register Display Dither Table (dcregDisplayDitherTableLow/High, for display0 offset 0x1420, high 0x1428).

   1) The Look-Up Table includes 16 entries, 4 bits for each.
   2) The Look-Up Table provides a value U[3:0] through the index X[1:0] and Y[1:0].
   3) The color value RedColor[3:0] is used to compare with this U[3:0].
   4) If RedColor[3:0] > U[3:0], and RedColor[7:2] is not 6'b111111, then the final color value is: NewRedColor = RedColor[7:2] + 1'b1.
   5) If RedColor[3:0] ≤ U[3:0], then NewRedColor = RedColor[7:2].

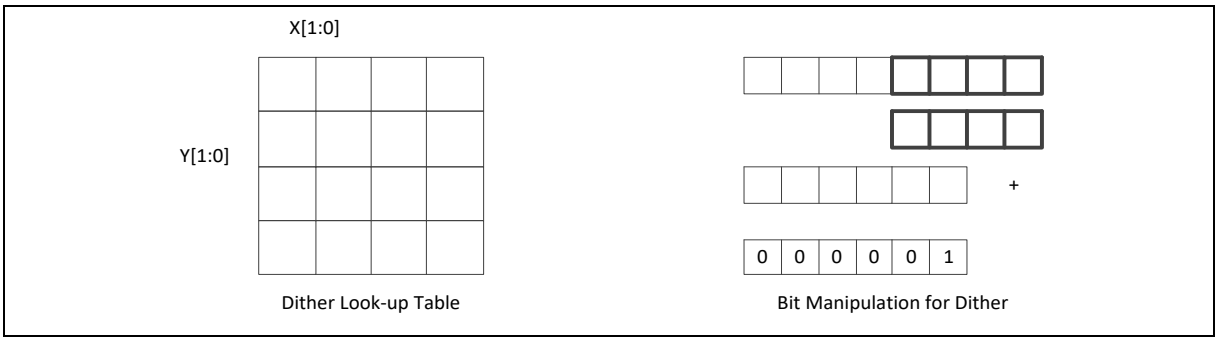


Figure 6.40-5 Dither Diagrams

6.40.5.6 Gamma Correction

Gamma Correction Look Up Table

The gamma correction look up table includes 3 look-up tables: red, green, and blue respectively.

The look-up tables are contained in the write-only register dcregGamma Data* (offset 0x1460 for display0).

Enabling Gamma Correction

Gamma correction could be achieved through the following example methods:

1. If you describe gamma correction as (originalColor, newColor), then you could set a gamma Look Up Table sequentially based on color value, starting from the smallest value progressing to the largest value.

2. If you want to do gamma correction using the following sequences: (0, 0), (1,2), (2, 5), (3, 6) ……, then you could set the register GammaIndex=0, which would mean gamma correction starts from "0." Then set register "GammaData" in this sequence: 0, 2, 5, 6……, for 256 times to fully fill the Look-Up Table.

6.40.5.7 Display Formats

Output Formats

Supported output formats are: DPI_D16CFG1, DPI_D16CFG2, DPI_D16CFG3, DPI_D18CFG1, DPI_D18CFG2, DPI_D24.

Input Formats

This Display Controller supports linear input formats as described in the tables below. Cursor format is RGB888.

Format	Linear Input
ARGB2101010	YES
ARGB8888/XRGB8888	YES
ARGB1555/XRGB1555	YES
RGB565	YES
ARGB4444/XRGB444	YES
YUV422-YUY2	YES
YUV422-UYVY	YES
YUV420-YV12	YES
YUV420-NV12	YES
YUV420-NV12_10-bit	YES
YUV420-P010_10-bit	YES
YUV422-NV16	YES
YUV420-NV16_10-bit	YES
Index1/2/4/8	YES

Table 6.40-3 Display Input Formats

Stride Alignment

Linear	Format	Alignment (In Pixels)	BaseAddress (Bytes)
Linear	planar YUV	128	128
	RGB /Index/ packed YUV	128	128

Table 6.40-4 Alignment and Base Addressing

6.40.5.8 Data Conversion

ARGB Data Conversion

The pixels read from source or destination will be expanded into ARGB formats to maintain lossless pixel operations. The resulting pixels will be converted into the destination format.

YUV to RGB Conversion

YUV data is converted using the 10-bit pipeline into 10-bit per component RGB format. Once converted, there is no way back to YUV format. The IP supports BT.2020 and user programmable coefficient YUV to RGB color conversion standards. The hardware can also support BT.709 for 8-bit.

6.40.5.9 Alpha Blending

On destination, the IP supports destination pre-multiply and destination alpha.

The general alpha blending equations are:

Cd = Fs * Cs' + Fd * Cd' **(1)**

Ad = Fs * As'' + Fd * Ad'' **(2)**

Where

Cs' is the source color component (adjusted for NPM if necessary)

Cd' is the destination color component (adjusted for NPM if necessary)

As'' is the modified source alpha component

Ad'' is the modified destination alpha component

Fs is fraction of the source that contributes to the final value

Fd is fraction of the destination that contributes to the final value

The blending is done in 5 logical stages (not real implementation stages):

1. Transparent/opaque conversion

   In this stage, the incoming alpha (source or destination independently) can be inverted if needed to match the internal alpha rule. Internally, an alpha of 0 means transparent, while an alpha of "0xFF" means opaque. External content might follow the opposite rule. The output of the block is either As (Ad for destination) or 1-As (1-Ad for destination).

2. Global value substitution

   A global alpha value from a register can be used to substitute the incoming alpha. An incoming alpha As can pass-through, be directly substituted by Ags (global alpha) or scaled by the global alpha value (As * Ags). The source and destination have distinct global alpha values.

3. Blending factor generation

   At this stages, the blending factors are generated (refer to table below). Each alpha can take the values 0, 1, A or 1-A depending on the blending mode.

4. Final blending

   This is the final stage which implements the operations described by equations (1) and (2),

The different stages are illustrated in the following figure.

Blending Mode	Fs	Fd
Clear	0	0
SRC	1	0
DST	0	1
SRC_OVER	1	1 – As''
DST_OVER	1 – Ad''	1
SRC_IN	Ad''	0
DST_IN	0	As''
SRC_OUT	1 – Ad''	0

Table 6.40-5 Blending Modes Fractions Description

6.40.5.10 DPI Interface Pin Configuration

The DPI interface is the standard interface for the DCUltra Series. For cores with MPU, some of these signals are used for MPU as well as DPI. The dcregDpiConfig0 registers are available for DPI format selection. Use DPI_DATA_FORMAT(dcregDpiConfig0 [2:0]) for data format. Default is 0 = D16CFG1. Other values are 1 = D16CFG2, 2 = D16CFG3, 3 = D18CFG1, 4 = D18CFG2, and 5 = D24. Refer to the table below for the bit mapping of these formats. Example mappings of Display 29~0 to display0Color are provided in the Display Bit Map table below.

	D23	D22	D21	D20	D19	D18	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
DPI 24 bit	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
DPI 18 bit Config 1							R5	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1	B0
DPI 18 bit Config 2			R5	R4	R3	R2	R1	R0			G5	G4	G3	G2	G1	G0			B5	B4	B3	B2	B1	B0
DPI 16 bit Config 1									R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
DPI 16 bit Config 2				R4	R3	R2	R1	R0			G5	G4	G3	G2	G1	G0				B4	B3	B2	B1	B0
DPI 16 bit Config 3			R4	R3	R2	R1	R0				G5	G4	G3	G2	G1	G0			B4	B3	B2	B1	B0	

Table 6.40-6 Display Bit Mapping Options with DPI

6.40.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
DISP Base Address: **DISP_BA = 0x4026_0000**				
AQHiClockControl	DISP_BA+0x0000	R/W	Clock Control Register	0x00000900
dcregFrameBufferAddress0	DISP_BA+0x1400	R/W	Framebuffer Start Address Register. Starting address of the framebuffer. **Note:** This register is double buffered.	0x00000000
dcregFrameBufferStride0	DISP_BA+0x1408	R/W	Framebuffer Stride Register. Stride of the framebuffer in bytes. **Note:** This register is double buffered.	0x00000000
dcregDisplayDitherConfig0	DISP_BA+0x1410	R/W	Display Dither Configuration Register. Configuration register for dithering. **Note:** This register is double buffered.	0x00000000
dcregPanelConfig0	DISP_BA+0x1418	R/W	Panel Configuration Register. **Note:** This register is double buffered.	0x00000000
dcregDisplayDitherTableLow0	DISP_BA+0x1420	R/W	Display Dither Table Register. **Note:** This register is double buffered.	0x00000000
dcregDisplayDitherTableHigh0	DISP_BA+0x1428	R/W	Display Dither Table Register. **Note:** This register is double buffered.	0x00000000
dcregHDisplay0	DISP_BA+0x1430	R/W	Horizontal Total and Display End Counter Register. **Note:** This register is double buffered	0x00000000
dcregHSync0	DISP_BA+0x1438	R/W	Horizontal Sync Counter Register. **Note:** This register is double buffered.	0x00000000
dcregVDisplay0	DISP_BA+0x1440	R/W	Vertical Total and Display End Counter Register. **Note:** This register is double buffered.	0x00000000
dcregVSync0	DISP_BA+0x1448	R/W	Vertical Sync Counter Register. **Note:** This register is double buffered.	0x00000000
dcregDisplayCurrentLocation0	DISP_BA+0x1450	R	Display Current Location Register. Current x,y location of display controller.	0x00000000
dcregGammaIndex0	DISP_BA+0x1458	W	Gamma Table Index Register. Index into gamma table. See dcregGammaData for more information.	0x00000000
dcregGammaData0	DISP_BA+0x1460	W	Gamma Data Translation Register. Translation values for the gamma table. When this register gets written, the data gets stored in the gamma table at the index specified by the dcregGammaIndex register. After the register is written, the index gets incremented.	0x00000000
dcregCursorConfig	DISP_BA+0x1468	R/W	Cursor Configuration Register.	0x00000000
dcregCursorAddress	DISP_BA+0x146C	R/W	Cursor Address Register. Address of the cursor shape. **Note:** This register is double buffered.	0x00000000

dcregCursorLocation	DISP_BA+0x1470	R/W	Cursor Location Register. Location of the cursor on the owning display. **Note:** This register is double buffered.	0x00000000
dcregCursorBackground	DISP_BA+0x1474	R/W	Masked Cursor Background Color Register. The background color for Masked cursors. **Note:** This register is double buffered.	0x00000000
dcregCursorForeground	DISP_BA+0x1478	R/W	Masked Cursor Foreground Color Register. The foreground color for Masked cursors. **Note:** This register is double buffered.	0x00000000
dcregDisplayIntr	DISP_BA+0x147C	R	Display Interrupt Register	0x00000000
dcregDisplayIntrEnable	DISP_BA+0x1480	R/W	Display Interrupt Enable Register. The interrupt enable register for display_0. **Note:** Interrupt enable for register dcregDisplayIntr.	0x00000000
dcregCursorModuleClockGatingControl	DISP_BA+0x1484	R/W	Clock Gating Control for Cursor Register. Module level clock gating control for cursor.	0x00000001
dcregGeneralConfig0	DISP_BA+0x14B0	R/W	General Miscellaneous Configuration Register. Misc option configuration. **Note:** This register is double buffered.	0x00000000
dcregDpiConfig0	DISP_BA+0x14B8	R/W	DPI Configuration Register. The configuration register for DPI output. **Note:** This register is double buffered.	0x00000000
dcregDebugCounterSelect0	DISP_BA+0x14D0	R/W	Debug Counter Selection Register.	0x00000000
dcregDebugCounterValue0	DISP_BA+0x14D8	R	Debug Counter Value Register	0x00000000
dcregFrameBufferColorKey0	DISP_BA+0x1508	R/W	Framebuffer Color Key Start Address Register. Start of color key range of framebuffer. **Note:** This register is double buffered.	0x00000000
dcregFrameBufferColorKeyHigh0	DISP_BA+0x1510	R/W	Framebuffer Color Key End Address Register. End of color key range of framebuffer. **Note:** This register is double buffered.	0x00000000
dcregFrameBufferConfig0	DISP_BA+0x1518	R/W	Framebuffer Configuration Register. Framebuffer attribute configuration. **Note:** This register is double buffered. Some fields are double buffered.	0x00000000
dcregFrameBufferBGColor0	DISP_BA+0x1528	R/W	Framebuffer Background Color Register. Background color used when a pixel from the framebuffer falls outside of the range of color key.	0x00000000
dcregFrameBufferUPlanarAddress0	DISP_BA+0x1530	R/W	Framebuffer Second Plane U Start Address Register. Starting address of the second planar (often the U plane) of the framebuffer if one exists.	0x00000000
dcregFrameBufferVPlanarAddress0	DISP_BA+0x1538	R/W	Framebuffer Third Plane V Start Address Register. Starting address of the third planar (often the V plane) of the framebuffer if one exists. **Note:** This register is double buffered.	0x00000000
dcregOverlayConfig0	DISP_BA+0x1540	R/W	Overlay Configuration Register. Overlay attributes control. **Note:** This register is double buffered.	0x00000000

dcregOverlayAlphaBlendConfig0	DISP_BA+0x1580	R/W	Overlay Alpha Blending Configuration Register. **Note:** This register is double buffered.	0x00000000
dcregOverlayAddress0	DISP_BA+0x15C0	R/W	Overlay Start Address Register. Starting address of the overlay. **Note:** This register is double buffered.	0x00000000
dcregOverlayStride0	DISP_BA+0x1600	R/W	Overlay Stride Register. Stride of the overlay in bytes. **Note:** This register is double buffered.	0x00000000
dcregOverlayTL0	DISP_BA+0x1640	R/W	Overlay Origin Register. Top left coordinate of the panel pixel where the overlay should start. Be aware there is no panning inside the overlay. **Note:** This register is double buffered	0x00000000
dcregOverlayBR0	DISP_BA+0x1680	R/W	Overlay End Register. Bottom right coordinate of the panel pixel where the overlay should end. The border is inclusive. **Note:** This register is double buffered.	0x00000000
dcregOverlaySrcGlobalColor0	DISP_BA+0x16C0	R/W	Overlay Source Global Color Register. Color value used when alpha blending process is configured to use global color for source. **Note:** This register is double buffered.	0x00000000
dcregOverlayDstGlobalColor0	DISP_BA+0x1700	R/W	Overlay Destination Global Color Register. Color value used when alpha blending process is configured to use global color for destination. **Note:** This register is double buffered.	0x00000000
dcregOverlayColorKey0	DISP_BA+0x1740	R/W	Overlay Color Key Start Address Register. Start of color key range for overlay. **Note:** This register is double buffered.	0x00000000
dcregOverlayColorKeyHigh0	DISP_BA+0x1780	R/W	Overlay Color Key End Address Register. End of color key range for overlay. **Note:** This register is double buffered.	0x00000000
dcregOverlaySize0	DISP_BA+0x17C0	R/W	Overlay Window Size Register. Window size of the overlay buffer in memory - in pixels. **Note:** This register is double buffered.	0x00000000
dcregFrameBufferUStride0	DISP_BA+0x1800	R/W	Framebuffer Second Plane U Stride Register. Stride of the second planar (often the U plane) of the framebuffer if one exists. **Note:** This register is double buffered.	0x00000000
dcregFrameBufferVStride0	DISP_BA+0x1808	R/W	Framebuffer Third Plane V Stride Register. Stride of the third planar (often the V plane) of the framebuffer if one exists. **Note:** This register is double buffered.	0x00000000
dcregFrameBufferSize0	DISP_BA+0x1810	R/W	Framebuffer Size Register. Window size of the framebuffer in memory - in pixels. **Note:** This register is double buffered.	0x00000000
dcregIndexColorTableIndex0	DISP_BA+0x1818	W	Index Color Table Index Register. Index into index color table. See dcregIndexColorTableData for more information.	0x00000000

dcregIndexColorTableData0	DISP_BA+0x1820	W	Index Color Table Data Translation Register. Translation values for the index color table. When this register gets written, the data gets stored in the index color table at the index specified by the dcregIndexColorTableIndex register. After the register is written, the index gets incremented. **Note:** This register is double buffered.	0x00000000
dcregOverlayUPlanarAddress0	DISP_BA+0x1840	R/W	Overlay Second Plane U Start Address Register. Address of the second planar (often U plane) of the overlay if one exists. **Note:** This register is double buffered.	0x00000000
dcregOverlayVPlanarAddress0	DISP_BA+0x1880	R/W	Overlay Third Plane V Start Address Register. Address of the third planar (often V plane) of the overlay if one exists. **Note:** This register is double buffered.	0x00000000
dcregOverlayUStride0	DISP_BA+0x18C0	R/W	Overlay Second Plane U Stride Register. Stride of the second planar of the overlay if one exists. **Note:** This register is double buffered.	0x00000000
dcregOverlayVStride0	DISP_BA+0x1900	R/W	Overlay Third Plane V Stride Register. Stride of the third planar of the overlay if one exists. **Note:** This register is double buffered.	0x00000000
dcregOverlayClearValue0	DISP_BA+0x1940	R/W	Overlay Clear Value Register. Clear value used when dcregOverlayConfig.Clear is enabled. Format is A8R8G8B8. **Note:** This register is double buffered.	0x00000000
dcregOverlayIndexColorTableIndex0	DISP_BA+0x1980	W	Overlay Index Color Table Index Register. Index into overlay index color table. See dcregOverlayIndexColorTableData for more information.	0x00000000
dcregOverlayIndexColorTableData0	DISP_BA+0x19C0	W	Index Color Table Data Translation Register. Translation values for the index color table of the overlay. When this register gets written, the data gets stored in the index color table at the index specified by the dcregOverlayIndexColorTableIndex register. After the register is written, the index gets incremented. **Note:** This register is double buffered.	0x00000000
dcregFrameBufferClearValue0	DISP_BA+0x1A18	R/W	Framebuffer Clear Value Register. Clear value used when dcregFrameBufferConfig. Clear is enabled, format is A8R8G8B8. **Note:** This register is double buffered.	0x00000603
dcregModuleClockGatingControl0	DISP_BA+0x1A28	R/W	Clock Gating Module Control Register. Module level clock gating control for video and overlay.	0x00000000
dcregLatencyCounter0	DISP_BA+0x1A30	R/W	Latency Counter Register.	0x00000000
dcregQos0	DISP_BA+0x1A38	R/W	Quality of Service Latency Value Register.	0x00000000
dcregMpuIntfCmd0	DISP_BA+0x1C40	R/W	MPU Command Configuration Register.	0x00000000
dcregMpuIntfReadPara00	DISP_BA+0x1C48	R	MPU Read Parameter0 Register.	0x00000000
dcregMpuIntfReadPara10	DISP_BA+0x1C50	R	MPU Read Parameter1 Register.	0x00000000

dcregMpuIntfReadPara20	DISP_BA+0x1C58	R	MPU Read Parameter2 Register.	0x00000000
dcregMpuIntfReadPara30	DISP_BA+0x1C60	R	MPU Read Parameter3 Register.	0x00000000
dcregMpuIntfReadPara40	DISP_BA+0x1C68	R	MPU Read Parameter4 Register.	0x00000000
dcregMpuIntfReadStatus0	DISP_BA+0x1C70	R	MPU Read Parameter Status Register.	0x00000000
dcregMpuIntfConfig0	DISP_BA+0x1C78	R/W	MPU Interface Configuration Register.	0x00000000
dcregMpuIntfFrame0	DISP_BA+0x1C80	R/W	MPU Frame Configuration Register. **Note:** This register is double buffered.	0x00000000
dcregMpuIntfACWrI800	DISP_BA+0x1C88	R/W	MPU Write AC Characteristics I80 Register.	0x00000000
dcregMpuIntfACRdI800	DISP_BA+0x1C90	R/W	MPU Read AC Characteristics I80 Register.	0x00000000
dcregMpuIntfACWrM680	DISP_BA+0x1C98	R/W	MPU Write AC Characteristics M68 Register.	0x00000000
dcregMpuIntfACRdM680	DISP_BA+0x1CA0	R/W	MPU Read AC Characteristics M68 Register.	0x00000000
dcregMpuIntfACVsyncCSX0	DISP_BA+0x1CA8	R/W	MPU CSX Assert Register.	0x00000000
dcregFrameBufferROIOrigin0	DISP_BA+0x1CB0	R/W	Framebuffer Region of Interest Origin Register. **Note:** This register is double buffered.	0x00000000
dcregFrameBufferROISize0	DISP_BA+0x1CB8	R/W	Framebuffer Region of Interest Size Register. **Note:** This register is double buffered.	0x00000000
dcregFrameBufferConfigEx0	DISP_BA+0x1CC0	R/W	Framebuffer Configuration Extensions Register. **Note:** This register is double buffered.	0x00000000
dcregOverlayROIOrigin0	DISP_BA+0x1D00	R/W	Overlay Region of Interest Origin Register. **Note:** This register is double buffered.	0x00000000
dcregOverlayROISize0	DISP_BA+0x1D40	R/W	Overlay Region of Interest Size Register. **Note:** This register is double buffered.	0x00000000

6.40.7 Register Description

AQHiClockControl Register (AQHiClockControl)

Register	Offset	R/W	Description	Reset Value
AQHiClockControl	DISP_BA+0x0000	R/W	Clock Control Register	0x00000900

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		DISABLE_RAM_POWER_OP	SOFT_RESET	DISABLE_DEBUG_REGISTER	DISABLE_RAM_CLOCK_GA	FSCALE_CMD_LOAD	FSCALE_VAL
7	6	5	4	3	2	1	0
FSCALE_VAL						CLKDC_DIS	Reserved

Bits	Description	
[31:14]	**Reserved**	Reserved.
[13]	**DISABLE_RAM_POWER_OPTIMIZATION**	Disables ram power optimization.
[12]	**SOFT_RESET**	Soft resets the IP.
[11]	**DISABLE_DEBUG_REGISTERS**	**Disable debug registers** If this bit is 1, debug registers are clock gated(reset=1).
[10]	**DISABLE_RAM_CLOCK_GATING**	Disables clock gating for rams.
[9]	**FSCALE_CMD_LOAD**	**Core clock frequency scale value enable** When writing a 1 to this bit, it updates the frequency scale factor with the value FSCALE_VAL[6:0] The bit must be set back to 0 after that If this bit is set and FSCALE_VAL=0 (an invalid combination), the HREADYOUT output signal will get stuck to 0.
[8:2]	**FSCALE_VAL**	Core clock frequency scale value.
[1]	**CLKDC_DIS**	Disable DC clock.
[0]	**Reserved**	Reserved.

DcregFrameBufferAddress0 Register (DcregFrameBufferAddress0)

Register	Offset	R/W	Description	Reset Value
dcregFrameBufferAddress0	DISP_BA+0x1400	R/W	Framebuffer Start Address Register. Starting address of the framebuffer. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
TYPE	ADDRESS						
23	22	21	20	19	18	17	16
ADDRESS							
15	14	13	12	11	10	9	8
ADDRESS							
7	6	5	4	3	2	1	0
ADDRESS							

Bits	Description	
[31]	**TYPE**	0 = SYSTEM 1 = VIRTUAL_SYSTEM
[30:0]	**ADDRESS**	Framebuffer Start Address

DcregFrameBufferStride0 Register (DcregFrameBufferStride0)

Register	Offset	R/W	Description	Reset Value
dcregFrameBufferStride0	DISP_BA+0x1408	R/W	Framebuffer Stride Register. Stride of the framebuffer in bytes. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							STRIDE
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
STRIDE							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
STRIDE							

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16:0]	**STRIDE**	Number of bytes from start of one line to next line.

DcregDisplayDitherConfig0 Register (DcregDisplayDitherConfig0)

Register	Offset	R/W	Description	Reset Value
dcregDisplayDitherConfig0	DISP_BA+0x1410	R/W	Display Dither Configuration Register. Configuration register for dithering. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
ENABLE	**Reserved**						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31]	**ENABLE**	Enabling dithering allows R8G8B8 modes to show better on panels with less bits- per-pixel **Note:** This field is double buffered.
[30:0]	**Reserved**	Reserved.

DcregPanelConfig0 Register (DcregPanelConfig0)

Register	Offset	R/W	Description	Reset Value
dcregPanelConfig0	DISP_BA+0x1418	R/W	Panel Configuration Register. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						CLOCK_POLARITY	CLOCK
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		DATA_POLARITY	DATA_ENABLE	Reserved		DE_POLARITY	DE

Bits	Description	
[31:10]	**Reserved**	Reserved.
[9]	**CLOCK_POLARITY**	**Clock polarity** 0 = POSITIVE 1 = NEGATIVE
[8]	**CLOCK**	**Clock enabled/disabled** 0 = DISABLED 1 = ENABLED
[7:6]	**Reserved**	Reserved.
[5]	**DATA_POLARITY**	**Data polarity** 0 = POSITIVE 1 = NEGATIVE
[4]	**DATA_ENABLE**	**Data enabled/disabled** 0 = DISABLED 1 = ENABLED
[3:2]	**Reserved**	Reserved.
[1]	**DE_POLARITY**	**Data Enable polarity** 0 = POSITIVE 1 = NEGATIVE
[0]	**DE**	**Data Enable enabled/disabled** 0 = DISABLED 1 = ENABLED

DcregDisplayDitherTableLow0 Register (DcregDisplayDitherTableLow0)

Register	Offset	R/W	Description	Reset Value
dcregDisplayDitherTableLow0	DISP_BA+0x1420	R/W	Display Dither Table Register. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
Y1_X3				Y1_X2			
23	22	21	20	19	18	17	16
Y1_X1				Y1_X0			
15	14	13	12	11	10	9	8
Y0_X3				Y0_X2			
7	6	5	4	3	2	1	0
Y0_X1				Y0_X0			

Bits	Description	
[31:28]	**Y1_X3**	Dither threshold value for x,y=3,1.
[27:24]	**Y1_X2**	Dither threshold value for x,y=2,1.
[23:20]	**Y1_X1**	Dither threshold value for x,y=1,1.
[19:16]	**Y1_X0**	Dither threshold value for x,y=0,1.
[15:12]	**Y0_X3**	Dither threshold value for x,y=3,0.
[11:8]	**Y0_X2**	Dither threshold value for x,y=2,0.
[7:4]	**Y0_X1**	Dither threshold value for x,y=1,0.
[3:0]	**Y0_X0**	Dither threshold value for x,y=0,0.

DcregDisplayDitherTableHigh0 Register (DcregDisplayDitherTableHigh0)

Register	Offset	R/W	Description	Reset Value
dcregDisplayDitherTableHigh0	DISP_BA+0x1428	R/W	Display Dither Table Register. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
Y3_X3				Y3_X2			
23	22	21	20	19	18	17	16
Y3_X1				Y3_X0			
15	14	13	12	11	10	9	8
Y2_X3				Y2_X2			
7	6	5	4	3	2	1	0
Y2_X1				Y2_X0			

Bits	Description	
[31:28]	**Y3_X3**	Dither threshold value for x,y=3,3.
[27:24]	**Y3_X2**	Dither threshold value for x,y=2,3.
[23:20]	**Y3_X1**	Dither threshold value for x,y=1,3.
[19:16]	**Y3_X0**	Dither threshold value for x,y=0,3.
[15:12]	**Y2_X3**	Dither threshold value for x,y=3,2.
[11:8]	**Y2_X2**	Dither threshold value for x,y=2,2.
[7:4]	**Y2_X1**	Dither threshold value for x,y=1,2.
[3:0]	**Y2_X0**	Dither threshold value for x,y=0,2.

DcregHDisplay0 Register (DcregHDisplay0)

Register	Offset	R/W	Description	Reset Value
dcregHDisplay0	DISP_BA+0x1430	R/W	Horizontal Total and Display End Counter Register. **Note:** This register is double buffered	0x00000000

31	30	29	28	27	26	25	24
Reserved	**TOTAL**						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TOTAL							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**DISPLAY_END**						
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DISPLAY_END							

Bits	Description	
[31]	**Reserved**	Reserved.
[30:16]	**TOTAL**	Total number of horizontal pixels.
[15]	**Reserved**	Reserved.
[14:0]	**DISPLAY_END**	Visible number of horizontal pixels.

DcregHSync0 Register (DcregHSync0)

Register	Offset	R/W	Description	Reset Value
dcregHSync0	DISP_BA+0x1438	R/W	Horizontal Sync Counter Register. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
POLARITY	PULSE	END					
23	22	21	20	19	18	17	16
END							
15	14	13	12	11	10	9	8
END	START						
7	6	5	4	3	2	1	0
START							

Bits	Description	
[31]	**POLARITY**	**Polarity of the horizontal sync pulse** 0 = POSITIVE 1 = NEGATIVE
[30]	**PULSE**	**Horizontal sync pulse control** 0 = DISABLED 1 = ENABLED
[29:15]	**END**	End of horizontal sync pulse.
[14:0]	**START**	Start of horizontal sync pulse.

DcregVDisplay0 Register (DcregVDisplay0)

Register	Offset	R/W	Description	Reset Value
dcregVDisplay0	DISP_BA+0x1440	R/W	Vertical Total and Display End Counter Register. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
Reserved	**TOTAL**						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TOTAL							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**DISPLAY_END**						
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DISPLAY_END							

Bits	Description	
[31]	**Reserved**	Reserved.
[30:16]	**TOTAL**	Total number of vertical lines.
[15]	**Reserved**	Reserved.
[14:0]	**DISPLAY_END**	Visible number of vertical lines

DcregVSync0 Register (DcregVSync0)

Register	Offset	R/W	Description	Reset Value
dcregVSync0	DISP_BA+0x1448	R/W	Vertical Sync Counter Register. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
POLARITY	PULSE	END					
23	22	21	20	19	18	17	16
END							
15	14	13	12	11	10	9	8
END	START						
7	6	5	4	3	2	1	0
START							

Bits	Description	
[31]	**POLARITY**	**Polarity of the vertical sync pulse** 0 = POSITIVE 1 = NEGATIVE
[30:30]	**PULSE**	**Vertical sync pulse control** 0 = DISABLED 1 = ENABLED
[29:15]	**END**	End of the vertical sync pulse.
[14:0]	**START**	Start of the vertical sync pulse

DcregDisplayCurrentLocation0 Register (DcregDisplayCurrentLocation0)

Register	Offset	R/W	Description	Reset Value
dcregDisplayCurrentLocation0	DISP_BA+0x1450	R	Display Current Location Register. Current x,y location of display controller.	0x00000000

31	30	29	28	27	26	25	24
Y							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Y							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
X							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
X							

Bits	Description	
[31:16]	**Y**	Current Y location.
[15:0]	**X**	Current X location.

DcregGammaIndex0 Register (DcregGammaIndex0)

Register	Offset	R/W	Description	Reset Value
dcregGammaIndex0	DISP_BA+0x1458	W	Gamma Table Index Register. Index into gamma table. See dcregGammaData for more information.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INDEX							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**INDEX**	Index into gamma table.

DcregGammaData0 Register (DcregGammaData0)

Register	Offset	R/W	Description	Reset Value
dcregGammaData0	DISP_BA+0x1460	W	Gamma Data Translation Register. Translation values for the gamma table. When this register gets written, the data gets stored in the gamma table at the index specified by the dcregGammaIndex register. After the register is written, the index gets incremented.	0x00000000

31	30	29	28	27	26	25	24
Reserved		RED					
23	22	21	20	19	18	17	16
RED				GREEN			
15	14	13	12	11	10	9	8
GREEN						BLUE	
7	6	5	4	3	2	1	0
BLUE							

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29:20]	**READ**	Red translation value.
[19:10]	**GREEN**	Green translation value.
[9:0]	**BLUE**	Blue translation value.

DcregCursorConfig Register (DcregCursorConfig)

Register	Offset	R/W	Description	Reset Value
dcregCursorConfig	DISP_BA+0x1468	R/W	Cursor Configuration Register.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			HOT_SPOT_X				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			HOT_SPOT_Y				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			DISPLAY	Reserved		FORMAT	

Bits	Description	
[31:21]	**Reserved**	Reserved.
[20:16]	**HOT_SPOT_X**	**Horizontal offset to cursor hotspot Note:** This field is double buffered.
[15:13]	**Reserved**	Reserved.
[12:8]	**HOT_SPOT_Y**	**Vertical offset to cursor hotspot Note:** This field is double buffered.
[7:5]	**Reserved**	Reserved.
[4]	**DISPLAY**	Display Controller owning the cursor.
[3:2]	**Reserved**	Reserved.
[1:0]	**FORMAT**	**Format of the cursor Note:** This field is double buffered.

DcregCursorAddress Register (DcregCursorAddress)

Register	Offset	R/W	Description	Reset Value
dcregCursorAddress	DISP_BA+0x146C	R/W	Cursor Address Register. Address of the cursor shape. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
TYPE	**ADDRESS**						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ADDRESS							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ADDRESS							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADDRESS							

Bits	Description	
[31]	**TYPE**	0 = SYSTEM 1 = VIRTUAL_SYSTEM
[30:0]	**ADDRESS**	Cursor Address

DcregCursorLocation Register (DcregCursorLocation)

Register	Offset	R/W	Description	Reset Value
dcregCursorLocation	DISP_BA+0x1470	R/W	Cursor Location Register. Location of the cursor on the owning display. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
Reserved	**Y**						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Y							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**X**						
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
X							

Bits	Description	
[31]	**Reserved**	Reserved.
[30:16]	**Y**	Y location of cursor's hotspot.
[15]	**Reserved**	Reserved.
[14:0]	**X**	X location of cursor's hotspot.

DcregCursorBackground Register (DcregCursorBackground)

Register	Offset	R/W	Description	Reset Value
dcregCursorBackground	DISP_BA+0x1474	R/W	Masked Cursor Background Color Register. The background color for Masked cursors. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RED							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
GREEN							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BLUE							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**RED**	Red value.
[15:8]	**GREEN**	Green value.
[7:0]	**BLUE**	Blue value.

DcregCursorForeground Register (DcregCursorForeground)

Register	Offset	R/W	Description	Reset Value
dcregCursorForeground	DISP_BA+0x1478	R/W	Masked Cursor Foreground Color Register. The foreground color for Masked cursors. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RED							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
GREEN							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BLUE							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**RED**	Red value.
[15:8]	**GREEN**	Green value.
[7:0]	**BLUE**	Blue value.

DcregDisplayIntr Register (DcregDisplayIntr)

Register	Offset	R/W	Description	Reset Value
dcregDisplayIntr	DISP_BA+0x147C	R	Display Interrupt Register	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**DISP0**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**DISP0**	Display0 interrupt(read only)

DcregDisplayIntrEnable Register (DcregDisplayIntrEnable)

Register	Offset	R/W	Description	Reset Value
dcregDisplayIntrEnable	DISP_BA+0x1480	R/W	Display Interrupt Enable Register. The interrupt enable register for display_0. **Note:** Interrupt enable for register dcregDisplayIntr.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							DISP0

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**DISP0**	Display0 interrupt enable (read only)

DcregCursorModuleClockGatingControl Register (DcregCursorModuleClockGatingControl)

Register	Offset	R/W	Description	Reset Value
dcregCursorModuleClockGatingControl	DISP_BA+0x1484	R/W	Clock Gating Control for Cursor Register. Module level clock gating control for cursor.	0x00000001

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							DISABLE_MODULE_CLOCK

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**DISABLE_MODULE_CLOCK_GATING_CURSOR**	**Disable module clock gating cursor** 0 = ENABLED 1 = DISABLED

DcregGeneralConfig0 Register (DcregGeneralConfig0)

Register	Offset	R/W	Description	Reset Value
dcregGeneralConfig0	DISP_BA+0x14B0	R/W	General Miscellaneous Configuration Register. Misc option configuration. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				DISABLE_IDLE	STALL_OUTPUT_WHEN_UN	ENDIAN_CONTROL	

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3]	**DISABLE_IDLE**	**Disable idle signal** 0 = DISABLED 1 = ENABLED
[2]	**STALL_OUTPUT_WHEN_UNDERFLOW**	**If enabled, when FIFO underflow happens, output is stalled** 0 = DISABLED 1 = ENABLED
[1:0]	**ENDIAN_CONTROL**	**Control endian swapping** 0 = NO_SWAP 1 = SWAP_WORD 2 = SWAP_DWORD 3 = SWAP_DDWORD

DcregDpiConfig0 Register (DcregDpiConfig0)

Register	Offset	R/W	Description	Reset Value
dcregDpiConfig0	DISP_BA+0x14B8	R/W	DPI Configuration Register. The configuration register for DPI output. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					DPI_DATA_FORMAT		

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2:0]	**DPI_DATA_FORMAT**	**DPI interface data format Refer to DPI spec 'Interface color coding' for details** 0 = D16CFG1 1 = D16CFG2 2 = D16CFG3 3 = D18CFG1 4 = D18CFG2 5 = D24

DcregDebugCounterSelect0 Register (DcregDebugCounterSelect0)

Register	Offset	R/W	Description	Reset Value
dcregDebugCounterSelect0	DISP_BA+0x14D0	R/W	Debug Counter Selection Register.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SELECT							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**SELECT**	**Write a value to this field to pick up from 0~255 counters Then the counter will be on dcregDebugCounterValue** 00 = TOTAL_AXI_VIDEO_RD_REQ_CNT * video read request number * 01 = TOTAL_AXI_VIDEO_RD_LAST_CNT * video read return data last number * 02 = TOTAL_AXI_VIDEO_REQ_BURST_CNT * video number of 16 byte burst requests bytes * 03 = TOTAL_AXI_VIDEO_RD_BURST_CUNT * video number of 16 byte burst read return data * 04 = TOTAL_PIXEL_CNT * total pixels sent * 05 = TOTAL_FRAME_CNT * total frames sent * 06 = TOTAL_INPUT_DBI_CMD_CNT * total DBI input command * 07 = TOTAL_OUTPUT_DBI_CMD_CNT * total DBI output command * 08 = DEBUG_SIGNALS0 * debug signals * 09 = TOTAL_AXI_OVERLAY0_RD_REQ_CNT * overlay read request number * 0A = TOTAL_AXI_OVERLAY0_RD_LAST_CNT * overlay read return data last number * 0B = TOTAL_AXI_OVERLAY0_REQ_BURST_CNT * overlay number of 16 byte bursts of request bytes * 0C = TOTAL_AXI_OVERLAY0_RD_BURST_CUNT * overlay number of 16 bytes bursts of read return data * 0D = DEBUG_SIGNALS_FREE_POOL 0E = DEBUG_SIGNALS_INFOBUF_RD 0F = DEBUG_SIGNALS_INFOBUF_WR0 10 = DEBUG_SIGNALS_INFOBUF_WR1 11 = DEBUG_SIGNALS_INFOBUF_WR2 12 = DEBUG_SIGNALS_INFOBUF_WR3 13 = DEBUG_SIGNALS_OVERLAY0_FREE_POOL 14 = DEBUG_SIGNALS_OVERLAY0_INFOBUF_RD 15 = DEBUG_SIGNALS_OVERLAY0_INFOBUF_WR0 16 = DEBUG_SIGNALS_OVERLAY0_INFOBUF_WR1 17 = DEBUG_SIGNALS_OVERLAY0_INFOBUF_WR2 18 = DEBUG_SIGNALS_OVERLAY0_INFOBUF_WR3 19 = OVERLAY_BLEND_DEBUGSIGNALS 1A = CURSOR_BLEND_DEBUGSIGNALS 1C = VIDEO_WALKER_DEBUGSIGNALS_1 1D = VIDEO_WALKER_DEBUGSIGNALS_2 1E = VIDEO_WALKER_DEBUGSIGNALS_3 1F = OVERLAY_WALKER_DEBUGSIGNALS_1 20 = OVERLAY_WALKER_DEBUGSIGNALS_2 21 = OVERLAY_WALKER_DEBUGSIGNALS_3 FF = RESET_ALL_DEBUG_COUNTERS * Reset all debug counters *.

DcregDebugCounterValue0 Register (DcregDebugCounterValue0)

Register	Offset	R/W	Description	Reset Value
dcregDebugCounterValue0	DISP_BA+0x14D8	R	Debug Counter Value Register	0x00000000

31	30	29	28	27	26	25	24
VALUE							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
VALUE							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
VALUE							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VALUE							

Bits	Description	
[31:0]	**VALUE**	Selected debug counter value

DcregFrameBufferColorKey0 Register (DcregFrameBufferColorKey0)

Register	Offset	R/W	Description	Reset Value
dcregFrameBufferColorKey0	DISP_BA+0x1508	R/W	Framebuffer Color Key Start Address Register. Start of color key range of framebuffer. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
ALPHA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RED							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
GREEN							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BLUE							

Bits	Description	
[31:24]	**ALPHA**	ALPHA value
[23:16]	**RED**	RED value
[15:8]	**GREEN**	GREEN value
[7:0]	**BLUE**	BLUE value

DcregFrameBufferColorKeyHigh0 Register (DcregFrameBufferColorKeyHigh0)

Register	Offset	R/W	Description	Reset Value
dcregFrameBufferColorKeyHigh0	DISP_BA+0x1510	R/W	Framebuffer Color Key End Address Register. End of color key range of framebuffer. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
ALPHA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RED							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
GREEN							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BLUE							

Bits	Description	
[31:24]	**ALPHA**	ALPHA value
[23:16]	**RED**	RED value
[15:8]	**GREEN**	GREEN value
[7:0]	**BLUE**	BLUE value

DcregFrameBufferConfig0 Register (DcregFrameBufferConfig0)

Register	Offset	R/W	Description	Reset Value
dcregFrameBufferConfig0	DISP_BA+0x1518	R/W	Framebuffer Configuration Register. Framebuffer attribute configuration. **Note:** This register is double buffered. Some fields are double buffered.	0x00000000

31	30	29	28	27	26	25	24
FORMAT						UV_SWIZZLE	SWIZZLE
23	22	21	20	19	18	17	16
SWIZZLE	Reserved						YUV
15	14	13	12	11	10	9	8
YUV		Reserved			TRANSPARENCY		CLEAR
7	6	5	4	3	2	1	0
Reserved	FLIP_IN_PROGRESS	UNDERFLOW	RESET	VALID	GAMMA	Reserved	OUTPUT

Bits	Description	
[31:26]	**FORMAT**	**The format of the framebuffer** 00 = X4R4G4B4 01 = A4R4G4B4 02 = X1R5G5B5 03 = A1R5G5B5 04 = R5G6B5 05 = X8R8G8B8 06 = A8R8G8B8 07 = YUY2 08 = UYVY 09 = INDEX8 0A = MONOCHROME 0F = YV12 10 = A8 11 = NV12 12 = NV16 13 = RG16 14 = R8 15 = NV12_10BIT 16 = A2R10G10B10 17 = NV16_10BIT 18 = INDEX1 19 = INDEX2 1A = INDEX4 1B = P010 1C = NV12_10BIT_L1 1D = NV16_10BIT_L1
[25]	**UV_SWIZZLE**	UV swizzle type
[24:23]	**SWIZZLE**	0 = ARGB 1 = RGBA 2 = ABGR 3 = BGRA
[22:17]	**Reserved**	Reserved.

[16:14]	**YUV**	**YUV standard** 1 = SELECT_709 * BT709 * 3 = SELECT_2020 * BT2020 *
[13:11]	**Reserved**	Reserved.
[10:9]	**TRANSPARENCY**	**Transparency of framebuffer** 0 = OPAQUE 1 = MASK 2 = KEY
[8]	**CLEAR**	**When enabled, the pixel value of the framebuffer comes from dcregFrameBufferClearValue; otherwise, the pixel value comes from memory** 0 = DISABLED 1 = ENABLED
[7]	**Reserved**	Reserved.
[6]	**FLIP_IN_PROGRESS**	**0: NO 1: YES When the framebuffer address gets written to, this bit gets set to one It will be reset to zero at the start of the next VBLANK when the registers gets copied into the working set This field is (READ ONLY)** 0 = NO 1 = YES.
[5]	**UNDERFLOW**	**0: NO 1: YES When the display FIFO underflows, this bit gets set to one Reading this register will reset it back to zero This field is READ ONLY** 0 = NO 1 = YES.
[4]	**RESET**	**Enable reset for the display controller** 0 = See below for DPI. For DBI,this field should be 0. 1: RESET. Enable DPI timing, start a DPI transfer. Write 0 to this bit to reset the display controller, then configure the other registers and lastly write a 1 to this bit to let the display controller start. When the display controller starts, it begins at VBLANK_START, and all registers get flopped to the working set at VSYNC_END. Counters will be reset to the end of HSYNC and VSYNC. (Refer the Timing Diagram provided in the Hardware Feature document). For DBI, do not write 1 to this field. Use dcregDbiReset DBI_IF_LEVEL_RESET instead. This bit is WRITE ONLY. 1 = RESET.
[3]	**VALID**	**The valid field defines whether we can copy a new set of registers at the next VBLANK or not This ensures a frame will always start with a valid working set if this register is programmed last, which reduces the need for SW to wait for the start of a VBLANK signal in order to ensure all states are loaded before the next VBLANK** 0 = WORKING 1 = PENDING.
[2]	**GAMMA**	**When Gamma is enabled, the R, G, and B channels will be routed through the Gamma LUT to perform gamma correction Note: This field is double buffered** 0 = DISABLED 1 = ENABLED.
[1]	**Reserved**	Reserved.
[0]	**OUTPUT**	**When Output is enabled, pixels will be displayed When Output is disabled, all pixels will be black This allows a panel to have the correct timing but without any pixels Note: This field is double buffered** 0 = DISABLED 1 = ENABLED.

DcregFrameBufferBGColor0 Register (DcregFrameBufferBGColor0)

Register	Offset	R/W	Description	Reset Value
dcregFrameBufferBGColor0	DISP_BA+0x1528	R/W	Framebuffer Background Color Register. Background color used when a pixel from the framebuffer falls outside of the range of color key.	0x00000000

31	30	29	28	27	26	25	24
ALPHA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RED							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
GREEN							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BLUE							

Bits	Description	
[31:24]	**ALPHA**	ALPHA value
[23:16]	**RED**	RED value
[15:8]	**GREEN**	GREEN value
[7:0]	**BLUE**	BLUE value

DcregFrameBufferUPlanarAddress0 Register (DcregFrameBufferUPlanarAddress0)

Register	Offset	R/W	Description	Reset Value
dcregFrameBufferUPlanarAddress0	DISP_BA+0x1530	R/W	Framebuffer Second Plane U Start Address Register. Starting address of the second planar (often the U plane) of the framebuffer if one exists.	0x00000000

31	30	29	28	27	26	25	24
TYPE	ADDRESS						
23	22	21	20	19	18	17	16
ADDRESS							
15	14	13	12	11	10	9	8
ADDRESS							
7	6	5	4	3	2	1	0
ADDRESS							

Bits	Description	
[31]	**TYPE**	0 = SYSTEM 1 = VIRTUAL_SYSTEM
[30:0]	**ADDRESS**	Framebuffer Second Plane U Start Address

DcregFrameBufferVPlanarAddress0 Register (DcregFrameBufferVPlanarAddress0)

Register	Offset	R/W	Description	Reset Value
dcregFrameBufferVPlanarAddress0	DISP_BA+0x1538	R/W	Framebuffer Third Plane V Start Address Register. Starting address of the third planar (often the V plane) of the framebuffer if one exists. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
TYPE	ADDRESS						
23	22	21	20	19	18	17	16
ADDRESS							
15	14	13	12	11	10	9	8
ADDRESS							
7	6	5	4	3	2	1	0
ADDRESS							

Bits	Description	
[31]	**TYPE**	0 = SYSTEM 1 = VIRTUAL_SYSTEM
[30:0]	**ADDRESS**	Framebuffer Third Plane V Start Address

DcregOverlayConfig Register (DcregOverlayConfig0)

Register	Offset	R/W	Description	Reset Value
dcregOverlayConfig0	DISP_BA+0x1540	R/W	Overlay Configuration Register. Overlay attributes control. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
Reserved						CLEAR	ENABLE
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
UNDERFLOW	Reserved	FORMAT					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
UV_SWIZZLE	SWIZZLE		Reserved				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
YUV			Reserved			TRANSPARENCY	

Bits	Description	
[31:26]	**Reserved**	Reserved.
[25]	**CLEAR**	**When enabled, the pixel value of the overlay comes from dcregOverlayClearValue; otherwise the pixel value comes from memory** 0 = DISABLED 1 = ENABLED
[24]	**ENABLE**	**Enable this overlay layer** 0 = DISABLE 1 = ENABLE
[23]	**UNDERFLOW**	**When the overlay FIFO underflows, this bit gets set to one Reading this register will reset it back to zero** 0 = NO 1 = YES.
[22]	**Reserved**	Reserved.
[21:16]	**FORMAT**	**The format of the overlay** 00 = X4R4G4B4 01 = A4R4G4B4 02 = X1R5G5B5 03 = A1R5G5B5 04 = R5G6B5 05 = X8R8G8B8 06 = A8R8G8B8 07 = YUY2 08 = UYVY 09 = INDEX8 0A = MONOCHROME 0F = YV12 10 = A8 11 = NV12 12 = NV16 13 = RG16 14 = R8 15 = NV12_10BIT

		16 = A2R10G10B10 17 = NV16_10BIT 18 = INDEX1 19 = INDEX2 1A = INDEX4 1B = P010 1C = NV12_10BIT_L1 1D = NV16_10BIT_L1.
[15]	**UV_SWIZZLE**	UV swizzle type.
[14:13]	**SWIZZLE**	0 = ARGB 1 = RGBA 2 = ABGR 3 = BGRA
[12:8]	**Reserved**	Reserved.
[7:5]	**YUV**	**YUV standard** 1 = SELECT_709 * BT709 * 3 = SELECT_2020 * BT2020 *.
[4:2]	**Reserved**	Reserved.
[1:0]	**TRANSPARENCY**	**Transparency of the overlay** 0 = OPAQUE 1 = MASK 2 = KEY

DcregOverlayAlphaBlendConfig Register (DcregOverlayAlphaBlendConfig0)

Register	Offset	R/W	Description	Reset Value
dcregOverlayAlphaBlendConfig0	DISP_BA+0x1580	R/W	Overlay Alpha Blending Configuration Register. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DST_ALPHA_FACTOR	DST_BLENDING_MODE			DST_GLOBAL_ALPHA_MODE		DST_ALPHA_MODE	SRC_ALPHA_FACTOR
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SRC_BLENDING_MODE			SRC_GLOBAL_ALPHA_MODE		Reserved	DISABLE_ALPHA_BLEND	SRC_ALPHA_MODE

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15]	**DST_ALPHA_FACTOR**	**Dst Blending factor is calculated from Dst alpha** 0 = DISABLED 1 = ENABLED
[14:12]	**DST_BLENDING_MODE**	0 = ZERO 1 = ONE 2 = NORMAL 3 = INVERSED 4 = COLOR 5 = COLOR_INVERSED 6 = SATURATED_ALPHA 7 = SATURATED_DEST_ALPHA
[11:10]	**DST_GLOBAL_ALPHA_MODE**	0 = NORMAL 1 = GLOBAL 2 = SCALED
[9]	**DST_ALPHA_MODE**	0 = NORMAL 1 = INVERSED
[8]	**SRC_ALPHA_FACTOR**	**Src Blending factor is calculated from Src alpha** 0 = DISABLED 1 = ENABLED
[7:5]	**SRC_BLENDING_MODE**	0 = ZERO 1 = ONE 2 = NORMAL 3 = INVERSED 4 = COLOR 5 = COLOR_INVERSED 6 = SATURATED_ALPHA 7 = SATURATED_DEST_ALPHA
[4:3]	**SRC_GLOBAL_ALPHA_MODE**	0 = NORMAL 1 = GLOBAL 2 = SCALED

[2]	**Reserved**	Reserved.
[1]	**DISABLE_ALPHA_BLEND**	0 = DISABLED 1 = ENABLED
[0]	**SRC_ALPHA_MODE**	0 = NORMAL 1 = INVERSED

DcregOverlayAddress Register (DcregOverlayAddress0)

Register	Offset	R/W	Description	Reset Value
dcregOverlayAddress0	DISP_BA+0x15C0	R/W	Overlay Start Address Register. Starting address of the overlay. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
TYPE	ADDRESS						
23	22	21	20	19	18	17	16
ADDRESS							
15	14	13	12	11	10	9	8
ADDRESS							
7	6	5	4	3	2	1	0
ADDRESS							

Bits	Description	
[31]	**TYPE**	0 = SYSTEM 1 = VIRTUAL_SYSTEM
[30:0]	**ADDRESS**	Overlay Start Address

DcregOverlayStride Register (DcregOverlayStride0)

Register	Offset	R/W	Description	Reset Value
dcregOverlayStride0	DISP_BA+0x1600	R/W	Overlay Stride Register. Stride of the overlay in bytes. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							STRIDE
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
STRIDE							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
STRIDE							

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16:0]	**STRIDE**	Number of bytes from the start of one line to next line.

DcregOverlayTL Register (DcregOverlayTL0)

Register	Offset	R/W	Description	Reset Value
dcregOverlayTL0	DISP_BA+0x1640	R/W	Overlay Origin Register. Top left coordinate of the panel pixel where the overlay should start. Be aware there is no panning inside the overlay. **Note:** This register is double buffered	0x00000000

31	30	29	28	27	26	25	24
Reserved		Y					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Y							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Y	X						
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
X							

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29:15]	**Y**	Top boundary of overlay window.
[14:0]	**X**	Left boundary of overlay window.

DcregOverlayBR Register (DcregOverlayBR0)

Register	Offset	R/W	Description	Reset Value
dcregOverlayBR0	DISP_BA+0x1680	R/W	Overlay End Register. Bottom right coordinate of the panel pixel where the overlay should end. The border is inclusive. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
Reserved		Y					
23	22	21	20	19	18	17	16
Y							
15	14	13	12	11	10	9	8
Y	X						
7	6	5	4	3	2	1	0
X							

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29:15]	**Y**	Bottom boundary of overlay.
[14:0]	**X**	Right boundary of overlay.

DcregOverlaySrcGlobalColor Register (DcregOverlaySrcGlobalColor0)

Register	Offset	R/W	Description	Reset Value
dcregOverlaySrcGlobalColor0	DISP_BA+0x16C0	R/W	Overlay Source Global Color Register. Color value used when alpha blending process is configured to use global color for source. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
ALPHA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RED							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
GREEN							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BLUE							

Bits	Description	
[31:24]	**ALPHA**	ALPHA value
[23:16]	**RED**	RED value
[15:8]	**GREEN**	GREEN value
[7:0]	**BLUE**	BLUE value

DcregOverlayDstGlobalColor Register (DcregOverlayDstGlobalColor0)

Register	Offset	R/W	Description	Reset Value
dcregOverlayDstGlobalColor0	DISP_BA+0x1700	R/W	Overlay Destination Global Color Register. Color value used when alpha blending process is configured to use global color for destination. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
ALPHA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RED							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
GREEN							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BLUE							

Bits	Description	
[31:24]	**ALPHA**	ALPHA value
[23:16]	**RED**	RED value
[15:8]	**GREEN**	GREEN value
[7:0]	**BLUE**	BLUE value

DcregOverlayColorKey Register (DcregOverlayColorKey0)

Register	Offset	R/W	Description	Reset Value
dcregOverlayColorKey0	DISP_BA+0x1740	R/W	Overlay Color Key Start Address Register. Start of color key range for overlay. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
ALPHA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RED							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
GREEN							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BLUE							

Bits	Description	
[31:24]	**ALPHA**	ALPHA value
[23:16]	**RED**	RED value
[15:8]	**GREEN**	GREEN value
[7:0]	**BLUE**	BLUE value

DcregOverlayColorKeyHigh Register (DcregOverlayColorKeyHigh0)

Register	Offset	R/W	Description	Reset Value
dcregOverlayColorKeyHigh0	DISP_BA+0x1780	R/W	Overlay Color Key End Address Register. End of color key range for overlay. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
ALPHA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RED							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
GREEN							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BLUE							

Bits	Description	
[31:24]	**ALPHA**	ALPHA value
[23:16]	**RED**	RED value
[15:8]	**GREEN**	GREEN value
[7:0]	**BLUE**	BLUE value

DcregOverlaySize Register (DcregOverlaySize0)

Register	Offset	R/W	Description	Reset Value
dcregOverlaySize0	DISP_BA+0x17C0	R/W	Overlay Window Size Register. Window size of the overlay buffer in memory - in pixels. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
Reserved		HEIGHT					
23	22	21	20	19	18	17	16
HEIGHT							
15	14	13	12	11	10	9	8
HEIGHT	WIDTH						
7	6	5	4	3	2	1	0
WIDTH							

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29:15]	**HEIGHT**	Overlay height
[14:0]	**WIDTH**	Overlay width

DcregFrameBufferUStride0 Register (DcregFrameBufferUStride0)

Register	Offset	R/W	Description	Reset Value
dcregFrameBufferUStride0	DISP_BA+0x1800	R/W	Framebuffer Second Plane U Stride Register. Stride of the second planar (often the U plane) of the framebuffer if one exists. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							STRIDE
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
STRIDE							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
STRIDE							

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16:0]	**STRIDE**	Number of bytes from the start of one line to next line.

DcregFrameBufferVStride0 Register (DcregFrameBufferVStride0)

Register	Offset	R/W	Description	Reset Value
dcregFrameBufferVStride0	DISP_BA+0x1808	R/W	Framebuffer Third Plane V Stride Register. Stride of the third planar (often the V plane) of the framebuffer if one exists. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							STRIDE
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
STRIDE							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
STRIDE							

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16:0]	**STRIDE**	Number of bytes from the start of one line to next line.

DcregFrameBufferSize0 Register (DcregFrameBufferSize0)

Register	Offset	R/W	Description	Reset Value
dcregFrameBufferSize0	DISP_BA+0x1810	R/W	Framebuffer Size Register. Window size of the framebuffer in memory - in pixels. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
Reserved		HEIGHT					
23	22	21	20	19	18	17	16
HEIGHT							
15	14	13	12	11	10	9	8
HEIGHT	WIDTH						
7	6	5	4	3	2	1	0
WIDTH							

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29:15]	**HEIGHT**	Frame height
[14:0]	**WIDTH**	Frame width

DcregIndexColorTableIndex0 Register (DcregIndexColorTableIndex0)

Register	Offset	R/W	Description	Reset Value
dcregIndexColorTableIndex0	DISP_BA+0x1818	W	Index Color Table Index Register. Index into index color table. See dcregIndexColorTableData for more information.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INDEX							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**INDEX**	index into index color table.

DcregIndexColorTableData0 Register (DcregIndexColorTableData0)

Register	Offset	R/W	Description	Reset Value
dcregIndexColorTableData0	DISP_BA+0x1820	W	Index Color Table Data Translation Register. Translation values for the index color table. When this register gets written, the data gets stored in the index color table at the index specified by the dcregIndexColorTableIndex register. After the register is written, the index gets incremented. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
ALPHA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RED							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
GREEN							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BLUE							

Bits	Description	
[31:24]	**ALPHA**	ALPHA value
[23:16]	**RED**	RED value
[15:8]	**GREEN**	GREEN value
[7:0]	**BLUE**	BLUE value

DcregOverlayUPlanarAddress Register (DcregOverlayUPlanarAddress0)

Register	Offset	R/W	Description	Reset Value
dcregOverlayUPlanarAddress0	DISP_BA+0x1840	R/W	Overlay Second Plane U Start Address Register. Address of the second planar (often U plane) of the overlay if one exists. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
TYPE	ADDRESS						
23	22	21	20	19	18	17	16
ADDRESS							
15	14	13	12	11	10	9	8
ADDRESS							
7	6	5	4	3	2	1	0
ADDRESS							

Bits	Description	
[31]	**TYPE**	0 = SYSTEM 1 = VIRTUAL_SYSTEM
[30:0]	**ADDRESS**	Overlay Third Plane V Start Address

DcregOverlayVPlanarAddress Register (DcregOverlayVPlanarAddress0)

Register	Offset	R/W	Description	Reset Value
dcregOverlayVPlanarAddress0	DISP_BA+0x1880	R/W	Overlay Third Plane V Start Address Register. Address of the third planar (often V plane) of the overlay if one exists. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
TYPE	ADDRESS						
23	22	21	20	19	18	17	16
ADDRESS							
15	14	13	12	11	10	9	8
ADDRESS							
7	6	5	4	3	2	1	0
ADDRESS							

Bits	Description	
[31]	**TYPE**	0 = SYSTEM 1 = VIRTUAL_SYSTEM
[30:0]	**ADDRESS**	Overlay Second Plane U Stride

DcregOverlayUStride Register (DcregOverlayUStride0)

Register	Offset	R/W	Description	Reset Value
dcregOverlayUStride0	DISP_BA+0x18C0	R/W	Overlay Second Plane U Stride Register. Stride of the second planar of the overlay if one exists. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							STRIDE
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
STRIDE							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
STRIDE							

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16:0]	**STRIDE**	Number of bytes from the start of one line to next line.

DcregOverlayVStride Register (DcregOverlayVStride0)

Register	Offset	R/W	Description	Reset Value
dcregOverlayVStride0	DISP_BA+0x1900	R/W	Overlay Third Plane V Stride Register. Stride of the third planar of the overlay if one exists. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							STRIDE
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
STRIDE							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
STRIDE							

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16:0]	**STRIDE**	Number of bytes from start of one line to next line.

DcregOverlayClearValue Register (DcregOverlayClearValue0)

Register	Offset	R/W	Description	Reset Value
dcregOverlayClearValue0	DISP_BA+0x1940	R/W	Overlay Clear Value Register. Clear value used when dcregOverlayConfig.Clear is enabled. Format is A8R8G8B8. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
ALPHA							
23	22	21	20	19	18	17	16
RED							
15	14	13	12	11	10	9	8
GREEN							
7	6	5	4	3	2	1	0
BLUE							

Bits	Description	
[31:24]	**ALPHA**	ALPHA value
[23:16]	**RED**	RED value
[15:8]	**GREEN**	GREEN value
[7:0]	**BLUE**	BLUE value

DcregOverlayIndexColorTableIndex Register (DcregOverlayIndexColorTableIndex0)

Register	Offset	R/W	Description	Reset Value
dcregOverlayIndexColorTableIndex0	DISP_BA+0x1980	W	Overlay Index Color Table Index Register. Index into overlay index color table. See dcregOverlayIndexColorTableData for more information.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INDEX							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**INDEX**	Index into index color table.

DcregOverlayIndexColorTableData Register (DcregOverlayIndexColorTableData0)

Register	Offset	R/W	Description	Reset Value
dcregOverlayIndexColorTableData0	DISP_BA+0x19C0	W	Index Color Table Data Translation Register. Translation values for the index color table of the overlay. When this register gets written, the data gets stored in the index color table at the index specified by the dcregOverlayIndexColorTableIndex register. After the register is written, the index gets incremented. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
ALPHA							
23	22	21	20	19	18	17	16
RED							
15	14	13	12	11	10	9	8
GREEN							
7	6	5	4	3	2	1	0
BLUE							

Bits	Description	
[31:24]	**ALPHA**	ALPHA value
[23:16]	**RED**	RED value
[15:8]	**GREEN**	GREEN value
[7:0]	**BLUE**	BLUE value

DcregFrameBufferClearValue0 Register (DcregFrameBufferClearValue0)

Register	Offset	R/W	Description	Reset Value
dcregFrameBufferClearValue0	DISP_BA+0x1A18	R/W	Framebuffer Clear Value Register. Clear value used when dcregFrameBufferConfig. Clear is enabled, format is A8R8G8B8. **Note:** This register is double buffered.	0x00000603

31	30	29	28	27	26	25	24
ALPHA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RED							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
GREEN							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BLUE							

Bits	Description	
[31:24]	**ALPHA**	ALPHA value
[23:16]	**RED**	RED value
[15:8]	**GREEN**	GREEN value
[7:0]	**BLUE**	BLUE value

DcregModuleClockGatingControl0 Register (DcregModuleClockGatingControl0)

Register	Offset	R/W	Description	Reset Value
dcregModuleClockGatingControl0	DISP_BA+0x1A28	R/W	Clock Gating Module Control Register. Module level clock gating control for video and overlay.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						DISABLE_MODULE_CLOCK	Reserved
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						DISABLE_MODULE_CLOCK	DISABLE_MODULE_CLOCK

Bits	Description	
[31:10]	**Reserved**	Reserved.
[9]	**DISABLE_MODULE_CLOCK_GATING_WB_FIFO**	**Disable module clock gating WBFifo** 0 = ENABLE 1 = DISABLE.
[8:2]	**Reserved**	Reserved.
[1]	**DISABLE_MODULE_CLOCK_GATING_OVERLAY0**	**Disable module clock gating overlay0** 0 = ENABLE 1 = DISABLE.
[0]	**DISABLE_MODULE_CLOCK_GATING_VIDEO**	**Disable module clock gating video** 0 = ENABLE 1 = DISABLE.

DcregLatencyCounter0 Register (DcregLatencyCounter0)

Register	Offset	R/W	Description	Reset Value
dcregLatencyCounter0	DISP_BA+0x1A30	R/W	Latency Counter Register.	0x00000000

31	30	29	28	27	26	25	24
COUNTER							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
COUNTER							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
COUNTER							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
COUNTER							

Bits	Description	
[31:0]	**COUNTER**	Latency counter value which is used to judge whether latency is low or high.

DcregQos0 Register (DcregQos0)

Register	Offset	R/W	Description	Reset Value
dcregQos0	DISP_BA+0x1A38	R/W	Quality of Service Latency Value Register.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
HIGH				**LOW**			

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:4]	**HIGH**	If latency high, will send high value.
[3:0]	**LOW**	If latency low, will send low value.

DcregMpuIntfCmd0 Register (DcregMpuIntfCmd0)

Register	Offset	R/W	Description	Reset Value
dcregMpuIntfCmd0	DISP_BA+0x1C40	R/W	MPU Command Configuration Register.	0x00000000

31	30	29	28	27	26	25	24
CMD		START	Reserved		READ_COUNT		
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
REG_DATA							
7	6	5	4	3	2	1	0
REG_DATA							

Bits	Description	
[31:30]	**CMD**	0: Not a valid command, HW ignores the CMD/REGDATA/READCOUNT field Usage: only want to trigger HW, don't configure other fields 1: Write register by MPU interface 2: Write register parameter by MPU interface 3: Read register by MPU interface.
[29]	**START**	HW generates a pulse while writing 1 to this field, then HW starts to send display data to the LCD.
[28:27]	**Reserved**	Reserved.
[26:24]	**READ_COUNT**	**0 is invalid value, one read operation returns 1 data at least, Max value is 5** It is designed to prepare 5 registers to save returned data.
[23:16]	**Reserved**	Reserved.
[15:0]	**REG_DATA**	Register address or register configured parameter.

DcregMpuIntfReadPara00 Register (DcregMpuIntfReadPara00)

Register	Offset	R/W	Description	Reset Value
dcregMpuIntfReadPara00	DISP_BA+0x1C48	R	MPU Read Parameter0 Register.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						DATA	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DATA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATA							

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17:0]	**DATA**	Read register returned parameter0.

DcregMpuIntfReadPara10 Register (DcregMpuIntfReadPara10)

Register	Offset	R/W	Description	Reset Value
dcregMpuIntfReadPara10	DISP_BA+0x1C50	R	MPU Read Parameter1 Register.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						DATA	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DATA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATA							

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17:0]	**DATA**	Read register returned parameter1.

DcregMpuIntfReadPara20 Register (DcregMpuIntfReadPara20)

Register	Offset	R/W	Description	Reset Value
dcregMpuIntfReadPara20	DISP_BA+0x1C58	R	MPU Read Parameter2 Register.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						DATA	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DATA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATA							

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17:0]	**DATA**	Read register returned parameter2.

DcregMpuIntfReadPara30 Register (DcregMpuIntfReadPara30)

Register	Offset	R/W	Description	Reset Value
dcregMpuIntfReadPara30	DISP_BA+0x1C60	R	MPU Read Parameter3 Register.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						DATA	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DATA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATA							

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17:0]	**DATA**	Read register returned parameter3.

DcregMpuIntfReadPara40 Register (DcregMpuIntfReadPara40)

Register	Offset	R/W	Description	Reset Value
dcregMpuIntfReadPara40	DISP_BA+0x1C68	R	MPU Read Parameter4 Register.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						DATA	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DATA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATA							

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17:0]	**DATA**	Read register returned parameter4.

DcregMpuIntfReadStatus0 Register (DcregMpuIntfReadStatus0)

Register	Offset	R/W	Description	Reset Value
dcregMpuIntfReadStatus0	DISP_BA+0x1C70	R	MPU Read Parameter Status Register.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							DATA

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**DATA**	**Returned parameter readiness** 0 = UNREADY. The returned parameters of all read registers are not ready. 1 = READY. The returned parameters of all read registers are ready.

DcregMpuIntfConfig0 Register (DcregMpuIntfConfig0)

Register	Offset	R/W	Description	Reset Value
dcregMpuIntfConfig0	DISP_BA+0x1C78	R/W	MPU Interface Configuration Register.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			ENABLE_MPU_INTF	INTERFACE_RESET	DATA_MODE24_BIT	DCX_POLARITY	TE_POLARITY
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ENABLE_TE	VSYNC_POLARITY	ENABLE_VSYNC	INTERFACE_MODE	DATA_BUS_MODE	DATA_BUS_WIDTH		REGISTER_MODE

Bits	Description	
[31:13]	**Reserved**	Reserved.
[12]	**ENABLE_MPU_INTF**	**O:Disable; 1: Enable MPU interface** 0 = DISABLE 1 = ENABLE
[11]	**INTERFACE_RESET**	Write 1 to this field, HW will reset the MPU interface.
[10]	**DATA_MODE24_BIT**	**16 bit data bus, 24 bit data output** 0 = MODE0. Mode0: 2 pixels per 3 transfers, 1 = MODE1. Mode1: 1 pixel per 2 transfers.
[9]	**DCX_POLARITY**	**DCX polarity for command and data** 0 = MODE0. Mode0 : 0-command, 1-data, 1 = MODE1. Mode1 : 1-command, 0-data.
[8]	**TE_POLARITY**	**TE signal is positive or negative value** 0 = POSITIVE 1 = NEGATIVE
[7]	**ENABLE_TE**	**0: Disable; 1: Enable TE interface** 0 = DISABLE 1 = ENABLE
[6]	**VSYNC_POLARITY**	**VSYNC signal is negative or positive value** 0 = NEGATIVE 1 = POSITIVE
[5]	**ENABLE_VSYNC**	**0: Disable; 1: Enable VSYNC interface** 0 = DISABLE 1 = ENABLE
[4]	**INTERFACE_MODE**	System Interface.
[3]	**DATA_BUS_MODE**	**Data bus mode** 0 = MODE0. Mode0: 8-bit:DB[7:0]; 9-bit:DB[8:0]; 16-bit:DB[15:0]; 18-bit:DB[17:0], 1 = MODE1. Mode1: 8-bit:DB[17:10]; 9-bit:DB[17:9]; 16-bit:{DB[17:10],DB[8:1]}; 18-bit:DB[17:0].

[2:1]	**DATA_BUS_WIDTH**	**MPU interface bus width** 0 = WIDTH_8_BIT 1 = WIDTH_9_BIT 2 = WIDTH_16_BIT 3 = WIDTH_18_BIT
[0]	**REGISTER_MODE**	**0: 8 bit register mode, only dcregMpuIntfCmd[7:0] are valid Only send 8 bits valid data each time to LCD by MPU Interface 1: 16 bit register mode, only dcregMpuIntfCmd[15:0] are valid, Only send 16 bits valid data each time to the LCD by MPU Interface** 0 = MODE_8_BIT 1 = MODE_16_BIT

DcregMpuIntfFrame0 Register (DcregMpuIntfFrame0)

Register	Offset	R/W	Description	Reset Value
dcregMpuIntfFrame0	DISP_BA+0x1C80	R/W	MPU Frame Configuration Register. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				MPU_WRITE_BACK	DATA_FORMAT		FRAME_UPDATE

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3]	**MPU_WRITE_BACK**	**0: Disable MPU data write back 1: Enable MPU interface data write back** 0 = DISABLE 1 = ENABLE.
[2:1]	**DATA_FORMAT**	**0: 16 bit format R5G6B5 1: 18 bit format R6G6B6 2: 24 bit format R8G8B8. 3: Reserved** 0 = R5G6B5 1 = R6G6B6 2 = R8G8B8
[0]	**FRAME_UPDATE**	**0: When using VSync interface/TE mode, if this bit is 0, this frame data won't be updated 1: When using VSync interface/TE mode, if this bit is 1, this frame data will be updated** 0 = NO 1 = YES.

DcregMpuIntfACWrI800 Register (DcregMpuIntfACWrI800)

Register	Offset	R/W	Description	Reset Value
dcregMpuIntfACWrI800	DISP_BA+0x1C88	R/W	MPU Write AC Characteristics I80 Register.	0x00000000

31	30	29	28	27	26	25	24
Reserved		WRX_DE_ASSERT					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
WRX_DE_ASSERT				WRX_ASSERT			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
WRX_ASSERT						WR_PERIOD_I80	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WR_PERIOD_I80							

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29:20]	**WRX_DE_ASSERT**	**I80 system WRX de-assert cycle number** **Minimum number is 1** Default number is 0, 0 means 1024 cycle number Cycle unit is pixel clock
[19:10]	**WRX_ASSERT**	**I80 system WRX assert cycle number** **Minimum number is 1** Default number is 0, 0 means 1024 cycle number Cycle unit is pixel clock.
[9:0]	**WR_PERIOD_I80**	**I80 system write period cycle number** **Minimum number is 3** Default number is 0, 0 means 1024 cycle number Cycle unit is pixel clock.

DcregMpuIntfACRdI800 Register (DcregMpuIntfACRdI800)

Register	Offset	R/W	Description	Reset Value
dcregMpuIntfACRdI800	DISP_BA+0x1C90	R/W	MPU Read AC Characteristics I80 Register.	0x00000000

31	30	29	28	27	26	25	24
Reserved		RDX_DE_ASSERT					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RDX_DE_ASSERT				RDX_ASSERT			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RDX_ASSERT						RD_PERIOD_I80	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RD_PERIOD_I80							

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29:20]	**RDX_DE_ASSERT**	**I80 system RDX de-assert cycle number** **Minimum number is 1** Default number is 0, 0 means 1024 cycle number Cycle unit is pixel clock.
[19:10]	**RDX_ASSERT**	**I80 system RDX assert cycle number** **Minimum number is 1** Default number is 0, 0 means 1024 cycle number Cycle unit is pixel clock.
[9:0]	**RD_PERIOD_I80**	**I80 system read period cycle number** **Minimum number is 3** Default number is 0, 0 means 1024 cycle number Cycle unit is pixel clock.

DcregMpuIntfACWrM680 Register (DcregMpuIntfACWrM680)

Register	Offset	R/W	Description	Reset Value
dcregMpuIntfACWrM680	DISP_BA+0x1C98	R/W	MPU Write AC Characteristics M68 Register.	0x00000000

31	30	29	28	27	26	25	24
Reserved		WR_EDE_ASSERT					
23	22	21	20	19	18	17	16
WR_EDE_ASSERT				WR_EASSERT			
15	14	13	12	11	10	9	8
WR_EASSERT						WR_PERIOD_M68	
7	6	5	4	3	2	1	0
WR_PERIOD_M68							

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29:20]	**WR_EDE_ASSERT**	**M68 system write E de-assert cycle number** **Minimum number is 1** Default number is 0, 0 means 1024 cycle number Cycle unit is pixel clock.
[19:10]	**WR_EASSERT**	**M68 system write E assert cycle number** **Minimum number is 1** Default number is 0, 0 means 1024 cycle number Cycle unit is pixel clock.
[9:0]	**WR_PERIOD_M68**	**M68 system write period cycle number** **Minimum number is 3** Default number is 0, 0 means 1024 cycle number Cycle unit is pixel clock.

DcregMpuIntfACRdM680 Register (DcregMpuIntfACRdM680)

Register	Offset	R/W	Description	Reset Value
dcregMpuIntfACRdM680	DISP_BA+0x1CA0	R/W	MPU Read AC Characteristics M68 Register.	0x00000000

31	30	29	28	27	26	25	24
Reserved		RDX_DE_ASSERT					
23	22	21	20	19	18	17	16
RDX_DE_ASSERT				RDX_ASSERT			
15	14	13	12	11	10	9	8
RDX_ASSERT						RD_PERIOD_I80	
7	6	5	4	3	2	1	0
RD_PERIOD_I80							

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29:20]	**RDX_DE_ASSERT**	**M68 system RDX de-assert cycle number** **Minimum number is 1** Default number is 0, 0 means 1024 cycle number Cycle unit is pixel clock.
[19:10]	**RDX_ASSERT**	**M68 system RDX assert cycle number** **Minimum number is 1** Default number is 0, 0 means 1024 cycle number Cycle unit is pixel clock.
[9:0]	**RD_PERIOD_I80**	**M68 system read period cycle number** **Minimum number is 3** Default number is 0, 0 means 1024 cycle number Cycle unit is pixel clock.

DcregMpuIntfACVsyncCSX0 Register (DcregMpuIntfACVsyncCSX0)

Register	Offset	R/W	Description	Reset Value
dcregMpuIntfACVsyncCSX0	DISP_BA+0x1CA8	R/W	MPU CSX Assert Register.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						CSX_ASSERT	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CSX_ASSERT							

Bits	Description	
[31:10]	**Reserved**	Reserved.
[9:0]	**CSX_ASSERT**	**Add delay between VSYNC-high and CSX- low** This field specifies how many transfer-one-pixel period.

DcregFrameBufferROIOrigin0 Register (DcregFrameBufferROIOrigin0)

Register	Offset	R/W	Description	Reset Value
dcregFrameBufferROIOrigin0	DISP_BA+0x1CB0	R/W	Framebuffer Region of Interest Origin Register. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
Y							
23	22	21	20	19	18	17	16
Y							
15	14	13	12	11	10	9	8
X							
7	6	5	4	3	2	1	0
X							

Bits	Description	
[31:16]	**Y**	OSD/Rectangle start point Y coordinate.
[15:0]	**X**	OSD/Rectangle start point X coordinate.

DcregFrameBufferROISize0 Register (DcregFrameBufferROISize0)

Register	Offset	R/W	Description	Reset Value
dcregFrameBufferROISize0	DISP_BA+0x1CB8	R/W	Framebuffer Region of Interest Size Register. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
HEIGHT							
23	22	21	20	19	18	17	16
HEIGHT							
15	14	13	12	11	10	9	8
WIDTH							
7	6	5	4	3	2	1	0
WIDTH							

Bits	Description	
[31:16]	**HEIGHT**	OSD/Rectangle height.
[15:0]	**WIDTH**	OSD/Rectangle width.

DcregFrameBufferConfigEx0 Register (DcregFrameBufferConfigEx0)

Register	Offset	R/W	Description	Reset Value
dcregFrameBufferConfigEx0	DISP_BA+0x1CC0	R/W	Framebuffer Configuration Extensions Register. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**ENABLE_ROI**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**ENABLE_ROI**	**0:Disable' 1: Enable ROI** 0 = DISABLE 1 = ENABLE

DcregOverlayROIOrigin Register (DcregOverlayROIOrigin0)

Register	Offset	R/W	Description	Reset Value
dcregOverlayROIOrigin0	DISP_BA+0x1D00	R/W	Overlay Region of Interest Origin Register. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
Y							
23	22	21	20	19	18	17	16
Y							
15	14	13	12	11	10	9	8
X							
7	6	5	4	3	2	1	0
X							

Bits	Description	
[31:16]	**Y**	OSD/Rectangle start point Y coordinate.
[15:0]	**X**	OSD/Rectangle start point X coordinate.

DcregOverlayROISize Register (DcregOverlayROISize0)

Register	Offset	R/W	Description	Reset Value
dcregOverlayROISize0	DISP_BA+0x1D40	R/W	Overlay Region of Interest Size Register. **Note:** This register is double buffered.	0x00000000

31	30	29	28	27	26	25	24
HEIGHT							
23	22	21	20	19	18	17	16
HEIGHT							
15	14	13	12	11	10	9	8
WIDTH							
7	6	5	4	3	2	1	0
WIDTH							

Bits	Description	
[31:16]	**HEIGHT**	OSD/Rectangle height.
[15:0]	**WIDTH**	OSD/Rectangle width.

6.41 2D Graphic Engine (GFX)

6.41.1 Overview

Vivante GC520L composition processing core (CPC) IP defines a high-performance multi-pipe 2D raster graphics core that accelerates the 2D graphics display on a variety of consumer devices and provides advanced compression capabilities. Addressable screen sizes range from the smallest cell phones up to 1080p displays.

This chapter contains copyright confidential and proprietary material disclosed with permission of Vivante Corporation. Vivante has authorized re-distribution of this material restricted to those NDA partners and licensees of Nuvoton who are engineering MPU product which includes the discussed Vivante IP. All rights reserved.

6.41.2 Features

- 2D
  - Bit Blit.
  - Stretch Blit.
  - Rectangle fill and clear.
  - Line drawing.
  - Filter Blit.
  - Mono expansion for text rendering.
  - ROP2, ROP3 and ROP4.
  - Alpha blending, including Java 2 Porter-Duff compositing blending rules.
  - 32K x 32K coordinate system.
  - 90 / 180 / 270 degree rotation.
  - Transparency by monochrome mask, chroma key, or pattern mask.
  - Supports 2x2 in 4x4 tile format.
  - Supports XMajor and YMajor Super Tile 64x64 format.
  - Supports Fast Clear when reading Super Tiled source.
  - A8 output with rotation in filter blit and bit blit.
  - Supports Src/Dest color key full bypass.
- Multi Source Blending
  - Supports Multi source blending with variable block size to improve BW and reduce SW overhead.
  - Supports up to 8 sources.
  - Programmable block size guarantees cache efficiency so each source is read once and each destination is written once.
  - Supports 90, 180, 270 degree rotation with different block size for higher cache efficiency.
- YUV Support
  - Supports Full Multi destination for converting non-planar YUV formats to planar YUV. Used in extracting various components from the input color into different destination planes.

- Supports YUV422 output with alpha blending.

- Clock Disabling
  - The core clock enters the frequency scaling block, and a clock with missing pulses is output, buffered, and sent to clock gating logic. The clock gate output will be used by the 2D GFX blocks. The core clock can be shut down through software by setting the proper register bits.
- AXI Bus
  - AXI 4-bit ID, such that all Read and Write transactions have a unique AXI ID.
  - Supports AXI out of order read return.
- Additional Enhancements
  - Supports Full functional MMU with variable page size.

6.41.3 Block Diagram

The main functional blocks of the GC520L are described here. A block diagram of the complete graphics pipeline is given in Figure 6.41-1.

Host Interface

Allows the GC520L core to communicate with external memory and the CPU through the AXI or the AHB bus. In this block, data crosses clock domain boundaries.

Memory Controller

Internal memory unit which serves as the block-to-host interface for memory requests.

Graphics Pipeline Front End

Inserts high level primitives and commands into the graphics pipeline.

2D Drawing and Scaling Engine

Draws 2D graphics primitives and rasterizes 2D images.

Pixel Engine

Pixel manipulation and filtering on rendered images. GC520L has four (4) pixel pipes to improve performance and an one-pass filter(OPF) for filter blit.

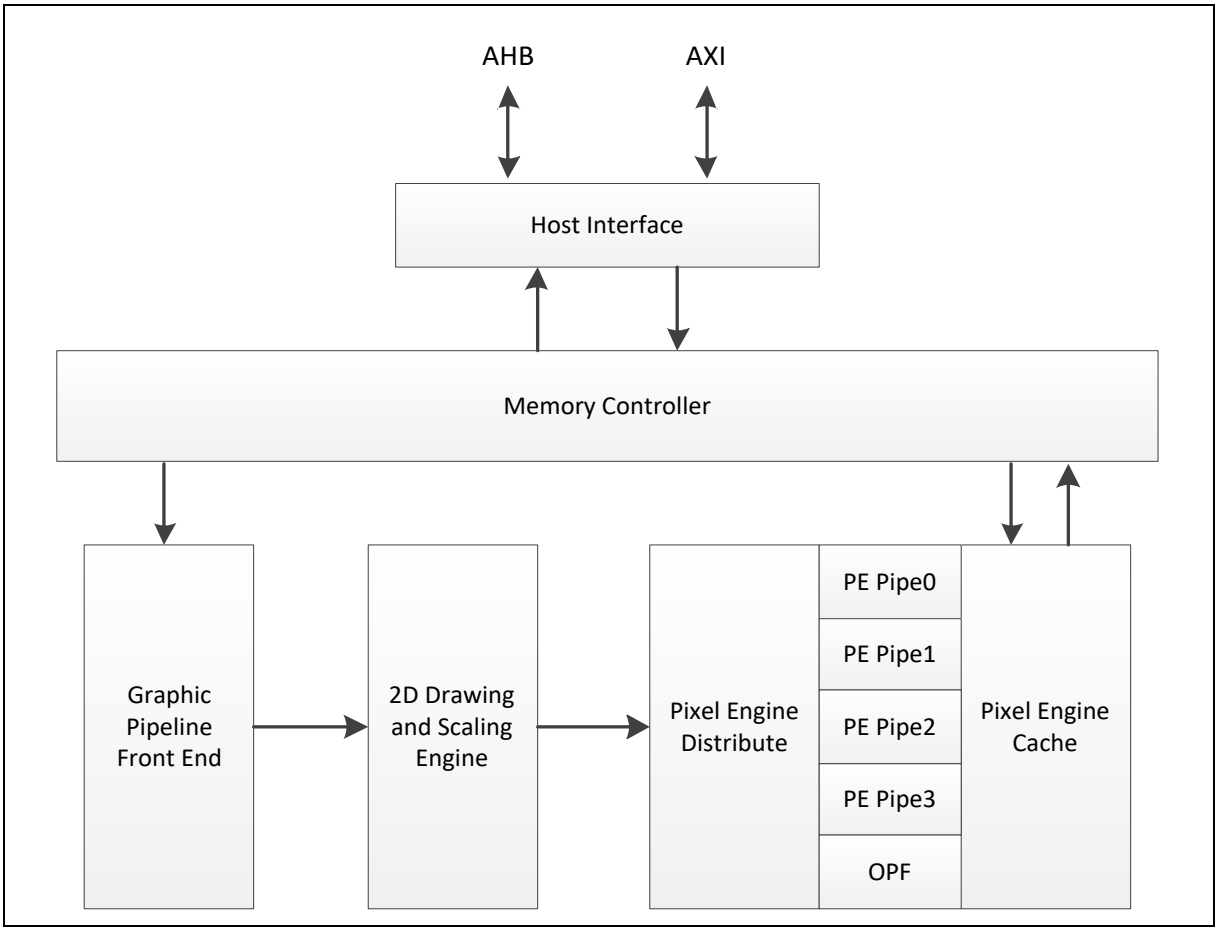


Figure 6.41-1 GC520L Block Diagram

6.41.4 Basic Configuration

- Clock Source Configuration
  - Select GFX engine clock source in GFXSEL (CLK_CLKSEL0[26])
  - Enable GFX engine clock in GFXEN (CLK_SYSCLK0[24])
- DDR AXI Configuration
  - Enable DDR AXI port1 in port_en (PCTRL_1[0])
- Reset Configuration
  - Reset GFX engine in GFXRST (SYS_IPRST0[12])

To control 2D graphic engine, Vivante GC520L, the 2D API provided by Verisilicon must be used.

Initial Flow

1. Configure GFXSEL(CLK_CLKSEL0[26])
2. Set GFXEN(CLK_SYSCLK0[24]) to 1 and set port_en to 1
3. Set GFXRST(SYS_IPRST0[12]) to 1 and wait more than 32 slowest clock cycles
4. Set GFXRST(SYS_IPRST0[12]) to 0 and wait more than 128 slowest clock cycles

6.41.5 Functional Description

6.41.5.1 Line

The LINE operation draws a line. Coordinates for two points are given: start point and end point. The end point is not drawn.

Lines are rendered using the Bresenham algorithm. The Bresenham algorithm has the advantage of using integer arithmetic and has no accumulation of rounding errors.

In the case of line, only ROP2 and ROP4 are supported. It operates on pattern and destination. The pattern should have a transparency mask in order to use ROP4.

Clipping is supported for lines on a per pixel basis.

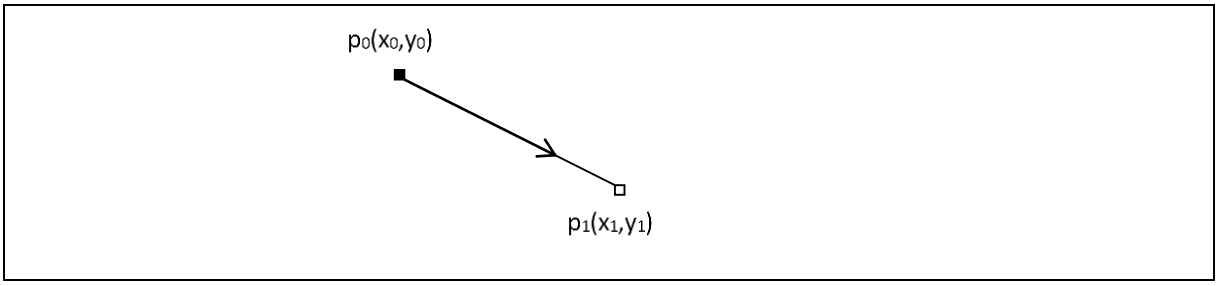


Figure 6.41-2 Line

6.41.5.2 Rectangle Fill and Clear

Rectangle fill suffuses a rectangle area with a given color. Essentially rectangle fill is a pattern fill, where an 8x8 pattern is initialized with the specified color. It supports ROP2 and ROP4 with the pattern and destination as its inputs. If ROP4 is used, the pattern should have a transparency mask.

Clear is similar to rectangle fill except that it does not use a pattern. A 32-bit clear value with 4-bit byte mask is used to fill the entire rectangle area.

Both rectangle fill and clear support clipping, which is performed on a per primitive basis.

6.41.5.3 Bit Blit

Bit blit transfers data from one area of a memory (source) to another area of the memory (destination). The source and destination can be from the same or from different memory locations. Both source and destination must be described by a rectangular area. The source and destination rectangles can be the same size (most bit blits are of this nature) or they can be different sizes, in which case the operation becomes a stretch or shrink blit.

Bit blit supports ROP2, ROP3, and ROP4 which includes source, destination, pattern, and an optional transparency color.

Clipping can be performed on a primitive basis.

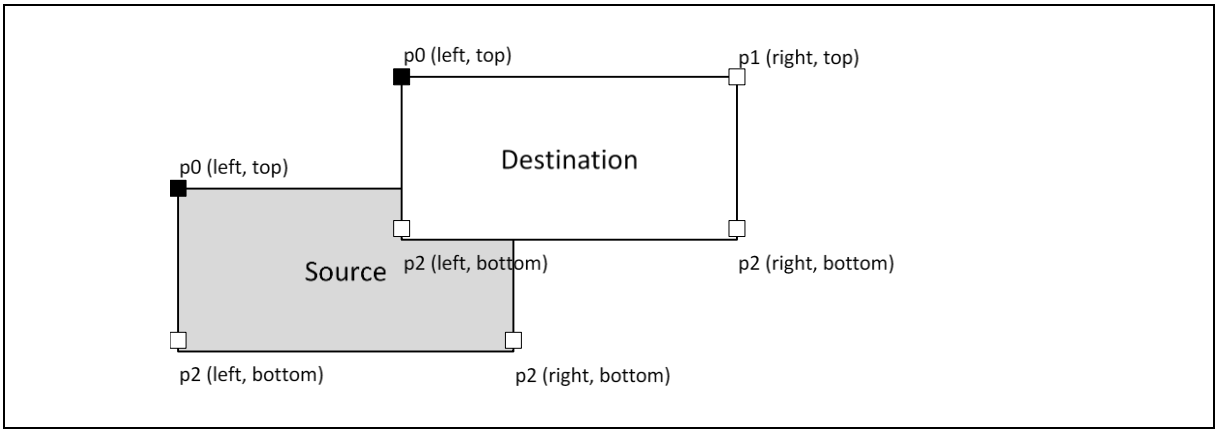


Figure 6.41-3 Bit Blit

6.41.5.4 Stretch Blit

The stretch blit primitive performs a bit blit operation with stretch or shrink. The modified Bresenham algorithm is used to generate corresponding coordinates for fast stretching. The stretch factor is specified in a 15.0 fixed-point format. Stretch blit is not allowed to overlap, that is, no part of source and destination can share any piece of memory. Non-stretch blits can overlap. For stretch blit clipping is performed on a per pixel basis.

6.41.5.5 Monochrome Expansion and Mask Blit

Monochrome expansion and mask blit are different operations, although both use the bit stream from the command buffer. And both can be the source for ROP4 source selection. This means that each output pixel can be a combination of source, pattern, monochrome mask (for masked blits) and destination.

Monochrome expansion

For monochrome expansion, the bit from the stream is used to switch on/off a solid color that is defined in a register. This mechanism enables the use of just one bit per pixel to represent colors. In effect, the MONO EXPANSION primitive increases color representation from one bit per pixel to multiple bits per pixel. A typical application for mono color is font drawing.

Monochrome expansion does not support overlapping of the source and destination. It is the responsibility of the driver to make sure that the command will never be executed on overlapping source and destination.

Mask Blit

For mask blit, the bit from the stream is used to toggle on/off a color in the source frame buffer. Mask blit takes its color source from memory and its monochrome mask from the command stream. Clipping is supported and is performed on a per pixel basis.

6.41.5.6 Filter Blit

Filter blit performs high quality scaling, up or down, using a FIR re-sampling filter with up to 9 taps. Sub-pixel coordinates (locations between the pixel grids) are generated by the drawing engine. The filter block in the drawing engine uses the sub-pixel information to select the appropriate filter kernel. GC520L processes 1 pixel every cycle when performing filter blit.

A stretch- or shrink-factor of 15.16 fixed-point format is supported. To generate a single destination pixel requires 9 source pixels. An image is scaled in two passes, one for X-dimension (HOR_FILTER_BLT) and the other for Y-dimension (VER_FILTER_BLT). Software sets up the filter kernel/coefficient table and the kernel size, as well as a temporary buffer for storing intermediate results. After the first pass is completed, intermediate results are sent back to memory, and then the second pass starts to scale the

first-pass image. Because of this two-step procedure, the throughput for filter blits is lower than that for stretch blits. Also the Filter Kernel Table may need to be reloaded, and some cycles are consumed in calculating the stepping parameters.

When the stretch or shrink factor is 1, the filter blit works as a bit blit copy. It can be used as format converter in that case, for instance, YUV to RGB converter. To use as a format converter, only one pass (HOR_FILTER_BLT or VER_FILTER_BLT) is needed. To optimize the memory bandwidth, when using filterBlit to do YUV to RGB filtering, the temporary target buffer format can be specified as YUY2 to process Y-dimension filtering (VER_FILTER_BLT). This is to avoid converting YUV to A8R8G8B8 in the first vertical pass to reduce the memory bandwidth and increase the pixel processing rate. This is the only special case where the GFX may use YUY2 as target format.

When the stretch or shrink factor (scale ratio) is not 1:1, filter blit requires both a vertical pass as well as a horizontal pass to do the scaling. Shrink performance will be less than 1 pixel per cycle for each vertical pass and for each horizontal pass. Stretch performance will be near to the performance for the 1:1 scale ratio.

6.41.5.7 Other Features

Rotation

- 90° / 180° / 270° / Mirror rotation is supported for all primitives.
- Independent source rotation

Transparency Mode

For monochrome expansion:

- Opaque.
- Conditional transparency. Transparent if the current pixel matches the specified value.

For blits:

- Opaque.
- Masked transparency. Transparent if the mask for the current pixel or pattern is zero.
- Source Conditional transparency. Transparent if the source pixel is within the specified value range.
- Destination Conditional transparency. Transparent if the destination pixel is not within the specified value range.

Clipping

- One clipping rectangle is supported for all bit blit primitives.

6.41.5.8 Supported Formats

The graphics engine supports the following source and destination formats for data, bit blits and filter blits. In addition to these source and destination RGB formats, their swizzle formats (ARGB, RGBA, ABGR, BGRA) are also supported. For YUV formats, the GFX supports their U/V swap formats.

Format	Bit Blit Input	Bit Blit Dest	Stretch Blit Input	Stretch Blit Dest	Filter Blit Input	Filter Blit Dest	OPF Input	OPF Dest	Multi-Source Input	Multi-Source Dest
A1R5G5B5	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
A4R4G4B4	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
A8R8G8B8	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
X1R5G5B5	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

X4R4G4B4	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
X8R8G8B8	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
R5G6B5	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
A8	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
YUY2 (packed YUV422)	Y	Y	Y	N	Y	Yv	Y	Yv	Y	Y
UYVY (packed YUV422)	Y	Y	Y	N	Y	N	Y	Y	Y	Y
YV12 (planar YUV420)	Y	MD	Y	N	Y	N	Y	N	Y	N
NV12 (semi-planar YUV420)	Y	MD	Y	N	Y	N	Y	N	Y	N
NV16 (semi-planar YUV422)	Y	MD	Y	N	Y	N	Y	N	Y	N
8-bit color index	N	N	Y	N	Y	N	N	N	Y	N
1-bit monochrome	Y	N	N	N	N	N	N	N	N	N

Table 6.41-1 Supported Data Formats

Note: Y = Yes, N = Not supported, MD = Multi-Destination support, Yv = Only possible when vertical filter blit is done on YUV input.

6.41.5.9 YUV Output for 2D Operations

YUV output is available for packed or planar YUV input formats in Table 6.41-2.

2D Operation	Destination
Bit blit	Packed YUV
One Pass Filter	Packed YUV
Multi-source	Packed YUV (no scaling)
Multi-destination	Planar YUV
Stretch blit	NO
Two Pass Filter	NO

Table 6.41-2 YUV Output Support for Packed or Planar YUV Source per 2D Operation

6.41.5.10 ARGB Data Conversion

The pixels read from source or destination will be expanded into A8R8G8B8 format to maintain lossless pixel operations. The resulting pixels will be converted into the destination format.

6.41.5.11 YUV to RGB Conversion

YUV data can be converted into 8-bit per component RGB format at the output of the cache only. Once converted, there is no way back to YUV format. GFX supports BT.601 and BT.709 YUV to RGB color conversion standards.

In BT.601, the YUV to RGB conversion is done using the following approximation:

$16 \le Y \le 235$

$16 \le U \le 240$

$16 \le V \le 240$

A = Y – 16

B = U – 128

C = V – 128

R = clip((298*A + 410*C + 128) >> 8)

G = clip((298*A – 101*B – 209*C + 128) >> 8)

B = clip((298*A + 519*B + 128) >> 8)

The Y, U and V components are clamped prior to the conversion.

Y is clamped between 16 and 235, inclusively.

U and V are clamped between 16 and 240, inclusively.

In BT.709, the R, G, B equations are slightly changed to

R = clip((298*A + 461*C + 128) >> 8)

G = clip((298*A – 55*B – 137*C + 128) >> 8)

B = clip((298*A + 543*B + 128) >> 8)

6.41.5.12 Color Index Input Conversion Support

Color index is supported for source data only. A look-up table with 256 entries is provided for indexing the data. The table is fully programmable. The conversion is done when pixels are read out of the cache.

6.41.5.13 Source/Destination Pre-multiply Support

The GFX supports source pre-multiply source alpha or global alpha, or source global color for global colorizing.

On destination, the GFX supports destination pre-multiply and destination alpha.

On destination, the GFX supports destination pre-multiply destination alpha, destination de-multiply alpha.

Microsoft Alpha Blending	**Equation**	
S:Fix Alpha	D.C=A*S.C+(1-A)*D.C D.A=A*S.A+(1-A)*D.A	**D:Destination** **S:Source** **A: Alpha** **C: Color**
S:Per-Pixel	D.C=S.C+(1-S.A)*D.C D.A=S.A+(1-S.A)*D.A	
S:Fix+Per-Pixel	D.C=A*S.C+(1-S.A)*D.C D.A=A*S.A+(1-S.A)*D.A	
S:Per-Pixel	D.C=S.A*S.C+(1-S.A)*D.C D.A=S.A+(1-S.A)*D.A	
S:Fix+Per-Pixel	D.C=A*S.A*S.C+(1-S.A)*D.C D.A=A*S.A+(1-S.A)*D.A	

Table 6.41-3 Pre-multiplied Modes

6.41.5.14 Alpha Blending

The GPX supports alpha blending together with ROP. Alpha blending function is performed on ROP function result source.

The general alpha blending equations are:

Cd = Fs * Cs' + Fd * Cd' (1)

Ad = Fs * As'' + Fd * Ad'' (2)

Where

Cs' is the source color component (adjusted for NPM if necessary).

Cd' is the destination color component (adjusted for NPM if necessary).

As'' is the modified source alpha component.

Ad'' is the modified destination alpha component.

Fs is fraction of the source that contributes to the final value.

Fd is fraction of the destination that contributes to the final value.

The blending is done in 4 logical stages (not real implementation stages):

1. Transparent/Opaque Conversion

   In this stage, the incoming alpha (source or destination independently) can be inverted if needed to match the internal alpha rule. Internally, an alpha of 0 means transparent, while an alpha of "0xFF" means opaque. External content might follow the opposite rule. The output of the block is either As (Ad for destination) or 1-As (1-Ad for destination).

2. Global Value Substitution

   A global alpha value from a register can be used to substitute or scaled the incoming alpha. An incoming alpha As can pass-through, be directly substituted by Ags (global alpha) or scaled by the global alpha value (As * Ags). The source and destination have distinct global alpha values.

3. Blending Factor Generation

   In this stages, the blending factors are generated (refer to table below). Each alpha can take the values 0, 1, A or 1-A depending on the blending mode.

4. Final Blending

   This is the final stage which implements the operations described by equations (1) and (2).

The different stages are illustrated in Figure 6.41-4.

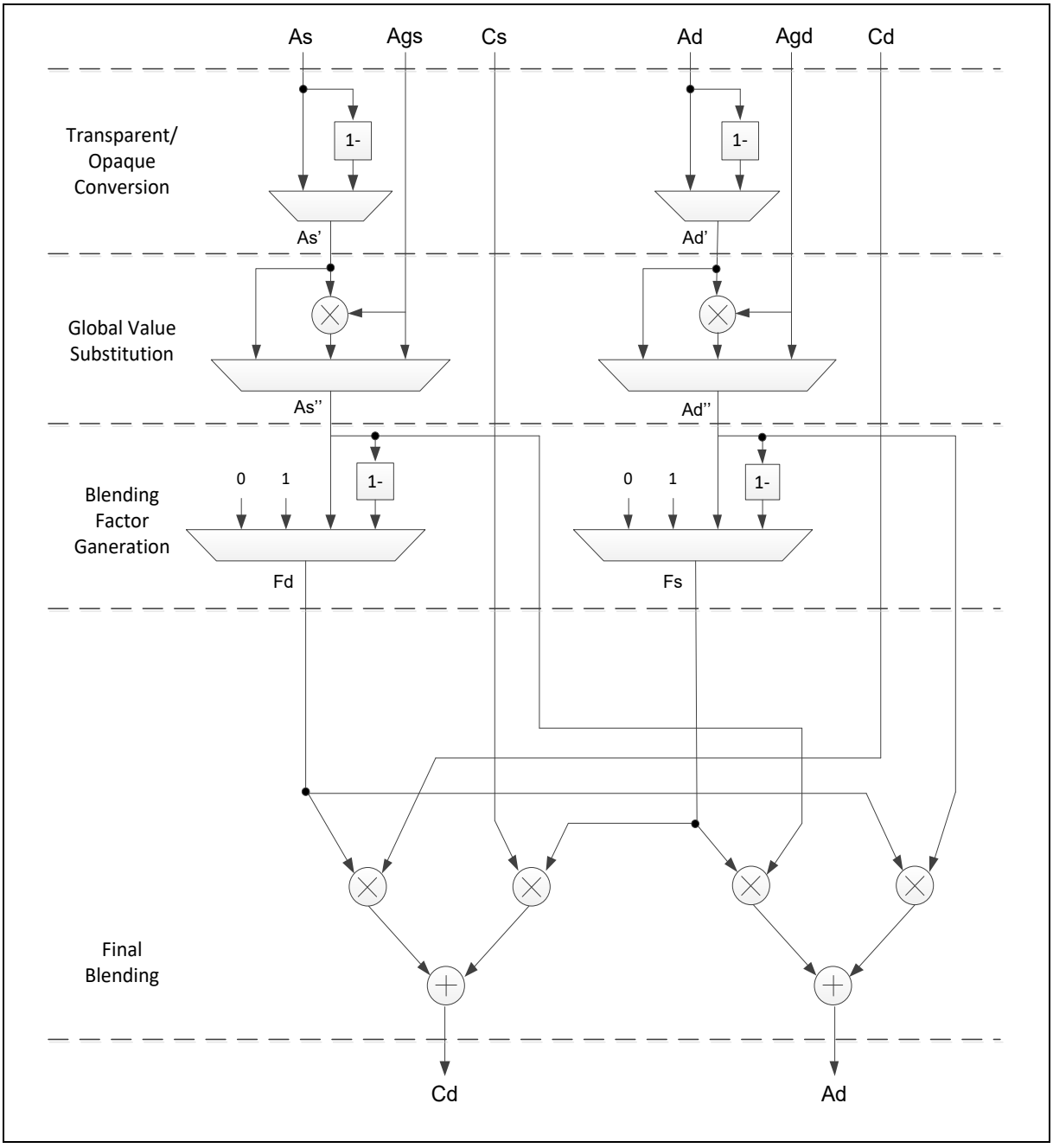


Figure 6.41-4 Alpha Blender for One Pixel

The fractions take the values described in Table 6.41-4, depending on the blending mode.

Blending Mode	Fs	Fd
Clear	0	0
SRC	1	0
DST	0	1
SRC_OVER	1	1 – As''
DST_OVER	1 – Ad''	1

SRC_IN	Ad''	0
DST_IN	0	As''
SRC_OUT	1 – Ad''	0
DST_OUT	0	1 – As''
SRC_ATOP	Ad''	1 – As''
DST_ATOP	1 – Ad''	As''
XOR	1 – Ad''	1 – As''

Table 6.41-4 Blending Modes Fractions Description

To control the blending modes, we have the following register fields:

1. 1 bit for transparent/opaque conversion for source alpha.
2. 1 bit for transparent/opaque conversion for destination alpha.
3. 2 bits for source alpha modifications, to specify the 3 cases (As, Ags, As*Ags).
4. 2 bits for destination alpha modifications, to specify the 3 cases (Ad, Agd, Ad*Agd).
5. 4 bits to select between the 12 blending modes.
6. 8-bits for global source alpha.
7. 8-bits for global destination alpha.

Alpha blending is supported on bit blit and filter blit primitives.

6.41.5.15 Dither

Color intensity at any point of an image can be thought of as a real number between 0.0 and 1.0 at each location. However, in reality only a discrete number of intensity levels can be represented. This discretization leads to a total error in the color intensities of an image equal to the square root of the sum of the squares from each point.

This error results in contouring. In an image where intensities change slowly, this will cause noticeable jumps. Dithering can be used to diffuse the intensity across neighboring pixels.

In dithering mode pixels are scanned in order, and errors in calculating a pixel's intensity are distributed (i.e., diffused) to neighboring pixels to keep the overall intensity of the image closer to the input intensity.

Dithering is supported for 16-bit color destination formats. The dither table is programmable by software.

6.41.5.16 Multi-Source Blending

Up to 8 sources are supported. Programmable block size guarantees cache efficiency so each source is read once and each destination is written once. It supports 90, 180, 270 degree destination and source rotation and mirror with different block size to have high cache efficiency. It also supports ROP2, ROP3, and ROP4 which includes source, destination and pattern, and an optional transparency color. Alpha blending between every source is supported.

6.41.5.17 GFX Cache Management

SW cache flush is supported to flush the GFX cache to memory. Auto-flush of GFX cache is also supported. SW sets up the auto-flush interval, and HW will do the cache flush automatically at programmable intervals.

6.41.5.18 Fast Clear

Fast Clear saves bandwidth when the source format is a Super Tile mode with a tile size of 256 Bytes.

Only bit blit and multi-source operations support Fast Clear.

6.41.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
GFX Base Address: **GFX_BA = 0x4028_0000**				
AQHiClockControl	GFX_BA+0x000	R/W	Clock control register	0x0000_0100
AQHiIdle	GFX_BA+0x004	R	Idle status register	0x0000_0007
AQAxiConfig	GFX_BA+0x008	R/W	AXI configuration register	0x0022_2200
AQAxiStatus	GFX_BA+0x00C	R	AXI status register	0x0000_0000
AQIntrAcknowledge	GFX_BA+0x010	R	Interrupt acknowledge register	0x0000_0000
AQIntrEnbl	GFX_BA+0x014	R/W	Interrupt enable register	0x0000_0000
GCChipRev	GFX_BA+0x024	R	Revision register	0x0000_5341
GCChipDate	GFX_BA+0x028	R	Release date register	0x0000_0000
gcTotalCycles	GFX_BA+0x078	R/W	Total cycles register	0x0000_0000
gcregHIChipPatchRev	GFX_BA+0x098	R	Patch revision level register	0x0000_0000
gcProductId	GFX_BA+0x0A8	R	Product ID register	0x0000_5202
gcModulePowerControls	GFX_BA+0x100	R/W	Control register for module level power	0x0000_0000
gcregMMUControl	GFX_BA+0x18C	W	MMU Control register	0x0000_0000
AQMemoryDebug	GFX_BA+0x414	R/W	Memory debug register	0x0000_0000
AQRegisterTImingControl	GFX_BA+0x42C	R/W	SRAM timing control register	0x0003_0000
AQCmdBufferAddr	GFX_BA+0x654	W	Command buffer base address register	0x0000_0000
AQCmdBufferCtrl	GFX_BA+0x658	R/W	Command buffer control register	0x0000_0000
AQFEDebugCurCmdAdr	GFX_BA+0x664	R	Command decoder address register	0x0000_0000

6.41.7 Register Description

AQHiClockControl Register (AQHiClockControl)

Register	Offset	R/W	Description	Reset Value
AQHiClockControl	GFX_BA+0x000	R/W	Clock control register	0x0000_0100

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				ISOLATE_GPU	Reserved	IDLE2_D	Reserved
15	14	13	12	11	10	9	8
Reserved			SOFT_RESET	Reserved	DISABLE_RAM_CLOCK_GA	FSCALE_CMD_LOAD	FSCALE_VAL
7	6	5	4	3	2	1	0
FSCALE_VAL						CLK2D_DIS	CLK3D_DIS

Bits	Description	
[31:20]	**Reserved**	Reserved.
[19]	**ISOLATE_GPU**	**Isolate GPU bit** Used for power on/off sequence.
[18]	**Reserved**	Reserved.
[17]	**IDLE2_D**	2D pipe is idle.
[16:13]	**Reserved**	Reserved.
[12]	**SOFT_RESET**	Soft resets the GFX.
[11]	**Reserved**	Reserved.
[10]	**DISABLE_RAM_CLOCK_GATING**	Disables clock gating for rams.
[9]	**FSCALE_CMD_LOAD**	**Core clock frequency scale value** When writing a 1 to this bit, it updates the frequency scale factor with the value FSCALE_VAL. The bit must be set back to 0 after that. If this bit is set and FSCALE_VAL=0 (an invalid combination), the HREADYOUT output signal will get stuck to 0.
[8:2]	**FSCALE_VAL**	**Core clock frequency scale value** If this value is set to 1, the core clock will be 1/64 of clk1x, otherwise clock is fully frequency
[1]	**CLK2D_DIS**	**Disable 2D clock** Software clock disable signal. For this core both bits CLK3D_DIS and CLK2D_DIS should be controlled by software. The AXI interface clock is the only block not stalled at that point.
[0]	**CLK3D_DIS**	**Disable 3D clock**

		Software core clock disable signal for core (clk1x) clock. When set to 1, the clock is frozen.

AQHildle Register (AQHildle)

Register	Offset	R/W	Description	Reset Value
AQHildle	GFX_BA+0x004	R	Idle status register	0x0000_0007

31	30	29	28	27	26	25	24
AXI_LP	Reserved						
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					IDLE_PE	IDLE_DE	IDLE_FE

Bits	Description	
[31]	**AXI_LP**	AXI is in low power mode.
[30:3]	**Reserved**	Reserved.
[2]	**IDLE_PE**	PE is idle.
[1]	**IDLE_DE**	DE is idle.
[0]	**IDLE_FE**	FE is idle.

AQAxiConfig Register (AQAxiConfig)

Register	Offset	R/W	Description	Reset Value
AQAxiConfig	GFX_BA+0x008	R/W	AXI configuration register	0x0022_2200

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
AXCACHE_OVERRIDE_SHARED				**AXDOMAIN_NON_SHARED**		**AXDOMAIN_SHARED**	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ARCACHE				**AWCACHE**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:20]	**AXCACHE_OVERRIDE_SHARED**	Configure AxCACHE value for shareable request.
[19:18]	**AXDOMAIN_NON_SHARED**	Configure AxDOMAIN value for non-shareable request.
[17:16]	**AXDOMAIN_SHARED**	Configure AxDOMAIN value for shareable request.
[15:12]	**ARCACHE**	Set ARCACHE[3:0] value
[11:8]	**AWCACHE**	Set AWCACHE[3:0] value
[7:0]	**Reserved**	Reserved.

AQAxiStatus Register (AQAxiStatus)

Register	Offset	R/W	Description	Reset Value
AQAxiStatus	GFX_BA+0x00C	R	AXI status register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						DET_RD_ERR	DET_WR_ERR
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RD_ERR_ID				WR_ERR_ID			

Bits	Description	
[31:10]	**Reserved**	Reserved.
[9]	**DET_RD_ERR**	1 = Detect read error
[8]	**DET_WR_ERR**	1 = Detect write error
[7:4]	**RD_ERR_ID**	Read error ID
[3:0]	**WR_ERR_ID**	Write error ID

AQIntrAcknowledge Register (AQIntrAcknowledge)

Register	Offset	R/W	Description	Reset Value
AQIntrAcknowledge	GFX_BA+0x010	R	Interrupt acknowledge register	0x0000_0000

31	30	29	28	27	26	25	24
INTR_VEC							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
INTR_VEC							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
INTR_VEC							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INTR_VEC							

Bits	Description	
[31:0]	**INTR_VEC**	**Interrupt Vector** Each bit represents a corresponding event being triggered. Reading from this register clears the outstanding interrupt. For each interrupt event, 0=Clear, 1=Interrupt Active. INTR_VEC[31] is AXI_BUS_ERROR, 0 = No Error.

AQIntrEnbl Register (AQIntrEnbl)

Register	Offset	R/W	Description	Reset Value
AQIntrEnbl	GFX_BA+0x014	R/W	Interrupt enable register	0x0000_0000

31	30	29	28	27	26	25	24
INTR_ENBL_VEC							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
INTR_ENBL_VEC							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
INTR_ENBL_VEC							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INTR_ENBL_VEC							

Bits	Description	
[31:0]	**INTR_ENBL_VEC**	**Interrupt enable** Each bit enables a corresponding event.

GCChipRev Register (GCChipRev)

Register	Offset	R/W	Description	Reset Value
GCChipRev	GFX_BA+0x024	R	Revision register	0x0000_5341

31	30	29	28	27	26	25	24
REV							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
REV							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
REV							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
REV							

Bits	Description	
[31:0]	**REV**	Revision

GCChipDate Register (GCChipDate)

Register	Offset	R/W	Description	Reset Value
GCChipDate	GFX_BA+0x028	R	Release date register	0x0000_0000

31	30	29	28	27	26	25	24
DATE							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DATE							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DATE							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATE							

Bits	Description	
[31:0]	**DATE**	Date (YYYY/MM/DD)

GcTotalCycles Register (GcTotalCycles)

Register	Offset	R/W	Description	Reset Value
gcTotalCycles	GFX_BA+0x078	R/W	Total cycles register	0x0000_0000

31	30	29	28	27	26	25	24
CYCLES							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CYCLES							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CYCLES							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CYCLES							

Bits	Description	
[31:0]	**CYCLES**	**Total cycles** This register is a free running counter. It can be reset by writing 0 to it.

GcregHIChipPatchRev Register (GcregHIChipPatchRev)

Register	Offset	R/W	Description	Reset Value
gcregHIChipPatchRev	GFX_BA+0x098	R	Patch revision level register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PATCH_REV							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**PATCH_REV**	Patch revision level for the chip.

GcProductId Register (GcProductId)

Register	Offset	R/W	Description	Reset Value
gcProductId	GFX_BA+0x0A8	R	Product ID register	0x0000_5202

31	30	29	28	27	26	25	24
Reserved				TYPE			
23	22	21	20	19	18	17	16
NUM							
15	14	13	12	11	10	9	8
NUM							
7	6	5	4	3	2	1	0
NUM				GRADE_LEVEL			

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:24]	**TYPE**	**Core Type** 0 = GC (2D or 3D Graphics Cores).
[23:4]	**NUM**	**Product Number** 520 for this core.
[3:0]	**GRADE_LEVEL**	**Grade Level** 0 = None-no extra letter on the product name;. 2 = L-Lite. 3 = UL-Ultra Lite.

GcModulePowerControls Register (GcModulePowerControls)

Register	Offset	R/W	Description	Reset Value
gcModulePowerControls	GFX_BA+0x100	R/W	Control register for module level power	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**DISABLE_STARVE_MODULE**	**DISABLE_STALL_MODULE_**	**ENABLE_MODULE_CLOCK_**

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**DISABLE_STARVE_MODULE_CLOCK_GATING**	Disables module level clock gating for starve/idle condition.
[1]	**DISABLE_STALL_MODULE_CLOCK_GATING**	Disables module level clock gating for stall condition.
[0]	**ENABLE_MODULE_CLOCK_GATING**	Enables module level clock gating.

GcregMMUControl Register (GcregMMUControl)

Register	Offset	R/W	Description	Reset Value
gcregMMUControl	GFX_BA+0x18C	W	MMU Control register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							ENABLE

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**ENABLE**	**Enable the MMU** For security reasons, once the MMU is enabled it cannot be disabled anymore. 1 = enable.

AQMemoryDebug Register (AQMemoryDebug)

Register	Offset	R/W	Description	Reset Value
AQMemoryDebug	GFX_BA+0x414	R/W	Memory debug register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MAX_OUTSTANDING_READS							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**MAX_OUTSTANDING_READS**	Limits the total number of outstanding read requests.

AQRegisterTImingControl Register (AQRegisterTImingControl)

Register	Offset	R/W	Description	Reset Value
AQRegisterTImingControl	GFX_BA+0x42C	R/W	SRAM timing control register	0x0003_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved	LIGHT_SLEEP	DEEP_SLEEP	POWER_DOW	FAST_WTC		FAST_RTC	
15	14	13	12	11	10	9	8
FOR_RF2P							
7	6	5	4	3	2	1	0
FOR_RF1P							

Bits	Description	
[31:23]	**Reserved**	Reserved.
[22]	**LIGHT_SLEEP**	Light sleep
[21]	**DEEP_SLEEP**	Deep sleep
[20]	**POWER_DOW**	Power down
[19:18]	**FAST_WTC**	WTC for fast RAMs
[17:16]	**FAST_RTC**	RTC for fast RAMs
[15:8]	**FOR_RF2P**	For 2 port RAM
[7:0]	**FOR_RF1P**	For 1 port RAM

AQCmdBufferAddr Register (AQCmdBufferAddr)

Register	Offset	R/W	Description	Reset Value
AQCmdBufferAddr	GFX_BA+0x654	W	Command buffer base address register	0x0000_0000

31	30	29	28	27	26	25	24
ADDRESS							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ADDRESS							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ADDRESS							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADDRESS							

Bits	Description	
[31:0]	**ADDRESS**	**Base address for the command buffer** The address must be 64-bit alignment and it is always physical. To check the value of the current fetch address use AQFEDebugCurCmdAdr register. Since this is a write only register is has no reset value.

AQCmdBufferCtrl Register (AQCmdBufferCtrl)

Register	Offset	R/W	Description	Reset Value
AQCmdBufferCtrl	GFX_BA+0x658	R/W	Command buffer control register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							ENABLE
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PREFETCH							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PREFETCH							

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**ENABLE**	**Enable** Enable the command parser.
[15:0]	**PREFETCH**	**Prefetch (write only)** Number of 64-bit words to fetch from the command buffer.

AQFEDebugCurCmdAdr Register (AQFEDebugCurCmdAdr)

Register	Offset	R/W	Description	Reset Value
AQFEDebugCurCmdAdr	GFX_BA+0x664	R	Command decoder address register	0x0000_0000

31	30	29	28	27	26	25	24
CUR_CMD_ADR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CUR_CMD_ADR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CUR_CMD_ADR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CUR_CMD_ADR				Reserved			

Bits	Description	
[31:4]	**CUR_CMD_ADR**	**This is the command decoder address** The address is always physical so the MSB should always be 0. Note that with the current MMU all 32 bits are valid. It has no reset value.
[3:0]	**Reserved**	Reserved.

6.42 H.264/JPEG Decoder (VC8000NanoD)

6.42.1 Overview

This chapter contains copyright confidential and proprietary material disclosed with permission of Vivante Corporation. Used with permission. Vivante has authorized re-distribution of this material restricted to those NDA partners and licensees of Nuvoton who are engineering MPU product which includes the discussed Vivante IP. All rights reserved.

The Hantro VC8000NanoD Multi-format Decoder is a single-core solution providing HD decoding for H.264 and JPEG. It is one of the smallest multi-format video decoder IP solutions and offers low power consumption and negligible load on the host CPU. Dynamic power consumption is minimized by extensive use of multi-level hierarchical clock gating.

An optimized software stack and a comprehensive set of software development tools are supported by a robust pipeline designed for industry-standard codecs and Hantro APIs, and with full support for Linux embedded development platforms. Software for VC8000NanoD IP supports Hantro APIs for the following industry standard codecs: AVC (H.264), MVC, SVC, and JPEG.

6.42.2 Features

- Video decoding
  - H.264 Decoder (Baseline/mainstream/High profile supported)
  - Single decoding pipeline: 1080p@30fps
  - Support picture size from 48x48 to 1920x1080
  - Support step size 16 pixels
  - Stream error detection
  - Support reference buffer
- JPEG decoding
  - JFIF file format: 1.02
  - Multiple input formats
  - YCbCr420 semi-planar raster scan output format
  - Support picture size from 48x48 to 16368x16368
  - Support step size 8 pixels
  - Stream error detection
  - JPEG compressed thumbnails supported
- Post Processing
  - Support image down-scaling
  - Support image up-scaling
  - Support YCbCr to RGB color conversion
  - Support dithering
  - Programmable alpha channel
  - Support alpha blending
  - Support deinterlacing
  - Support contrast, brightness, and color saturation adjustment for RGB image
  - Support image cropping and digital zoom

- Support picture in picture
- Support output image masking
- Support image rotation

6.42.3 Block Diagram

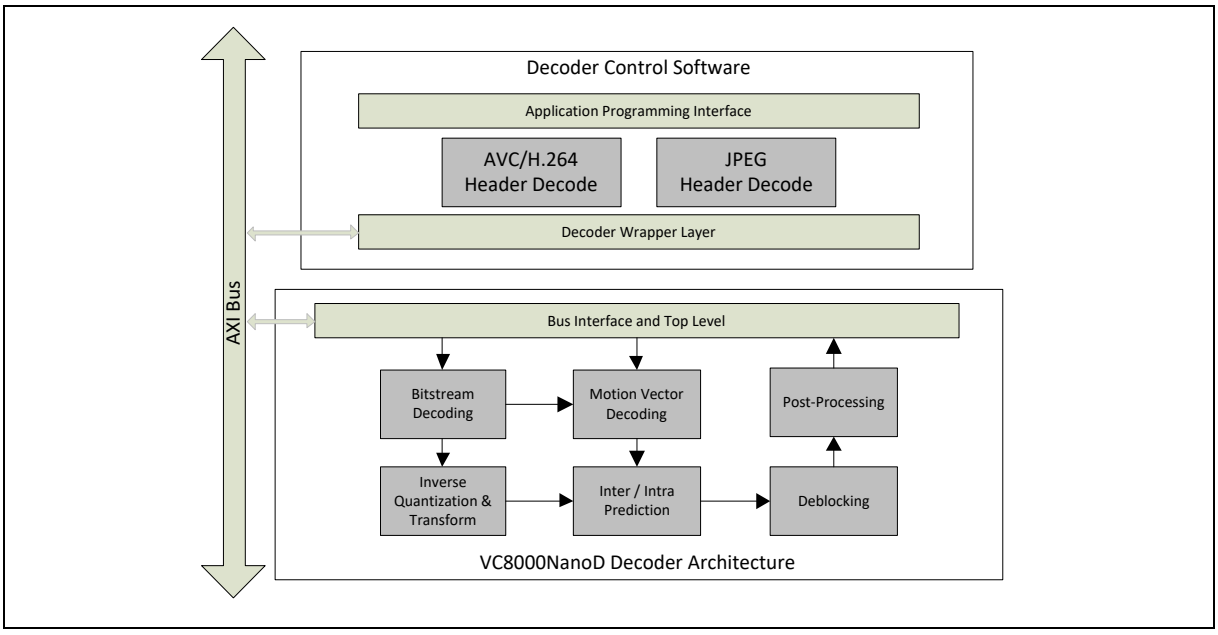


Figure 6.42-1 VC8000NanoD Decoder Diagram

6.42.4 Functional Description

The following tables list the decoding feature support available in VC8000NanoD for the designated standards.

6.42.4.1 Decoding Feature Support for AVC(H.264) / MVC / SVC

Feature	Decoder Support
Input stream format	AVC(H.264) stream – Byte stream – NAL unit stream MVC stream SVC stream
Output picture format	YCbCr 4:2:0 semi-planar raster-scan YCbCr 4:2:0 semi-planar 8x4 tiled YCbCr 4:0:0 (monochrome)
Decoding scheme	Frame by frame (or field by field) Slice by slice
Supported picture size	48 x 48 to 1920 x 1080 Step size 16 pixels
Maximum frame rate	Single decoding pipeline: 1920 x 1080 @ 30fps
Maximum bit rate	Single decoding pipeline: H.264 HP level 5.1

Error concealment	Stream error detection Continue decoding from next slice
Output picture buffer stride	Supported

Table 6.42-1 AVC(H.264) / MVC / SVC Decoding Features

6.42.4.2 Decoding Feature Support for JPEG

Feature	Decoder Support
Input stream format	JFIF file format 1.02 YCbCr 4:0:0, 4:2:0, 4:2:2, 4:4:0, 4:1:1 and 4:4:4 sampling formats
Output picture format	YCbCr420 semi-planar raster scan
Decoding scheme	Input: buffer by buffer, from 5 Kbytes to 8 Mbytes at a time Output: from 1 Mbytes row to 16 Mpixels at a time
Supported picture size	48 x 48 to 16368 x 16368 Step size 8 pixels
Maximum pixel rate	Single decoding pipeline: Up to 250M pixels per second
Error concealment	Stream error detection Continue decoding from next restart point Some error correction
Thumbnail decoding	JPEG compressed thumbnails supported

Table 6.42-2 JPEG Decoding Features

6.42.4.3 Post-Processing Features

Post-processor Feature	Support
Work scheme	Pipelined with decoder
Input data format	Any format generated by the decoder in combined mode. In standalone mode: • YCbCr 4:2:0 semi-planar raster-scan • YCbCr 4:2:0 semi-planar tiled (8x4) • YCbCr 4:2:0 planar • YCbYCr 4:2:2, YCrYCb 4:2:2 • CbYCrY 4:2:2, CrYCbY 4:2:2
Post-processing scheme	Frame by frame. Post-processor handles the image macroblock by macroblock, also in standalone mode.
Input image source	Internal source (combined mode): VC8000NanoD decoder External source (standalone mode): e.g., a software decoder or camera interface
Input image size (combined mode)	48 x 48 to 16368 x 16386 (256 Mpixels) Step size 16 pixels
Input image size (standalone mode)	Width from 48 to 8176 Height from 48 to 8176 Maximum size limited to 16.7 Mpixels 1) Step size 16 pixels

Available PP outputs	1, 2, 3, 4
Output image size	16 x 16 to 1920 x 1080 Horizontal step size 8 Vertical step size 2
Output data format	YCbCr 4:2:0 semi-planar • YCbYCr 4:2:2 raster-scan or 4x4 tiled • YCrYCb 4:2:2 raster-scan or 4x4 tiled • CbYCrY 4:2:2 raster-scan or 4x4 tiled • CrYCbY 4:2:2 raster-scan or 4x4 tiled • Fully configurable ARGB channel lengths and locations inside 32 bits, e.g. ARGB 32-bit (8-8-8-8), RGB 16-bit (5-6-5), ARGB 16-bit (4-4-4-4)
Image down-scaling	Proprietary averaging filter • Arbitrary, non-integer scaling ratio separately for both dimensions • Unlimited down-scaling ratio (e.g. from 16Mpixel to QVGA)
Image up-scaling	Bicubic polynomial interpolation with a fourtap horizontal kernel and a two-tap vertical kernel • Arbitrary, non-integer scaling ratio separately for both dimensions • Maximum output width is 3x the input width (within the maximum output image size limit) • Maximum output height is 3x the input height – 2 pixels (within the maximum output image size limit)
YCbCr to RGB color conversion	BT.601-5 compliant [15] BT.709 compliant [16] User definable conversion coefficient
Dithering	2x2 ordered spatial dithering for 4, 5 and 6 bit RGB channel precision
Programmable alpha channel	Constant eight bit value can be set to the alpha channel of the 24-bit RGB output data to control the transparency of the output picture. The resulting 32-bit ARGB data can be used as input data for later alpha blending.
Alpha blending	Output image can be alpha blended with two rectangular areas. YCbCr semi-planar 4:2:0 PP output format is not supported when performing alpha blending. The supported overlay input formats are the following: 1. alpha value + YCbCr 4:4:4, 8 bits each 2. 8 bit alpha value + 24 bit RGB Alpha blending input data can be cropped from a larger area by defining the scan line length.
Deinterlacing	Conditional spatial deinterlace filtering. Supports only YCbCr 4:2:0 input format.
RGB image contrast adjustment	Segmented linear
RGB image brightness adjustment	Linear
RGB image color saturation adjustment	Linear
Image cropping / digital zoom	User definable start position, height and width. Can be used with scaling to perform digital zoom. Usable only for JPEG or stand-alone mode.
Picture in picture	Output image can be written to any location inside video memory. Up to 1920 x 1080 sized displays supported.
Output image masking	Output image writing can be prevented on two rectangular areas in the image. The masking feature is exclusive with alpha blending; however it is possible to have one

	masking area and one blending area.
Image rotation	Rotation 90, 180 or 270 degrees Horizontal and vertical flip

Table 6.42-3 Post-Processing Features

6.42.5 Register Map

This chip uses the Application Programming Interface (API) to access the VC8000NanoD instead of directly read or write the VC8000NanoD's control and status registers.

6.42.6 Register Description

This chip uses the Application Programming Interface (API) to access the VC8000NanoD instead of directly read or write the VC8000NanoD's control and status registers.

6.43 Keypad Interface (KPI)

6.43.1 Overview

The Keypad Interface (**KPI**) is an APB slave with configurable minimum 2-row up to 8-row scan output and minimum 1-column up to 8-column scan input. Any keys in the array pressed or released are de-bounced and generate an interrupt.

The KPI supports release multiple keys, press multiple keys scan interrupt and specified INT3KEYs interrupt for chip reset. If the 3 pressed keys matches with the 3 keys defined in **KPI3KCONF**, it will generate an interrupt for chip reset depending on the **EN3KYRST (KPI3KCONF[24])** setting. The period of three key reset is 64 * KPI engine clock. The interrupt is generated whenever it detects any key in the keypad pressing or releasing or waking up from IDLE or three-key reset. User can know the interrupt source by querying KPISTATUS register.

6.43.2 Features

- Matrix keypad interface (maximum 8x8 array, and minimum 2x1array)
- Programmable de-bounce time
- Low-power wakeup mode
- Programmable three-key reset
- Generates interrupt and updates all the keys(maximum 64 keys, minimum 2 keys) information (press/release) every time the user pressing or releasing

6.43.3 Block Diagram

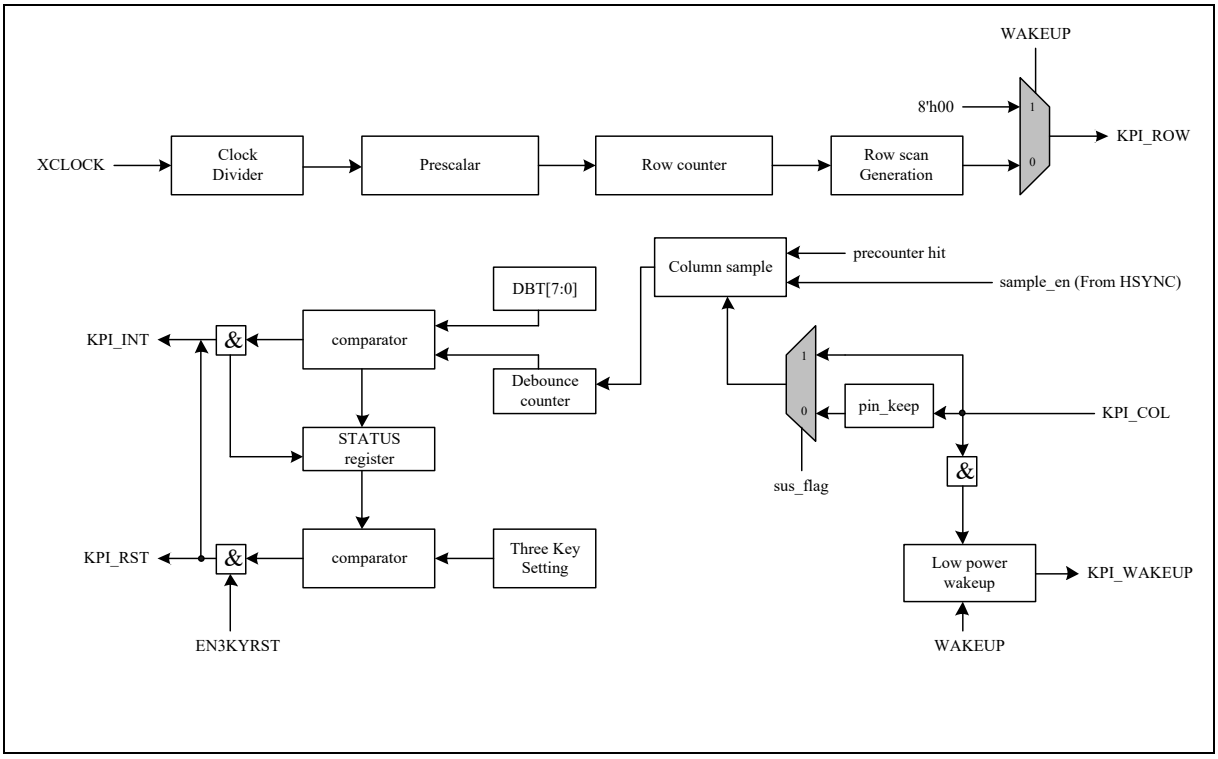


Figure 6.43-1 Keypad Controller Block Diagram

6.43.4 Basic Configuration

- Clock Source Configuration

- The KPI peripheral clock can be enabled in KPICKEN (CLK_APBCLK0[31]). The KPI engine clock source is selected by KPISEL (CLK_CLKSEL4[30]) and KPI engine clock divider is determined by KPIDIV (CLK_CLKDIV4[31:24]).

- Reset Configuration
  - The KPI reset can be enabled in KPIRST (SYS_IPRST3 [24]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
KPI	KPI_ROW0	PF.4	MFP14
	KPI_ROW1	PF.5	
	KPI_ROW2	PF.6	
	KPI_ROW3	PF.7	
	KPI_ROW4	PF.12	
	KPI_ROW5	PF.13	
	KPI_ROW6	PB.8	MFP15
	KPI_ROW7	PB.9	
	KPI_COL0	PF.0	MFP14
	KPI_COL1	PF.1	
	KPI_COL2	PF.2	
	KPI_COL3	PF.3	
	KPI_COL4	PF.8	
	KPI_COL5	PF.9	
	KPI_COL6	PF.10	
	KPI_COL7	PF.11	

6.43.5 Functional Description

Keypads are often used as a primary input device for embedded microcontrollers. The keypads actually consist of a number of switches, connected in a row/column arrangement as shown in Figure 6.43-2.

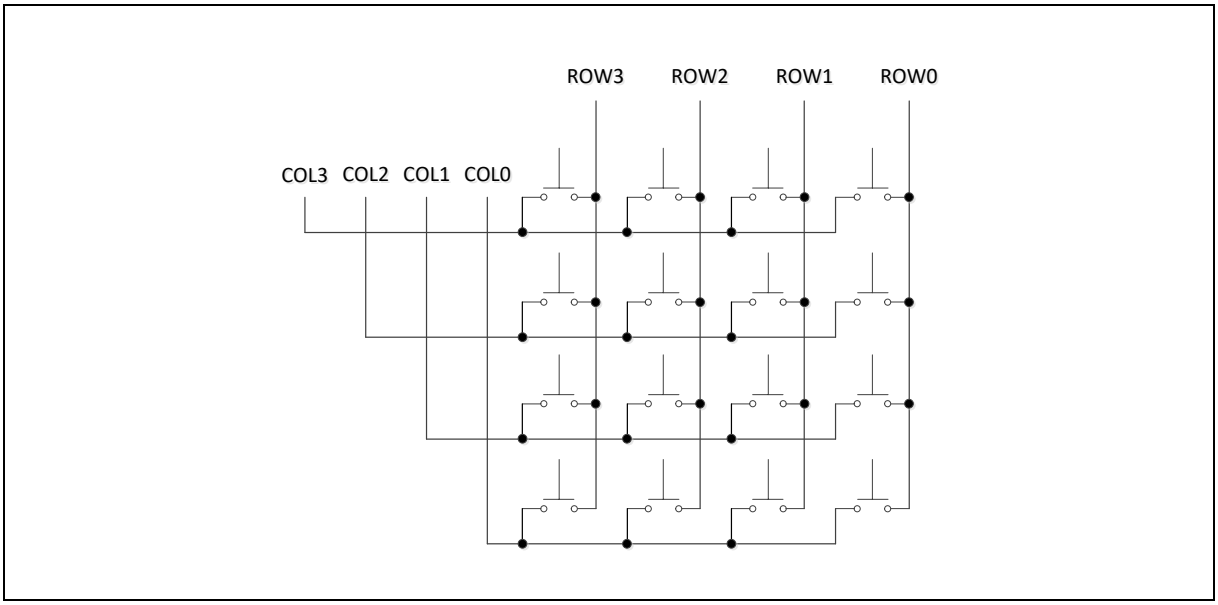


Figure 6.43-2 Keypad Connection

In order for the microcontroller to scan the keypad, it outputs a nibble to force one (only one) of the rows low and then reads the columns to see if any buttons in that column have been pressed. The columns are pulled up by the internal weak pull-up resistor.

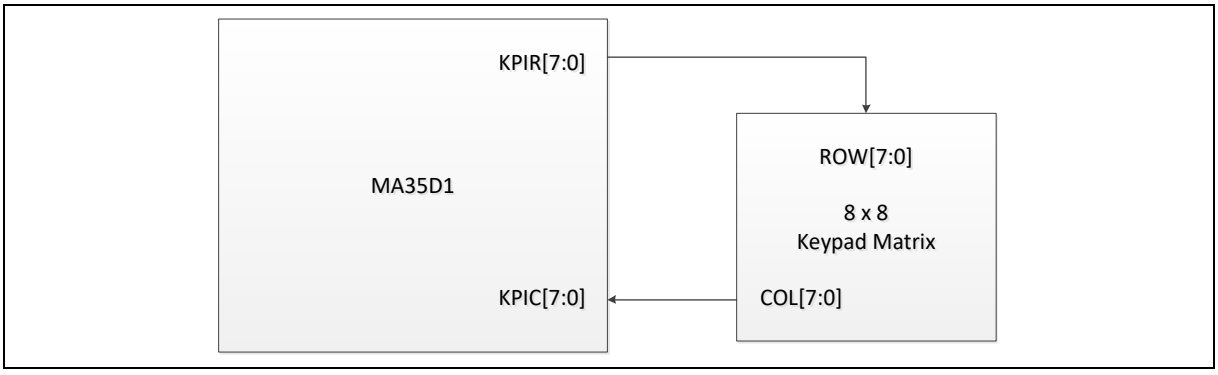


Figure 6.43-3 Keypad Interface

Consequently, as long as no buttons are pressed, the microcontroller sees a logic high on each of the pins attached to the keypad columns. The nibble driven onto the rows always contains only a single 0. The only way the microcontroller can find a 0 on any column pin is for the keypad button to be pressed that connects the row set to 0 to a row. The controller knows which column is at a 0-level and which column reads 0, allowing it to determine which key is pressed. The keypad scan timing is shown in Figure 6.43-4.

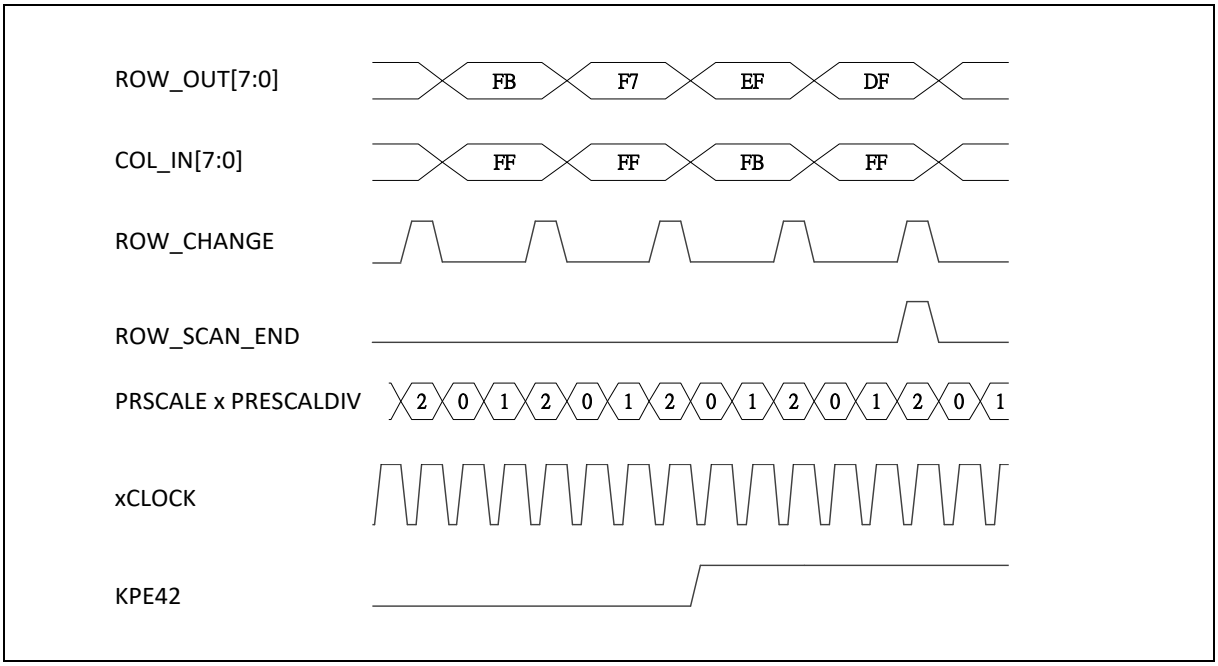


Figure 6.43-4 8 x8 Keypad Scan Timing Diagram

6.43.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
KPI Base Address: **KPI_BA = 0x404A_0000**				
KPICONF	KPI_BA+0x000	R/W	Keypad Configuration Register	0x0003_0000
KPI3KCONF	KPI_BA+0x004	R/W	Keypad 3-keys Configuration Register	0x0000_0000
KPISTATUS	KPI_BA+0x008	R/W	Keypad Status Register	0x0000_0000
KPIKEST0	KPI_BA+0x010	R	Keypad State Register 0	0xFFFF_FFFF
KPIKEST1	KPI_BA+0x014	R	Keypad State Register 1	0xFFFF_FFFF
KPIKPE0	KPI_BA+0x018	R/W	Lower 32 Key Press Event Indicator	0x0000_0000
KPIKPE1	KPI_BA+0x01C	R/W	Upper 32 Key Press Event Indicator	0x0000_0000
KPIKRE0	KPI_BA+0x020	R/W	Lower 32 Key Release Event Indicator	0x0000_0000
KPIKRE1	KPI_BA+0x024	R/W	Upper 32 Key Release Event Indicator	0x0000_0000
KPIPRESCALDIV	KPI_BA+0x028	R/W	Pre-scale Divider	0x0000_001F

6.43.7 Register Description

Keypad Configuration Register (KPICONF)

Register	Offset	R/W	Description	Reset Value
KPICONF	KPI_BA+0x000	R/W	Keypad Configuration Register	0x0003_0000

31	30	29	28	27	26	25	24
Reserved	KROW			Reserved	KCOL		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SCANROWD		Reserved		DBCLKSEL			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PRESCALE							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		WAKEUP	Reserved	INTEN	RKINTEN	PKINTEN	ENKP

Bits	Description	
[31]	**Reserved**	Reserved.
[30:28]	**KROW**	**Keypad Matrix ROW Number** The keypad matrix is set by ROW x COL. The ROW number can be set 2 to 8. 000 = reserved. 001 = 2. 010 = 3. 011 = 4. 100 = 5. 101 = 6. 110 = 7. 111 = 8.
[27]	**Reserved**	Reserved.
[26:24]	**KCOL**	**Keypad Matrix COL Number** The keypad matrix is set by ROW x COL. The COL number can be set 1 to 8. 000 = 1. 001 = 2. 010 = 3. 011 = 4. 100 = 5. 101 = 6. 110 = 7. 111 = 8.

[23:22]	**SCANROWD**	**Scan Row Delay** Setting delay cycle when row change, for preventing KPI from detecting wrong key. 00 = 4 KPI engine clock cycle. 01 = 8 KPI engine clock cycle. 10 = 16 KPI engine clock cycle. 11 = 32 KPI engine clock cycle. **Note:** scan row delay cycle < debounce sampling cycle. row scan time > scan row delay cycle + (2 * debounce sampling cycle) + 1 xclock cycle(change row) + 2 xclock cycle(cross clock domain).
[21:20]	**Reserved**	Reserved.
[19:16]	**DBCLKSEL**	**Scan in De-bounce Sampling Cycle Selection** 0000 = Reserved. 0001 = Reserved. 0010 = Reserved. 0011 = Sample interrupt input once per 8 clocks. 0100 = Sample interrupt input once per 16 clocks. 0101 = Sample interrupt input once per 32 clocks. 0110 = Sample interrupt input once per 64 clocks. 0111 = Sample interrupt input once per 128 clocks. 1000 = Sample interrupt input once per 256 clocks. 1001 = Sample interrupt input once per 512 clocks. 1010 = Sample interrupt input once per 1024 clocks. 1011 = Sample interrupt input once per 2048 clocks. 1100 = Sample interrupt input once per 4096 clocks. 1101 = Sample interrupt input once per 8192 clocks. 1110 = reserved. 1111 = reserved. **Note:** scan row delay cycle < debounce sampling cycle. row scan time > scan row delay cycle + (2 * debounce sampling cycle) + 1 xclock cycle(change row) + 2 xclock cycle(cross clock domain). row scan time = xCLOCK x PRESCALE x PrescaleDivider. For example, if debounce sampling cycle is 1024 xclock, and scan row delay cycle is 8 xclock, row scan time should be larger than (8+2048+3) xclock, suppose PrescaleDivider = 0x1F, then prescale = 65 (2059/32 = 64.35).

[15:8]	**PRESCALE**	**Row Scan Cycle Pre-scale Value** This value is used to pre-scale row scan cycle. The pre-scale counter is clocked by the divided crystal clock, xCLOCK. The divided number is from 1 to 256. E.g.If the crystal clock is 1Mhz then the xCLOCK period is 1us. If the keypad matric is 3x3 then Each row scan time = xCLOCK x PRESCALE x PrescaleDivider. Key array scan time = Each row scan time x ROWS. Example scan time for PRESCALE = 0x40, and PrescaleDivider = 0x1F. Each row scan time = 1us x 65 x 32 = 2.08 ms. Scan time = 2.08 x 3 = 6.24 ms. **Note:** When PRESCALE is determined, De-bounce sampling cycle should not exceed the half of (PRESCALE x PrescaleDivider), in the above example, and if scan row delay cycle is 4 xclock The maximum DBCLKSEL should be 1024 xCLOCK, bouncing time is 1 ms.
[7:6]	**Reserved**	Reserved.
[5]	**WAKEUP**	**Lower Power Wakeup Enable Bit** Setting this bit enables low power wakeup. 0 = Wakeup Disabled. 1 = Wakeup Enabled. **Note:** Setting the bit will force all KPI scan out to low.
[4]	**Reserved**	Reserved.
[3]	**INTEN**	**Key Interrupt Enable Bit** 0 = Keypad interrupt Disabled. 1 = Keypad interrupt Enabled. **Note:** The bit will be reset when KPI reset occurs.
[2]	**RKINTEN**	**Release Key Interrupt Enable Bit** The keypad controller will generate an interrupt when the controller detects keypad status changes from press to release. 0 = Keypad release interrupt Disabled. 1 = Keypad release interrupt Enabled. **Note:** The bit will be reset when KPI reset occurs.
[1]	**PKINTEN**	**Press Key Interrupt Enable Bit** The keypad controller will generate an interrupt when the controller detects any effective key press. 0 = Keypad press interrupt Disabled. 1 = Keypad press interrupt Enabled. **Note:** The bit will be reset when KPI reset occurs.
[0]	**ENKP**	**Keypad Scan Enable Bit** Setting this bit high enables the key scan function. 0 = Keypad scan Disabled. 1 = Keypad scan Enabled.

Keypad 3-keys Configuration Register (KPI3KCONF)

Register	Offset	R/W	Description	Reset Value
KPI3KCONF	KPI_BA+0x004	R/W	Keypad 3-keys Configuration Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							EN3KYRST
23	22	21	20	19	18	17	16
Reserved		K32R			K32C		
15	14	13	12	11	10	9	8
Reserved		K31R			K31C		
7	6	5	4	3	2	1	0
Reserved		K30R			K30C		

Bits	Description	
[31:25]	**Reserved**	Reserved.
[24]	**EN3KYRST**	**Enable Three-key Reset** Setting this bit enables hardware reset when three-key is detected 0 = Three-key function Disabled. 1 = Three-key function Enabled. **Note:** The bit will be reset when KPI reset occurs. The period of three key reset is 64 * KPI engine clock.
[23:22]	**Reserved**	Reserved.
[21:19]	**K32R**	**The #2 Key Row Address** The #2 means the row address and the column address is the highest of the specified 3-keys
[18:16]	**K32C**	**The #2 Key Column Address**
[15:14]	**Reserved**	Reserved.
[13:11]	**K31R**	**The #1 Key Row Address** The #1 means the row address and the column address is the 2nd of the specified 3-keys
[10:8]	**K31C**	**The #1 Key Column Address**
[7:6]	**Reserved**	Reserved.
[5:3]	**K30R**	**The #0 Key Row Address** The #0 means the row address and the column address is the lowest of the specified 3-keys
[2:0]	**K30C**	**The #0 Key Column Address**

Keypad Status Register (KPISTATUS)

Register	Offset	R/W	Description	Reset Value
KPISTATUS	KPI_BA+0x008	R/W	Keypad Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			PKEYINT	RKEYINT	KEYINT	RST3KEY	PDWAKE

Bits	Description	
[31:5]	**Reserved**	Reserved.
[4]	**PKEYINT**	**Press Key Interrupt** This bit indicates that some keys (one or multiple key) have been pressed. Read Operation: 0 = No key press. 1 = At least one key press. **Note:** To clear PKEYINT, software must clear each pressing event shown on “KPIKPE1, KPIKPE0”. C code example: DWORD PKE0, PKE1; PKE0 = reg_read(KPIKPE0); PKE1 = reg_read(KPIKPE1);. Reg_write(KPIKPE0, PKE0); Reg_write(KPIKPE1, PKE1);
[3]	**RKEYINT**	**Release Key Interrupt** This bit indicates that some keys (one or multiple key) have been released. Read Operation: 0 = No key release. 1 = At least one key release. **Note:** To clear RKEYINT, software must clear each releasing event shown on “key releasing event”. C code example: DWORD RKE0, RKE1; PKE0 = reg_read(KPIKRE0); PKE1 = reg_read(KPIKRE1);. Reg_write(KPIKRE0, RKE0); Reg_write(KPIKRE1, RKE1);
[2]	**KEYINT**	**Key Interrupt** This bit indicates the key scan interrupt is active when any key press or release or three key reset or wake up. Read Operation: 0 = No reset. 1 = Key press/release/3-key reset/wakeup interrupt occurred.

[1]	**RST3KEY**	**3-keys Reset Flag** This bit will be set after 3-keys reset occurs. Read Operation: 0 = No reset. 1 = 3 keys reset interrupt occurred. Write Operation: 0 = No operation. 1 = Clear interrupt flag.
[0]	**PDWAKE**	**Power Down Wakeup Flag** This flag indicates the chip is woken up from power down by keypad. Read Operation: 0 = No wakeup. 1 = Wake-up by keypad. Write Operation: 0 = No operation. 1 = Clear interrupt flag.

Keypad State Register 0 (KPIKEST0)

Register	Offset	R/W	Description	Reset Value
KPIKEST0	KPI_BA+0x010	R	Keypad State Register 0	0xFFFF_FFFF

31	30	29	28	27	26	25	24
KEST37	KEST36	KEST35	KEST34	KEST33	KEST32	KEST31	KEST30
23	22	21	20	19	18	17	16
KEST27	KEST26	KEST25	KEST24	KEST23	KEST22	KEST21	KEST20
15	14	13	12	11	10	9	8
KEST17	KEST16	KEST15	KEST14	KEST13	KEST12	KEST11	KEST10
7	6	5	4	3	2	1	0
KEST07	KEST06	KEST05	KEST04	KEST03	KEST02	KEST01	KEST00

Bits	Description	
[31:0]	**KESTmn**	**Key State** KESTm,n: m is row number, n is column number 0 = Key m,n is pressing. 1 = Key m,n is releasing.

Keypad State Register 1 (KPIKEST1)

Register	Offset	R/W	Description	Reset Value
KPIKEST1	KPI_BA+0x014	R	Keypad State Register 1	0xFFFF_FFFF

31	30	29	28	27	26	25	24
KEST77	KEST76	KEST75	KEST74	KEST73	KEST72	KEST71	KEST70
23	22	21	20	19	18	17	16
KEST67	KEST66	KEST65	KEST64	KEST63	KEST62	KEST61	KEST60
15	14	13	12	11	10	9	8
KEST57	KEST56	KEST55	KEST54	KEST53	KEST52	KEST51	KEST50
7	6	5	4	3	2	1	0
KEST47	KEST46	KEST45	KEST44	KEST43	KEST42	KEST41	KEST40

Bits	Description	
[31:0]	**KESTmn**	**Key State** KESTm,n: m is row number, n is column number 0 = Key m,n is pressing. 1 = Key m,n is releasing.

Lower 32 Key Press Event Indicator (KPIKPE0)

Register	Offset	R/W	Description	Reset Value
KPIKPE0	KPI_BA+0x018	R/W	Lower 32 Key Press Event Indicator	0x0000_0000

31	30	29	28	27	26	25	24
KPE37	KPE36	KPE35	KPE34	KPE33	KPE32	KPE31	KPE30
23	22	21	20	19	18	17	16
KPE27	KPE26	KPE25	KPE24	KPE23	KPE22	KPE21	KPE20
15	14	13	12	11	10	9	8
KPE17	KPE16	KPE15	KPE14	KPE13	KPE12	KPE11	KPE10
7	6	5	4	3	2	1	0
KPE07	KPE06	KPE05	KPE04	KPE03	KPE02	KPE01	KPE00

Bits	Description	
[31:0]	**KPEmn**	**Lower 32 Key Press Event Change Indicator** KPE mn[X] = 1, m=row, n=column. 0 = No key event. 1 = Corresponding key has a high to low event change. **Note:** Hardware will set this bit, and software should clear this bit by writing 1. Software can clear PKEYINT (KPISTATUS[4]) by writing 1 bit by bit to this register.

Upper 32 Key Press Event Indicator (KPIKPE1)

Register	Offset	R/W	Description	Reset Value
KPIKPE1	KPI_BA+0x01C	R/W	Upper 32 Key Press Event Indicator	0x0000_0000

31	30	29	28	27	26	25	24
KPE77	KPE76	KPE75	KPE74	KPE73	KPE72	KPE71	KPE70
23	22	21	20	19	18	17	16
KPE67	KPE66	KPE65	KPE64	KPE63	KPE62	KPE61	KPE60
15	14	13	12	11	10	9	8
KPE57	KPE56	KPE55	KPE54	KPE53	KPE52	KPE51	KPE50
7	6	5	4	3	2	1	0
KPE47	KPE46	KPE45	KPE44	KPE43	KPE42	KPE41	KPE40

Bits	Description	
[31:0]	**KPEmn**	**Upper 32 Key Press Event Change Indicator** KPE mn[X] = 1, m=row, n=column. 0 = No key event. 1 = Corresponding key has a high to low event change. **Note:** Hardware will set this bit, and software should clear this bit by writing 1. Software can clear PKEYINT (KPISTATUS[4]) by writing 1 bit by bit to this register.

Lower 32 Key Release Event Indicator (KPIKRE0)

Register	Offset	R/W	Description	Reset Value
KPIKRE0	KPI_BA+0x020	R/W	Lower 32 Key Release Event Indicator	0x0000_0000

31	30	29	28	27	26	25	24
KRE37	KRE36	KRE35	KRE34	KRE33	KRE32	KRE31	KRE30
23	22	21	20	19	18	17	16
KRE27	KRE26	KRE25	KRE24	KRE23	KRE22	KRE21	KRE20
15	14	13	12	11	10	9	8
KRE17	KRE16	KRE15	KRE14	KRE13	KRE12	KRE11	KRE10
7	6	5	4	3	2	1	0
KRE07	KRE06	KRE05	KRE04	KRE03	KRE02	KRE01	KRE00

Bits	Description	
[31:0]	**KREmn**	**Lower 32 Key Release Event Change Indicator** KRE mn[X] = 1, m=row, n=column. 0 = No key event. 1 = Corresponding key has a low to high event change. **Note:** Hardware will set this bit, and software should clear this bit by writing 1. Software can clear RKEYINT (KPISTATUS[3]) by writing 1 bit by bit to this register.

Upper 32 Key Release Event Indicator (KPIKRE1)

Register	Offset	R/W	Description	Reset Value
KPIKRE1	KPI_BA+0x024	R/W	Upper 32 Key Release Event Indicator	0x0000_0000

31	30	29	28	27	26	25	24
KRE77	KRE76	KRE75	KRE74	KRE73	KRE72	KRE71	KRE70
23	22	21	20	19	18	17	16
KRE67	KRE66	KRE65	KRE64	KRE63	KRE62	KRE61	KRE60
15	14	13	12	11	10	9	8
KRE57	KRE56	KRE55	KRE54	KRE53	KRE52	KRE51	KRE50
7	6	5	4	3	2	1	0
KRE47	KRE46	KRE45	KRE44	KRE43	KRE42	KRE41	KRE40

Bits	Description	
[31:0]	**KREmn**	**Upper 32 Key Release Event Change Indicator** KRE mn[X] = 1, m=row, n=column. 0 = No key event. 1 = Corresponding key has a low to high event change. **Note:** Hardware will set this bit, and software should clear this bit by writing 1. Software can clear RKEYINT (KPISTATUS[3]) by writing 1 bit by bit to this register.

Pre-scale Divider (KPIPRESCALDIV)

Register	Offset	R/W	Description	Reset Value
KPIPRESCALDIV	KPI_BA+0x028	R/W	Pre-scale Divider	0x0000_001F

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRESCALDIV							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**PRESCALDIV**	**Pre-scale Divider** This value is used to divide RESCALE that is set in KPICONF[15:8]. The prescale divider counter is clocked by the divided crystal clock, xCLOCK. The number is from 1 to 256. E.g. If the crystal clock is 1Mhz then the xCLOCK period is 1us. If the keypad matrix is 3x3. Then, each row scan time = xCLOCK x PRESCALE x PrescaleDivider. key array scan time = each row scan time x ROWS. example scan time for PRESCALE = 0x40, and PrescaleDivider = 0x1F. each row scan time = 1us x 65 x 32 = 2.08 ms. scan time = 2.08 x 3 = 6.24 ms. **Note:** When PRESCALE (KPICONF[15:8]) is determined, De-bounce sampling cycle should not exceed the half of (PRESCALE x PrescaleDivider), in the above example, and if scan row delay cycle is 4 xclock The maximum DBCLKSEL(KPICONF[19:16]) should be 1024 x clock, bouncing time is 1 ms.

6.44 Analog to Digital Converter (ADC)

6.44.1 Overview

The chip contains one 12-bit Successive Approximation Register analog-to-digital converter (SAR A/D converter) with 8 input channels. The A/D converter supports two operation modes: 4-wire or 5-wire mode. The ADC is especially suitable to act as touch screen controller.

6.44.2 Features

- Resolution: 12-bit resolution.
- DNL: +/-1.5 LSB, INL: +/-3 LSB.
- Maximum ADC clock frequency is 16 MHz
- Up to 727.2 KSPS conversion rate when ADC clock frequency is 16 MHz in high speed mode
- Up to 145.4 KSPS conversion rate when ADC clock frequency is 3.2 MHz in low speed mode
- Analog Input Range: V_{REF} to AGND, can be rail-to-rail.
- Analog Supply: 2.7~3.6V.
- Digital Supply: 0.99~1.21V.
- 8 Single-Ended analog inputs.
- Compatible with 4-wire or 5-wire Touch Screen Interface.
- Touch Pressure Measurement for 4-wire touch screen application.
- Low Power Consumption: 600uA (727.2 KSPS) / 300uA (145.4 KSPS)

6.44.3 Block Diagram

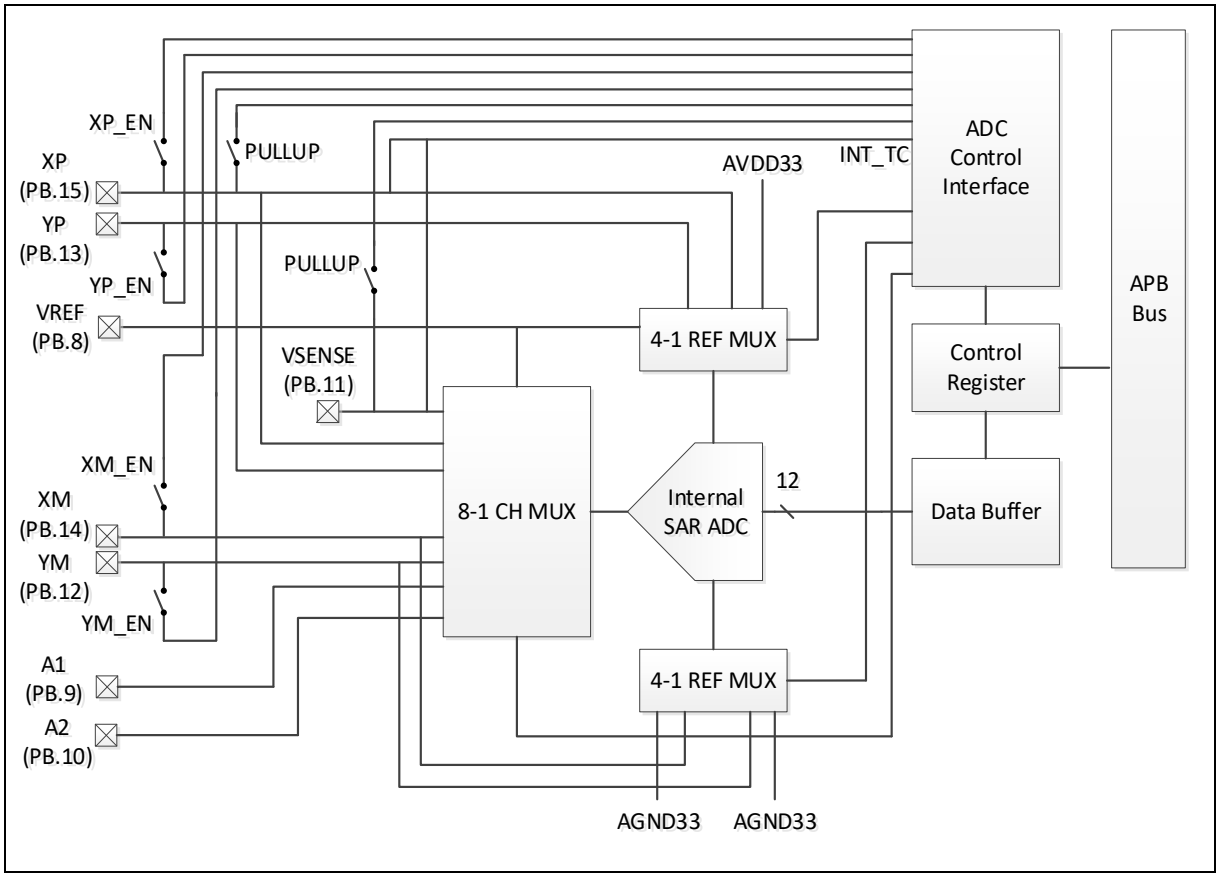


Figure 6.44-1 ADC Functional Block Diagram

6.44.4 Basic Configuration

- Clock Source Configuration
  - Select the clock divider number on ADC_DIV (CLK_CLKDIV4[20:4])
  - Enable ADC peripheral clock on ADCCKEN (CLK_APBCLK2[24]).
- Reset Configuration
  - Reset ADC controller on ADC0RST(SYS_IPRST2[31])
- Pin Configuration

Group	Pin Name	GPIO	MFP
ADC0	ADC0_CH0 (VREF)	PB.8	MFP8
	ADC0_CH1 (A1)	PB.9	MFP8
	ADC0_CH2 (A2)	PB.10	MFP8
	ADC0_CH3 (VSENSE)	PB.11	MFP8
	ADC0_CH4 (YM)	PB.12	MFP8
	ADC0_CH5 (YP)	PB.13	MFP8
	ADC0_CH6 (XM)	PB.14	MFP8
	ADC0_CH7 (XP)	PB.15	MFP8

6.44.5 Functional Description

6.44.5.1 ADC Transfer Function

The code output by ADC (transfer function) is encoded in binary, 1LSB=V_{REF}/4096, the transfer characteristic is shown in Figure 6.44-2:

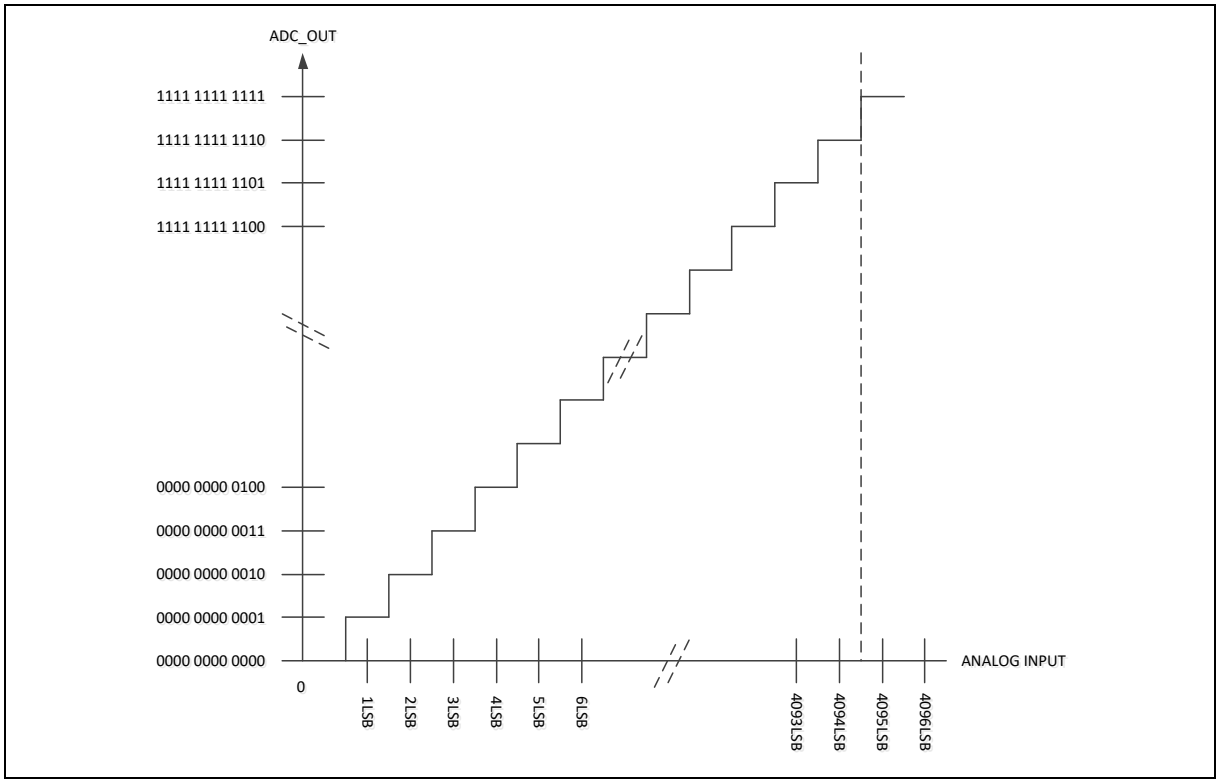


Figure 6.44-2 ADC Transfer Function

6.44.5.2 Selection of Input Signal

CHSEL[2:0]	Select ADC Analog Input Signal	Description
000	V_{REF}	V_{REF} input. Positive ADC reference voltage. When SPEED (ADC_CONF[22]) is set to 1, it supports low speed mode and ADC clock should be slower than 3.2 MHz; when SPEED is set to 0, it supports high speed mode and ADC clock should be slower than 16 MHz
001	A1	Normal input. ADC analog input. When SPEED is set to 1, it supports low speed mode and ADC clock should be slower than 3.2 MHz; when SPEED is set to 0, it supports high speed mode and ADC clock should be slower than 16 MHz
010	A2	Normal input. ADC analog input. When SPEED is set to 1, it supports low speed mode and ADC clock should be slower than 3.2 MHz; when SPEED is set to 0, it supports high speed mode and ADC clock should be slower than 16 MHz
011	VSENSE	Used for 5-wire touch screen detection. When SPEED is set to 1, it supports low speed mode and ADC clock

		should be slower than 3.2 MHz; when SPEED is set to 0, it supports high speed mode and ADC clock should be slower than 16 MHz
100	YM	If used in touch screen, it should connect to the negative end of Y axis. If used in 5-wire touch screen, it should connect to the lower-left electrode. When SPEED is set to 1, it supports low speed mode and ADC clock should be slower than 3.2 MHz; when SPEED is set to 0, it supports high speed mode and ADC clock should be slower than 16 MHz
101	YP	If used in touch screen, it should connect to the positive end of Y axis. If used in 5-wire touch screen, it should connect to the upper-right electrode. When SPEED is set to 1, it supports low speed mode and ADC clock should be slower than 3.2 MHz; when SPEED is set to 0, it supports high speed mode and ADC clock should be slower than 16 MHz
110	XM	If used in touch screen, it should connect to the negative end of X axis. If used in 5-wire touch screen, it should connect to the lower-right electrode. When SPEED is set to 1, it supports low speed mode and ADC clock should be slower than 3.2 MHz; when SPEED is set to 0, it supports high speed mode and ADC clock should be slower than 16 MHz
111	XP	If used in touch screen, it should connect to the positive end of X axis. If used in 5-wire touch screen, it should connect to the upper-left electrode. When SPEED is set to 1, it supports low speed mode and ADC clock should be slower than 3.2 MHz; when SPEED is set to 0, it supports high speed mode and ADC clock should be slower than 16 MHz

The input current on analog input pins depends on analog input voltage and sampling rate. In the sampling mode, the input current charges the internal sampling capacitor. After the capacitor is fully charged, the current will be off. The internal sampling capacitor must be charged to 12-bit settling level at sampling time, under afore mentioned condition.

The signal source impedance connected to analog input needs to satisfy the following expression and Figure 6.44-3:

$$0.69 \times (12 + 1) \times C_{samp} \times 1.1 \times (R_{src} + R_s) < \frac{1}{f_{clk}} \times 3$$

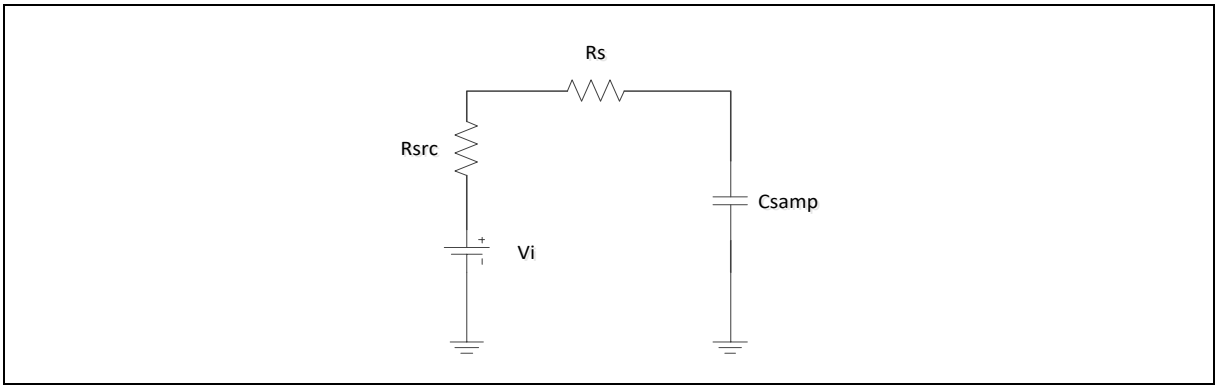


Figure 6.44-3 Simplified Sampling Diagram

6.44.5.3 Selection of Reference Voltage

REFSEL[1:0]	ADC Analog Reference Pair Selection Signals
00	AGND33 Vs V_{REF} input

01	YM Vs YP
10	XM Vs XP
11	AGND33 Vs AVDD33

Table 6.44-1 Relation between Reference Voltage and Reference Selection

Reference voltage can be selected according to the application. When REFSEL is set to 00, reference voltage of ADC is switched to V_{REF} pin. When REFSEL is set to 11, reference voltage of ADC is switched to AVDD33. For 4-wire/5-wire touch Screen Y axis conversion, REFSEL can be set to 01. For 4-wire/5-wire touch Screen X axis conversion, REFSEL can be set to 10.

6.44.5.4 ADC Timing Diagram

If the ADC is waken up from Power-down mode, ADEN(ADC_CTL[0]) = 0, after 100 us, the ADC is ready to receive the analog signal to start a conversion. The ADC needs time for the internal bias circuit setup, if the ADC didn't wait 100us, the ADC can still work but cannot guarantee 12-bit-accuracy. When A/D conversion for each enabled function is completed, the result is transferred to the A/D data register corresponding to each functional.

After analog ready, user can set MST(ADC_CTL[8]) to start first task which can be touch screen function or normal A/D conversion. The ADC need two ADC engine clocks to distinguish which function(touch screen or normal A/D conversion) be triggered by user and up to three ADC engine clocks to synchronize MST(ADC_CTL[8]) from PCLK to ADC engine clock domain. The SOC will rising after five ADC engine clocks, and analog need 16 ADC engine clocks to generate 12-bit result. The ADC requires an additional one ADC engine clock to synchronize the EOC signal from analog and two PCLK clocks to generate ADF.

The ADC conversion timing diagram is shown in Figure 6.44-4.

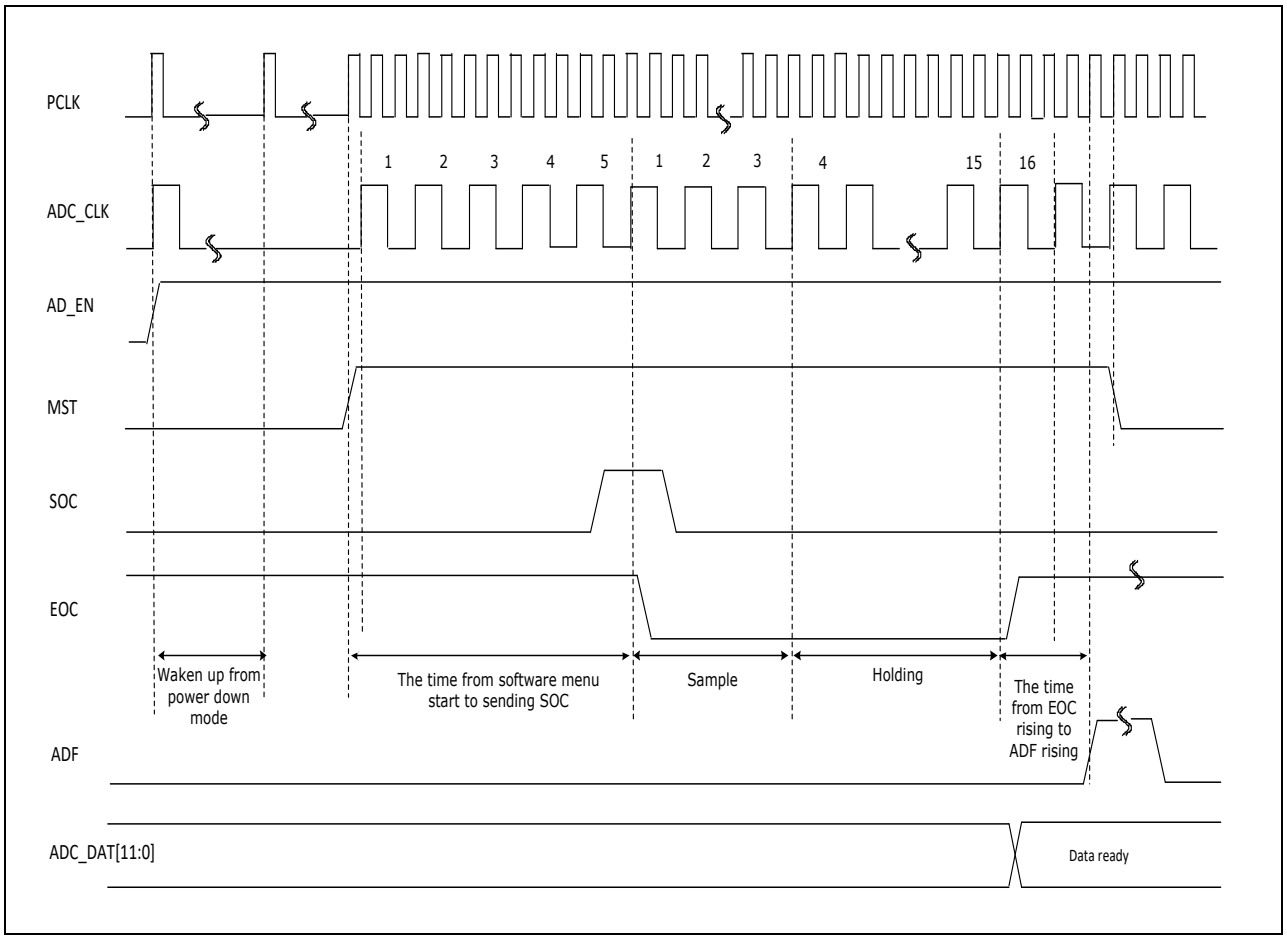


Figure 6.44-4 ADC Timing Diagram

6.44.5.5 4-wire and 5-wire Touch Screen

The touch screen control logic and the switch can control the 4-wire and 5-wire type touch screen. For ADC configure register TEN (ADC_CONF[0]) should be set to 1. ADC control register WMSWCH (ADC_CTL[16]) (Wire Mode Switch) for 5-wire/4-wire configuration. Figure 6.44-5 and Figure 6.44-6 show the interface for 4-wire, 5-wire touch screen respectively.

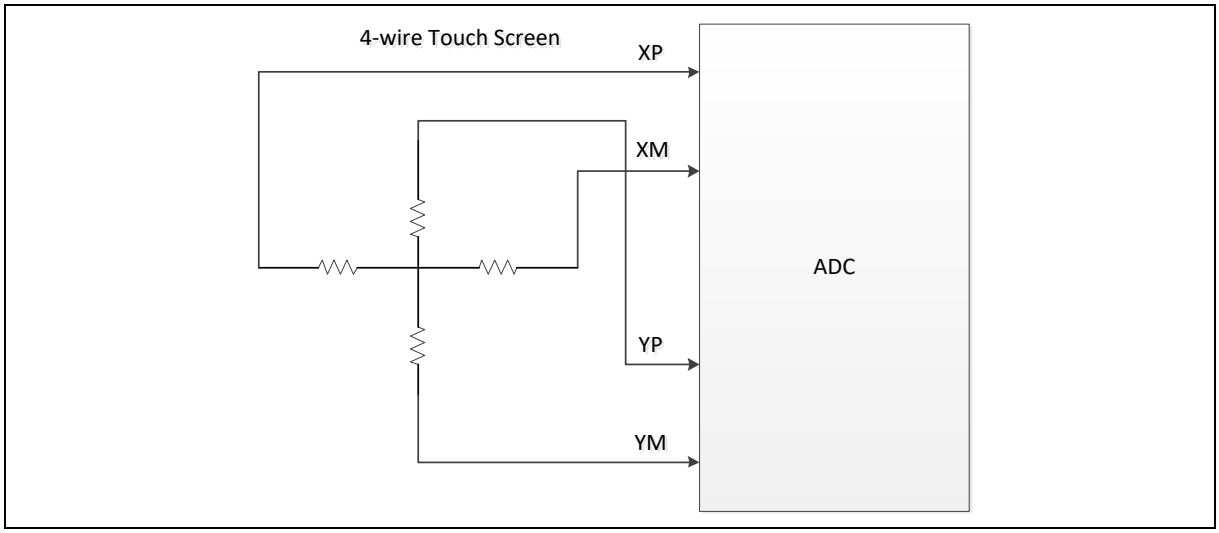


Figure 6.44-54-wire Touch Screen Connection Diagram

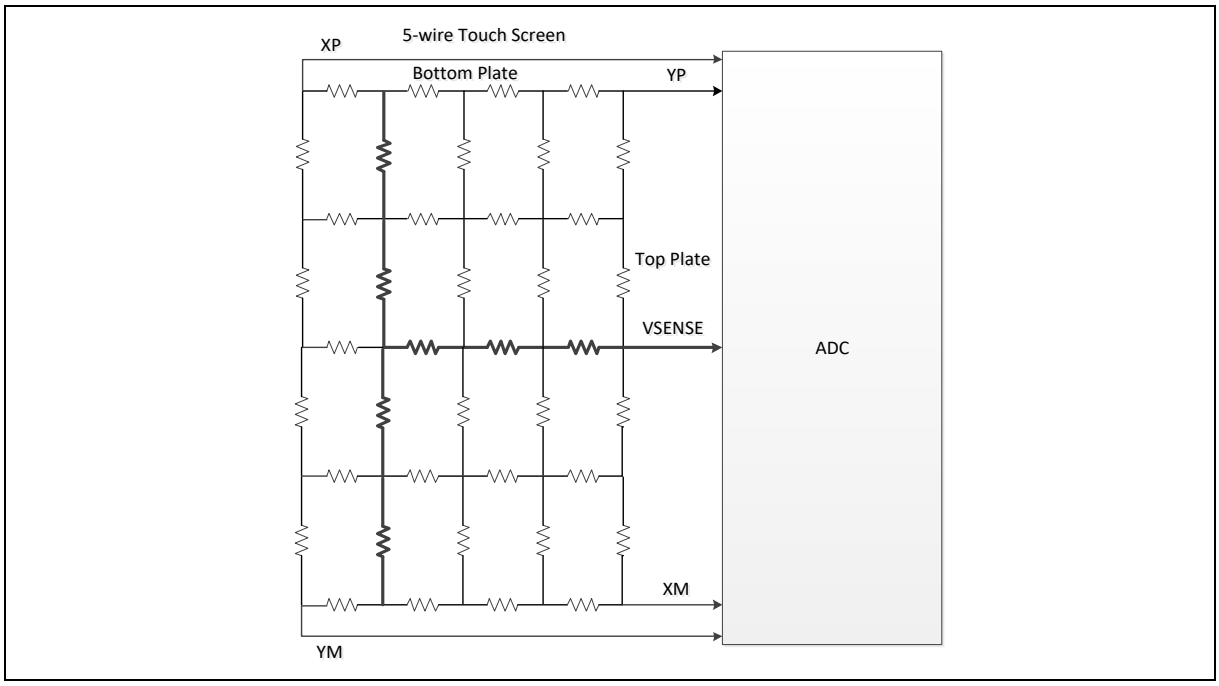


Figure 6.44-65-wire Touch Screen Connection Diagram

6.44.5.6 4-wire Pressure Measurement

To distinguish pen or finger touch, the pressure of the touch needs to be determined. The chip provides two solutions. The first method requires knowing the X-plate resistance, measurement of the X-Position, and two additional cross-panel measurements (Z1 and Z2) of the touch screen. Use the following Equation to calculate the touch resistance:

$$R_{Touch} = R_{X-plate} \times \frac{X_{position}}{4096}\left(\frac{Z_2}{Z_1} - 1\right)$$

The second method requires knowing both the X-plate and Y-plate resistance, measurement of X-Position and Y-Position, and Z1. Use the following Equation to calculate the touch resistance:

$$R_{Touch} = \frac{R_{X-plate} \times X_{position}}{4096}\left(\frac{4096}{Z_1} - 1\right) - R_{Y-plate}(1 - \frac{Y_{position}}{4096})$$

For ADC configure register ZEN (ADC_CONF[1]) should be set to 1. When ZEN (ADC_CONF[1]) is set; the touch pressure measure Z will be stored in this ADC_ZDATA register.

6.44.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write.

Register	Offset	R/W	Description	Reset Value
ADC Base Address: **ADC_BA = 0x4042_0000**				
ADC_CTL	ADC_BA+0x00	R/W	ADC Control	0x0000_0000
ADC_CONF	ADC_BA+0x04	R/W	ADC Configure	0x0000_0000
ADC_IER	ADC_BA+0x08	R/W	ADC Interrupt Enable Register	0x0000_0000
ADC_ISR	ADC_BA+0x0C	R/W	ADC Interrupt Status Register	0x0000_0000
ADC_WKISR	ADC_BA+0x10	R	ADC Wake-up interrupt Status Register	0x0000_0000
ADC_XYDATA	ADC_BA+0x20	R	ADC Touch X,Y Position Data	0x0000_0000
ADC_ZDATA	ADC_BA+0x24	R	ADC Touch Z Pressure Data	0x0000_0000
ADC_DATA	ADC_BA+0x28	R	ADC Normal Conversion Data	0x0000_0000
ADC_XYSORT0	ADC_BA+0x1F4	R	ADC Touch XY Position Mean Value Sort 0	0x0000_0000
ADC_XYSORT1	ADC_BA+0x1F8	R	ADC Touch XY Position Mean Value Sort 1	0x0000_0000
ADC_XYSORT2	ADC_BA+0x1FC	R	ADC Touch XY Position Mean Value Sort 2	0x0000_0000
ADC_XYSORT3	ADC_BA+0x200	R	ADC Touch XY Position Mean Value Sort 3	0x0000_0000
ADC_ZSORT0	ADC_BA+0x204	R	ADC Touch Z Pressure Mean Value Sort 0	0x0000_0000
ADC_ZSORT1	ADC_BA+0x208	R	ADC Touch Z Pressure Mean Value Sort 1	0x0000_0000
ADC_ZSORT2	ADC_BA+0x20C	R	ADC Touch Z Pressure Mean Value Sort 2	0x0000_0000
ADC_ZSORT3	ADC_BA+0x210	R	ADC Touch Z Pressure Mean Value Sort 3	0x0000_0000

6.44.7 Register Description

ADC Control (ADC_CTL)

Register	Offset	R/W	Description	Reset Value
ADC_CTL	ADC_BA+0x00	R/W	ADC Control	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							WMSWCH
15	14	13	12	11	10	9	8
Reserved				WKTEN	Reserved	PEDEEN	MST
7	6	5	4	3	2	1	0
Reserved							ADEN

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**WMSWCH**	**Wire Mode Switch for 5-wire/4-wire Configuration** 0 = 4-wire mode. 1 = 5-wire mode.
[15:12]	**Reserved**	Reserved.
[11]	**WKTEN**	**Touch Wake Up Enable Bit** 0 = Touch wake-up Disabled. 1 = Touch wake-up Enabled.
[10]	**Reserved**	Reserved.
[9]	**PEDEEN**	**Pen Down Event Enable Bit** 0 = Pen down event interrupt Disabled. 1 = Pen down event interrupt Enabled.
[8]	**MST**	**Menu Start Conversion** 0 = Functional menu not started. 1 = Start all enable bit in ADC_CONF register. **Note:** This bit is set by software and cleared by hardware when all the tasks listed in ADC_CONF are done.
[7:1]	**Reserved**	Reserved.
[0]	**ADEN**	**ADC Power Control** 0 = Power down ADC. 1 = Power on ADC.

ADC Configure (ADC_CONF)

Register	Offset	R/W	Description	Reset Value
ADC_CONF	ADC_BA+0x04	R/W	ADC Configure	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved	SPEED	ZMAVDIS	TMAVDIS	Reserved			
15	14	13	12	11	10	9	8
Reserved	CHSEL			Reserved			
7	6	5	4	3	2	1	0
REFSEL		Reserved			NACEN	ZEN	TEN

Bits	Description	
[31:23]	**Reserved**	Reserved.
[22]	**SPEED**	**Speed Mode Selection** 0 = All ADC channels set to high speed mode. 1 = All ADC channels set to low speed mode.
[21]	**ZMAVDIS**	**Display Z Mean Average Disable Bit** Pressure mean average for Z1 and Z2 function disable bit. 0 = Pressure mean average for Z1 and Z2 function Enabled. 1 = Pressure mean average for Z1 and Z2 function Disabled.
[20]	**TMAVDIS**	**Display T Mean Average Disable Bit** Touch mean average for X and Y function disable bit. 0 = Touch mean average for X and Y function Enabled. 1 = Touch mean average for X and Y function Disabled.
[19:15]	**Reserved**	Reserved.
[14:12]	**CHSEL**	**Channel Selection** ADC input channel selection. 000 = V_{REF}. 001 = A1. 010 = A2. 011 = VSENSE. 100 = YM. 101 = YP. 110 = XM. 111 = XP.
[11:8]	**Reserved**	Reserved.

[7:6]	**REFSEL**	**ADC Reference Select** ADC reference voltage select when ADC operates in normal conversion. 00 = AGND33 vs V_{REF} input. 01 = YM vs YP. 10 = XM vs XP. 11 = AGND33 vs AVDD33.
[5:3]	**Reserved**	Reserved.
[2]	**NACEN**	**Normal A/D Conversion Enable Bit** 0 = Normal A/D Conversion Disabled. 1 = Normal A/D Conversion Enabled.
[1]	**ZEN**	**Press Measure Enable Bit** 1 = Press measure function Disabled. 1 = Press measure function Enabled.
[0]	**TEN**	**Touch Detection Enable Bit** 0 = Touch detection function Disabled. 1 = Touch detection function Enabled.

ADC Interrupt Enable Register (ADC_IER)

Register	Offset	R/W	Description	Reset Value
ADC_IER	ADC_BA+0x08	R/W	ADC Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	PEUEIEN	Reserved		WKTIEN	PEDEIEN	Reserved	MIEN

Bits	Description	
[31:7]	**Reserved**	Reserved.
[6]	**PEUEIEN**	**Pen Up Event Interrupt Enable Bit** 0 = Pen up event detection interrupt Disabled. 1 = Pen up event detection interrupt Enabled.
[5:4]	**Reserved**	Reserved.
[3]	**WKTIEN**	**Wake Up Touch Interrupt Enable Bit** 0 = Wake up touch detection interrupt Disabled. 1 = Wake up touch detection interrupt Enabled.
[2]	**PEDEIEN**	**Pen Down Event Interrupt Enable Bit** 0 = Pen down event detection interrupt Disabled. 1 = Pen down event detection interrupt Enabled.
[1]	**Reserved**	Reserved.
[0]	**MIEN**	**Menu Interrupt Enable Bit** Function menu complete interrupt enable. 0 = Menu interrupt Disabled. 1 = Menu interrupt Enabled.

ADC Interrupt Status Register (ADC_ISR)

Register	Offset	R/W	Description	Reset Value
ADC_ISR	ADC_BA+0x0C	R/W	ADC Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						INTTC	Reserved
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					NACF	ZF	TF
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			PEUEF	Reserved	PEDEF	Reserved	MF

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17]	**INTTC**	**Interrupt Signal for Touch Screen Touching Detection** This signal is directly from analog macro without de-bouncing and can be used to determine the pen down touch event together with PEDEF (ADC_ISR[2]) flag.
[16:11]	**Reserved**	Reserved.
[10]	**NACF**	**Normal AD Conversion Finish** Functional menu normal AD conversion finish. **Note:** Set by hardware and write 1 to clear this bit.
[9]	**ZF**	**Press Conversion Finish** Functional menu press measure conversion finish. **Note:** Set by hardware and write 1 to clear this bit.
[8]	**TF**	**Touch Conversion Finish** Functional menu touch detection conversion finish. **Note:** Set by hardware and write 1 to clear this bit.
[7:5]	**Reserved**	Reserved.
[4]	**PEUEF**	**Pen Up Event Flag** Pen up event status indicator. **Note:** Set by hardware and write 1 to clear this bit.
[3]	**Reserved**	Reserved.
[2]	**PEDEF**	**Pen Down Event Flag** Pen down event status indicator. **Note:** Set by hardware and write 1 to clear this bit.
[1]	**Reserved**	Reserved.
[0]	**MF**	**Menu Complete Flag** Function menu complete status indicator.

		Note: Set by hardware and write 1 to clear this bit.

ADC Wake-up Interrupt Status Register (WKISR)

Register	Offset	R/W	Description	Reset Value
ADC_WKISR	ADC_BA+0x10	R	ADC Wake-up interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**WPEDEF**	**Reserved**

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**WPEDEF**	**Wake Up Pen Down Event Flag** Pen down event wake up status indicator.
[0]	**Reserved**	Reserved.

ADC Touch X,Y Position Data (ADC_XYDATA)

Register	Offset	R/W	Description	Reset Value
ADC_XYDATA	ADC_BA+0x20	R	ADC Touch X,Y Position Data	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				YDATA			
23	22	21	20	19	18	17	16
YDATA							
15	14	13	12	11	10	9	8
Reserved				XDATA			
7	6	5	4	3	2	1	0
XDATA							

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:16]	**YDATA**	**ADC Y Data** When TEN (ADC_CONF[0]) is set, the touch y-position will be stored in this register. **Note:** If the TMAVDIS (ADC_CONF[20]) = 0, both x and y position are the results of the mean average of x and y in ADC_XYSORT0 ~ ADC_XYSORT3.
[15:12]	**Reserved**	Reserved.
[11:0]	**XDATA**	**ADC X Data** When TEN (ADC_CONF[0]) is set, the touch x-position will be stored in this register. **Note:** If the TMAVDIS (ADC_CONF[20]) = 0, both x and y position are the results of the mean average of x and y in ADC_XYSORT0 ~ ADC_XYSORT3.

ADC Touch Z Pressure Data (ADC_ZDATA)

Register	Offset	R/W	Description	Reset Value
ADC_ZDATA	ADC_BA+0x24	R	ADC Touch Z Pressure Data	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				Z2DATA			
23	22	21	20	19	18	17	16
Z2DATA							
15	14	13	12	11	10	9	8
Reserved				Z1DATA			
7	6	5	4	3	2	1	0
Z1DATA							

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:16]	**Z2DATA**	**ADC Z2 Data** When ZEN (ADC_CONF[1]) is set; the touch pressure measure Z2 will be stored in this register. **Note:** If the ZMAVDIS (ADC_CONF[21]) = 0, both Z1 and Z2 data is the results of the mean average of Z1 and Z2 in ADC_ZSORT0 ~ ADC_ZSORT3.
[15:12]	**Reserved**	Reserved.
[11:0]	**Z1DATA**	**ADC Z1 Data** When ZEN (ADC_CONF[1]) is set; the touch pressure measure Z1 will be stored in this register. **Note:** If the ZMAVDIS (ADC_CONF[21]) = 0, both Z1 and Z2 data is the results of the mean average of Z1 and Z2 in ADC_ZSORT0 ~ ADC_ZSORT3.

ADC Normal Conversion Data (ADC_DATA)

Register	Offset	R/W	Description	Reset Value
ADC_DATA	ADC_BA+0x28	R	ADC Normal Conversion Data	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				ADCDATA			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADCDATA							

Bits	Description	
[31:12]	**Reserved**	Reserved.
[11:0]	**ADCDATA**	**ADC Data** When NACEN (ADC_CONF[2]) is enabled, the AD converting result with corresponding channel is stored in this register.

ADC Touch XY Position Mean Value Sort 0 (ADC_XYSORT0)

Register	Offset	R/W	Description	Reset Value
ADC_XYSORT0	ADC_BA+0x1F4	R	ADC Touch XY Position Mean Value Sort 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				YSORT0			
23	22	21	20	19	18	17	16
YSORT0							
15	14	13	12	11	10	9	8
Reserved				XSORT0			
7	6	5	4	3	2	1	0
XSORT0							

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:16]	**YSORT0**	**Y Position Sort Data 0** Y position mean average sort data 0.
[15:12]	**Reserved**	Reserved.
[11:0]	**XSORT0**	**X Position Sort Data 0** X position mean average sort data 0.

ADC Touch XY Position Mean Value Sort 1 (ADC_XYSORT1)

Register	Offset	R/W	Description	Reset Value
ADC_XYSORT1	ADC_BA+0x1F8	R	ADC Touch XY Position Mean Value Sort 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				YSORT1			
23	22	21	20	19	18	17	16
YSORT1							
15	14	13	12	11	10	9	8
Reserved				XSORT1			
7	6	5	4	3	2	1	0
XSORT1							

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:16]	**YSORT1**	**Y Position Sort Data 1** Y position mean average sort data 1.
[15:12]	**Reserved**	Reserved.
[11:0]	**XSORT1**	**X Position Sort Data 1** X position mean average sort data 1.

ADC Touch XY Position Mean Value Sort 2 (ADC_XYSORT2)

Register	Offset	R/W	Description	Reset Value
ADC_XYSORT2	ADC_BA+0x1FC	R	ADC Touch XY Position Mean Value Sort 2	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				YSORT2			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
YSORT2							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				XSORT2			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
XSORT2							

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:16]	**YSORT2**	**Y Position Sort Data 2** Y position mean average sort data 2.
[15:12]	**Reserved**	Reserved.
[11:0]	**XSORT2**	**X Position Sort Data 2** X position mean average sort data 2.

ADC Touch XY Position Mean Value Sort 3 (ADC_XYSORT3)

Register	Offset	R/W	Description	Reset Value
ADC_XYSORT3	ADC_BA+0x200	R	ADC Touch XY Position Mean Value Sort 3	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				YSORT3			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
YSORT3							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				XSORT3			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
XSORT3							

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:16]	**YSORT3**	**Y Position Sort Data 3** Y position mean average sort data 3.
[15:12]	**Reserved**	Reserved.
[11:0]	**XSORT3**	**X Position Sort Data 3** X position mean average sort data 3.

ADC Touch Z Pressure Mean Value Sort 0 (ADC_ZSORT0)

Register	Offset	R/W	Description	Reset Value
ADC_ZSORT0	ADC_BA+0x204	R	ADC Touch Z Pressure Mean Value Sort 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				Z2SORT0			
23	22	21	20	19	18	17	16
Z2SORT0							
15	14	13	12	11	10	9	8
Reserved				Z1SORT0			
7	6	5	4	3	2	1	0
Z1SORT0							

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:16]	**Z2SORT0**	**Z2 Position Sort Data 0** Z2 position Mean average sort data 0.
[15:12]	**Reserved**	Reserved.
[11:0]	**Z1SORT0**	**Z1 Position Sort Data 0** Z1 position Mean average sort data 0.

ADC Touch Z Pressure Mean Value Sort 1 (ADC_ZSORT1)

Register	Offset	R/W	Description	Reset Value
ADC_ZSORT1	ADC_BA+0x208	R	ADC Touch Z Pressure Mean Value Sort 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				Z2SORT1			
23	22	21	20	19	18	17	16
Z2SORT1							
15	14	13	12	11	10	9	8
Reserved				Z1SORT1			
7	6	5	4	3	2	1	0
Z1SORT1							

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:16]	**Z2SORT1**	**Z2 Position Sort Data 1** Z2 position Mean average sort data 1.
[15:12]	**Reserved**	Reserved.
[11:0]	**Z1SORT1**	**Z1 Position Sort Data 1** Z1 position Mean average sort data 1.

ADC Touch Z Pressure Mean Value Sort 2 (ADC_ZSORT2)

Register	Offset	R/W	Description	Reset Value
ADC_ZSORT2	ADC_BA+0x20C	R	ADC Touch Z Pressure Mean Value Sort 2	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				Z2SORT2			
23	22	21	20	19	18	17	16
Z2SORT2							
15	14	13	12	11	10	9	8
Reserved				Z1SORT2			
7	6	5	4	3	2	1	0
Z1SORT2							

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:16]	**Z2SORT2**	**Z2 Position Sort Data 2** Z2 position Mean average sort data 2.
[15:12]	**Reserved**	Reserved.
[11:0]	**Z1SORT2**	**Z1 Position Sort Data 2** Z1 position Mean average sort data 2.

ADC Touch Z Pressure Mean Value Sort 3 (ADC_ZSORT3)

Register	Offset	R/W	Description	Reset Value
ADC_ZSORT3	ADC_BA+0x210	R	ADC Touch Z Pressure Mean Value Sort 3	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				Z2SORT3			
23	22	21	20	19	18	17	16
Z2SORT3							
15	14	13	12	11	10	9	8
Reserved				Z1SORT3			
7	6	5	4	3	2	1	0
Z1SORT3							

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:16]	**Z2SORT3**	**Z2 Position Sort Data 3** Z2 position Mean average sort data 3.
[15:12]	**Reserved**	Reserved.
[11:0]	**Z1SORT3**	**Z1 Position Sort Data 3** Z1 position Mean average sort data 3.

6.45 Enhanced 12-bit Analog-to-Digital Converter (EADC)

6.45.1 Overview

The chip contains one 12-bit successive approximation analog-to-digital converter (SAR EADC converter) with 8 external input channels and 1 internal channel. The EADC converter can be started by software trigger, EPWM0/1/2 triggers, Timer0~11 overflow pulse triggers, ADINT0, ADINT1 interrupt EOC (End of conversion) pulse trigger and external pin (EADC0_ST) input signal.

6.45.2 Features

- Analog input voltage range: 0~V_{REF} (Max to 3.6V)
- Reference voltage from V_{REF} pin
- 12-bit resolution
- Up to 8 single-end analog external input channels and 4 pair differential analog input channels
- Up to 1 internal channel for Battery power (V_{BAT})
- Four EADC interrupts (ADINT0~3) with individual interrupt vector addresses
- Maximum EADC clock frequency of 80 MHz
- Up to 4.7 MSPS conversion rate when EADC clock frequency is 80 MHz in high speed mode
- Up to 941.1 KSPS conversion rate when EADC clock frequency is 16 MHz in low speed mode
- Configurable EADC internal sampling time
- Supports calibration function
- Supports internal reference voltage V_{REF}: 1.6V, 2.0V, 2.5V, and 3.0V
- Supports three power saving modes:
  - Deep Power-down mode
  - Power-down mode
  - Standby mode
- Up to 9 sample modules:
  - Each of sample modules which is configurable for EADC converter channel EADC_CH0~7 and trigger source
  - Sample module 8 is fixed for EADC channel 8 input source as battery power (V_{BAT})
  - Double buffer for sample control logic module 0~3
  - Configurable sampling time for each sample module
  - Conversion results are held in 9 data registers with valid and overrun indicators
- An EADC conversion can be started by:
  - Write 1 to SWTRG (EADC_SWTRG[n], n = 0~8)
  - External pin EADC0_ST
  - Timer0~11 overflow pulse triggers
  - ADINT0 and ADINT1 interrupt EOC (End of conversion) pulse triggers
  - EPWM triggers

- Supports PDMA transfer
- Conversion Result Monitor by Compare Mode

6.45.3 Block Diagram

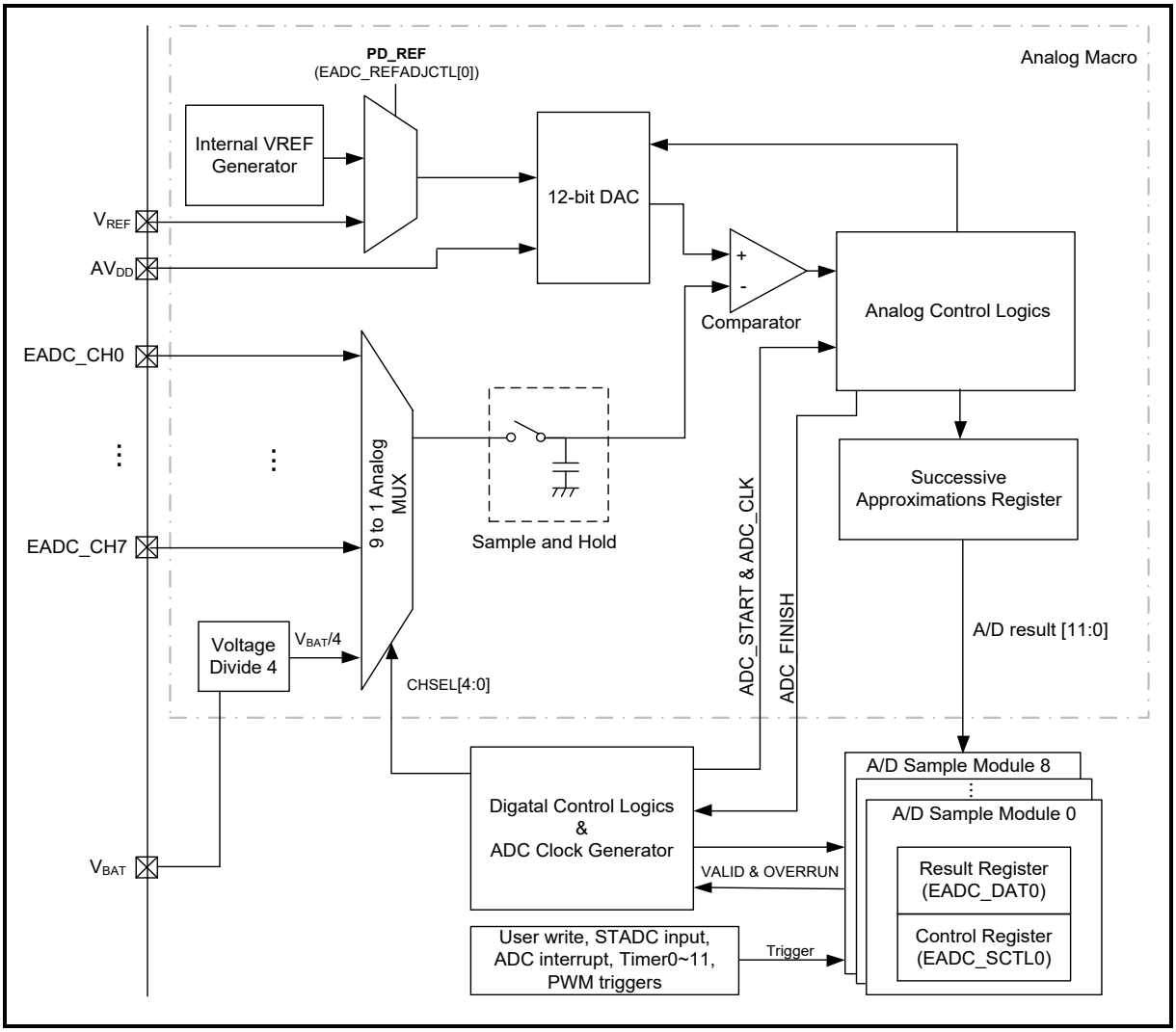


Figure 6.45-1 EADC Converter Block Diagram

6.45.4 Basic Configuration

- Clock Source Configuration
  - Select the clock divider number on EADCDIV (CLK_CLKDIV4[3:0])
  - Enable EADC peripheral clock in EADCCKEN (CLK_APBCLK2[25]).
- Reset Configuration
  - Reset EADC controller in EADCRST (EADC_CTL [1]).
  - Reset EADC controller in EADCRST(SYS_IPRST1[28]).
- Pin Configuration

Group	Pin Name	GPIO	MFP

EADC0	EADC0_CH0	-	MFP8
	EADC0_CH1	-	MFP8
	EADC0_CH2	-	MFP8
	EADC0_CH3	-	MFP8
	EADC0_CH4	-	MFP8
	EADC0_CH5	-	MFP8
	EADC0_CH6	-	MFP8
	EADC0_CH7	-	MFP8
	EADC0_ST	PF.14	MFP2
		PG.0, PJ.15, PK.1	MFP6
		PK.8	MFP7
		PK.5, PN.15	MFP9
		PD.7	MFP11

6.45.5 Functional Description

The EADC controller consists of a 9 channel analog switch, 9 sample modules and a 12-bit successive approximation analog-to-digital converter. The EADC operation is based on sample module 0~8, and each of them has its configuration to decide which trigger source to start the conversion, which channel to convert. Sample module 0~7 can be configured to EADC_CH0~7 channel, and different trigger source. It provides user a flexible means to get the over-sampling results. The sample module 0~7 and sample module 8 are shown as follows.

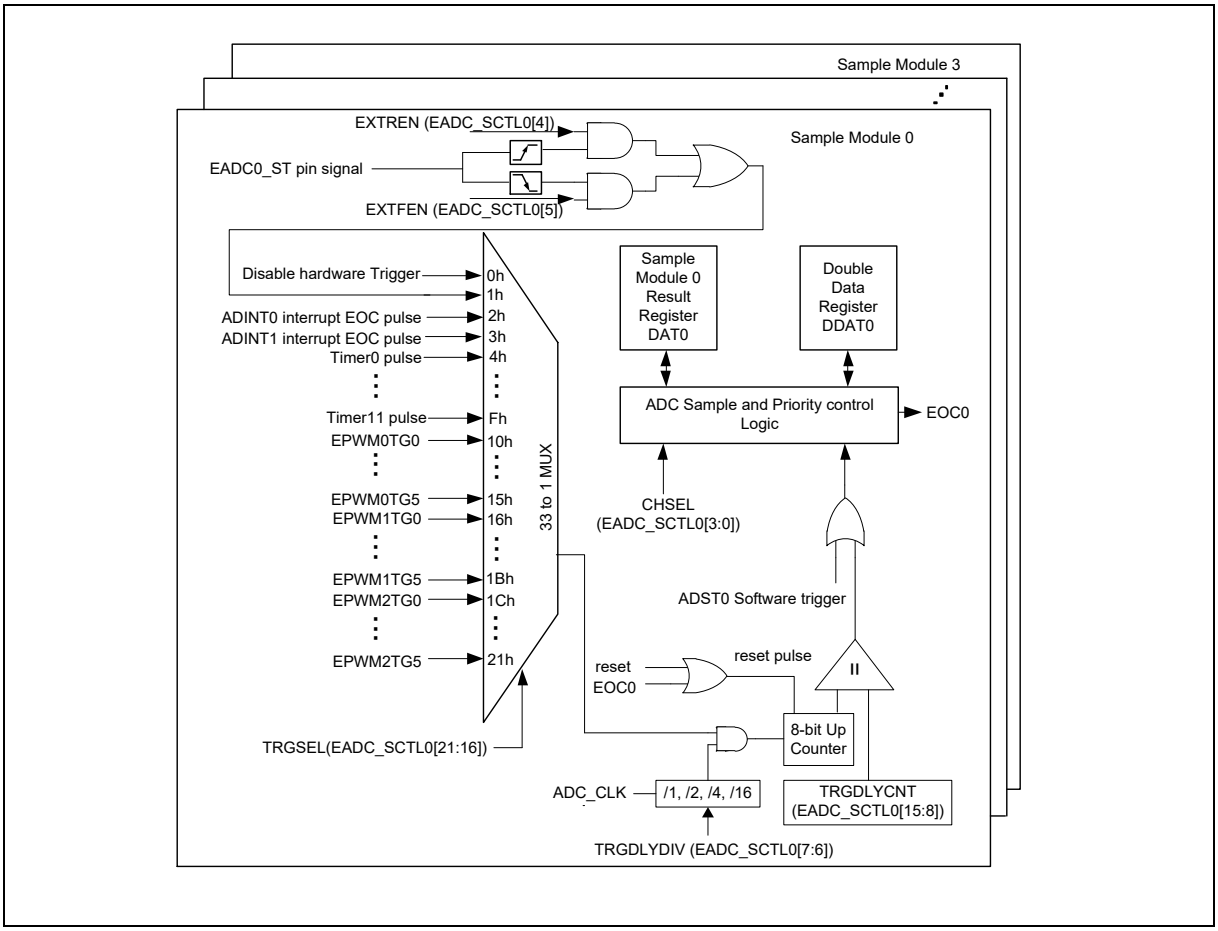


Figure 6.45-2 Sample Module 0~3 Block Diagram

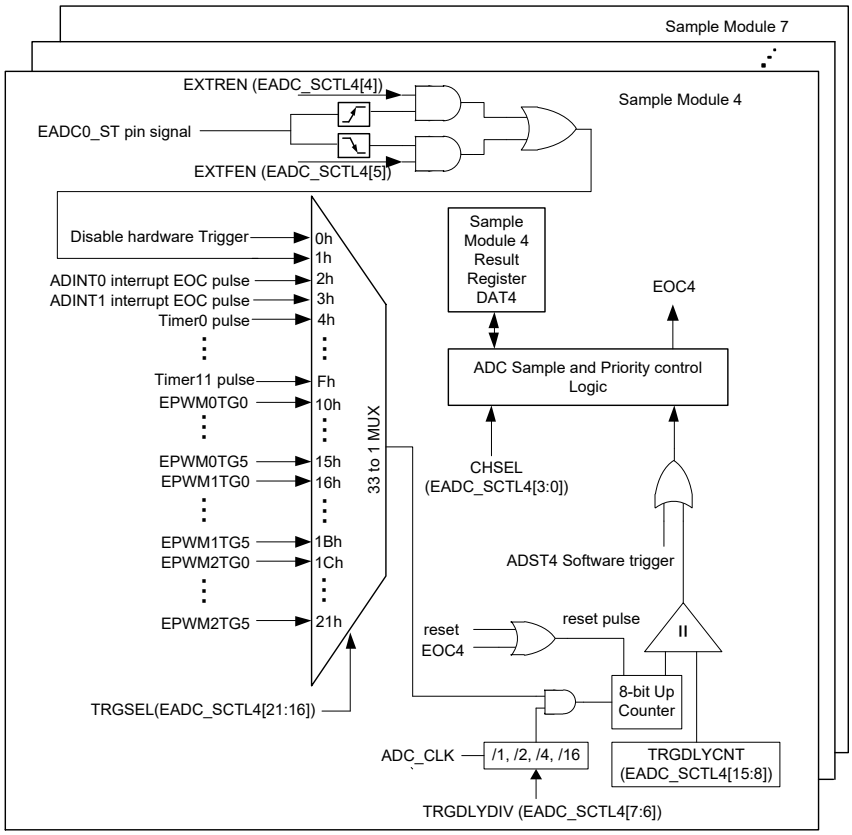


Figure 6.45-3 Sample Module 4~7 Block Diagram

Sample module 8 can convert internal channel (V_{BAT}) and can be triggered by user write SWTRG (EADC_SWTRG[n], n = 8). Figure 6.45-4 shows the sample module 8.

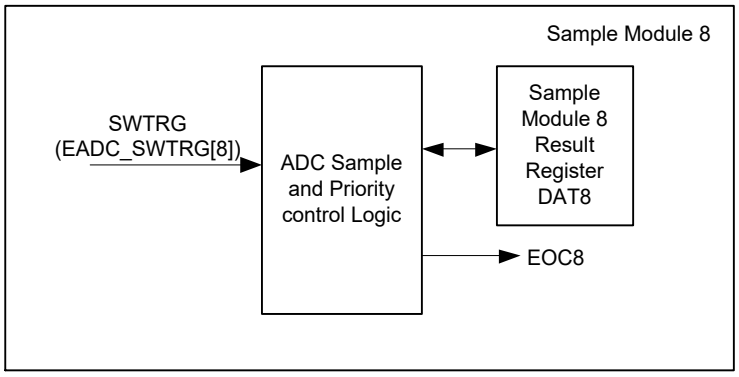


Figure 6.45-4 Sample Module 8 Block Diagram

The EADC conversion trigger sources in sample module 0~8 are listed below:

- Write 1 to SWTRG (EADC_SWTRG[n], n = 0~8)
- External pin EADC0_ST
- Timer0~11 overflow pulse triggers
- ADINT0, ADINT1 EADC interrupt EOC (End of conversion) pulse triggers
- EPWM triggers

The ADINT0 or ADINT1 interrupt pulses are generated whenever the specific sample module EADC EOC (End of conversion) pulse is generated. ADINT0 or ADINT1 interrupt pulse triggers can be fed back to trigger another EADC conversion, and is useful if a continuous scan conversion is needed.

6.45.5.1 EADC Clock Generator

The clock control of EADC is shown as Figure 6.45-5. The EADC peripheral clock source is from PCLK1 clock, the EADC clock frequency is divided by an 8-bit pre-scalar with the following formula:

EADC clock frequency = (PCLK1) / 2*(EADCDIV (CLK_CLKDIV4[3:0])+1)

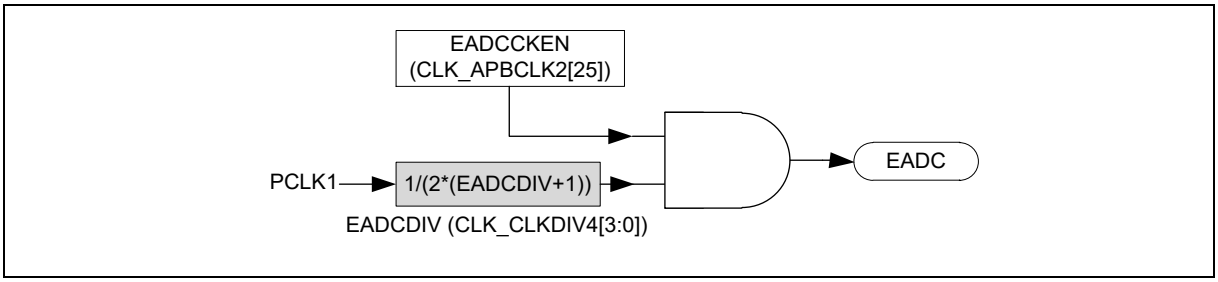


Figure 6.45-5 EADC Clock Control

6.45.5.2 EADC Software Trigger Mode

When an EADC conversion is performed on the sample module specified single channel, the operations are as follows:

1. EADC conversion is started when the SWTRG (EADC_SWTRG[n], n=0~8) is set to 1 by user or other trigger inputs.
2. When EADC conversion is finished, the 12-bit result is stored in the EADC data register EADC_DATn (n=0~8) corresponding to the sample module.
3. On completion of conversion, the ADIFn (EADC_STATUS2[3:0], n=0~3) is set to 1 and EADC interrupt (ADINTn, n=0~3) is requested if the EADCIENn (EADC_CTL[5:2], n=0~3) bit is set to 1.
4. The SWTRG (n=0~8) bit remains 1 during EADC conversion. When EADC conversion ends, the SWTRG (n=0~8) bit is automatically cleared to 0 and the EADC converter will do another pending conversion.

The timing diagram of a conversion cycle is shown in Figure 6.45-6.

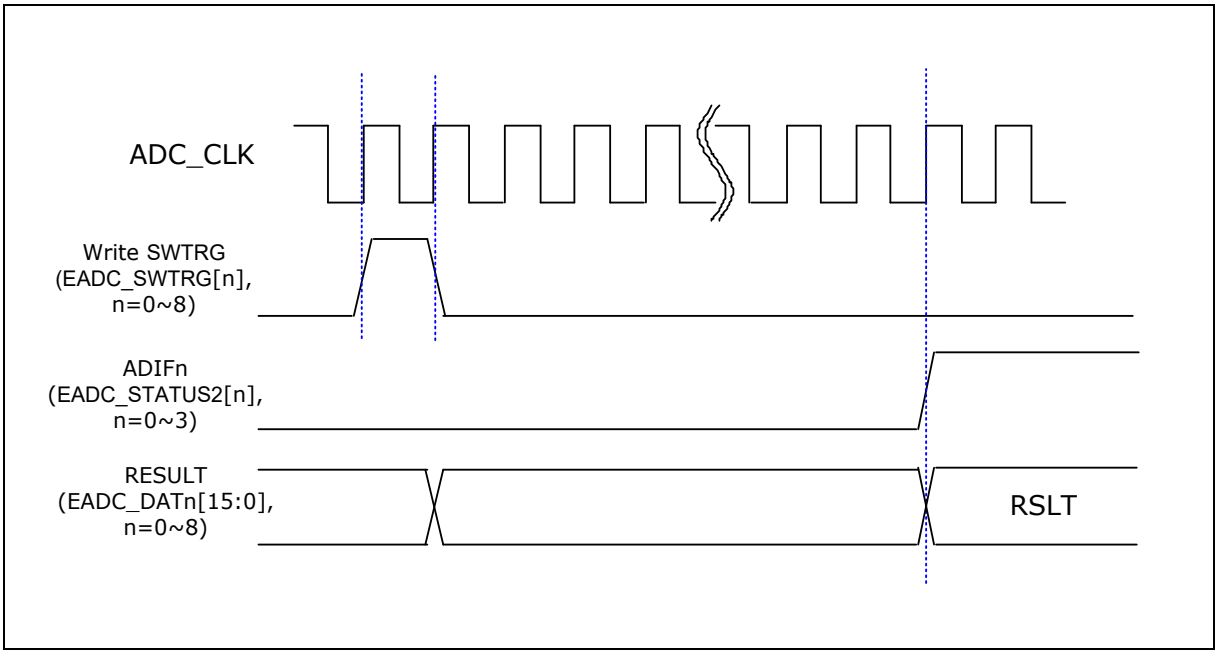


Figure 6.45-6 Example EADC Conversion Timing Diagram, n=0~8

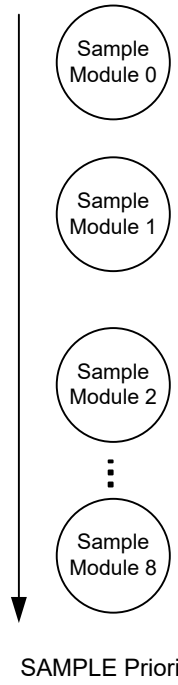
If more than one sample module is enabled to convert analog signal, the sample module specified channel with highest priority is firstly converted and other enabled sample module will be pended. The lower number sample module has higher priority. The sample module 0 is the highest priority and the sample module 8 is the lowest priority.

Note: If the interval between next conversions is more than one EADC clock, EADC will enter standby mode automatically. In other words, the EADC enter standby mode in one EADC clock cycle when the last conversion is finished and signal SOC is low.

6.45.5.3 EADC Conversion Priority

There is a priority group converter for determining the conversion order when multiple sample module trigger flags are set at the same time. Sample module with lower number has higher priority than the higher number sample module. The priority of sample module is shown as Figure 6.45-7. When more than one Sample Module are triggered at the same time, the Sample Module with lower number will start to convert first. The other Sample Module will be in the queue and the corresponding pending flag STPF(EADC_PENDSTS[n], n=0~8) are set to 1 by hardware. After the Sample Module finishes the conversion, STPF(EADC_PENDSTS[n], n=0~8) will be set to 0 automatically. If the Sample Module in the queue is triggered once more, the corresponding Overrun Flag SPOVF(EADC_OVSTS[n], n=0~8) will be set to 1 by hardware.

For example, the Sample Module 0, 2, 3, 5 are triggered simultaneously. The input channel of Sample Module 0 will be converted first. The sample Module 2, 3, 5 will be suspended and STPF (EADC_PENDSTS[2], EADC_PENDSTS [3], EADC_PENDSTS [5]) will be set to 1. If Sample Module 5 is triggered once more in the same time, SPOVF(EADC_OVSTS[5]) will be set to 1.



Sample module 0 has highest priority, sample module 8 has lowest priority for ADC converter

Figure 6.45-7 Sample Module Conversion Priority Arbitrator Diagram

6.45.5.4 Conversion Cycles and Sampling Rate Frequency

There are two kinds of speed modes: high and low speed mode. The relation between resolution and conversion cycles is shown in Table 6.45-1. The maximum sampling rate is up to 4.7 MSPS in high speed mode when EADC clock frequency is up to 80 MHz. The maximum sampling rate is up to 941.1 KSPS in low speed mode when EADC clock frequency is up to 16 MHz. Setting SPEED(EADC_CTL[12]) can select EADC into high speed mode or low speed mode.

Resolution	Minimum Conversion Cycles
12 bit	17 EADC_CLK

Table 6.45-1 Relation between Resolution and Conversion Cycles

Exceeding the limitation of sampling frequency will cause wrong conversion results. The sampling rate frequency can be computed with the following formula:

Sampling rate frequency = (EADC clock frequency) / (conversion cycles+ EADC sampling time extend)

Note: EADC sampling time extend is decided by the option of SELSMPn (EADC_SELSMP[2n+1:2n], n=0~8)

6.45.5.5 Maximum Sampling Frequency Conversion by Software Trigger

If user needs to scan the fast channel at maximum sampling frequency, the conversion needs to be executed by the condition as: multiple sample modules, triggered by software, and triggered repeatedly during the last conversion. An example of continuous scan is as follows:

- Using Module 0~8 to carry out successive conversion. Set CHSEL (EADC_SCTL0~8[3:0]) as one of channel (EADC_CH0~EADC_CH7, V_{BAT}/4). Set SELSMPn (EADC_SELSMP[2n+1:2n], n=0~8) as 0x0 to minimize the sampling time.
- Set SWTRG (EADC_SWTRG[8:0]) as 0x1ff to trigger Module 0~8.
- Wait CURSPL (EADC_STATUS3[4:0]) changes to 0x7 which means Module 0~7 have been executed and Module 8 is in the process. Set SWTRG (EADC_SWTRG[8:0]) as 0xff to trigger Module 0~7 again for next round.
- Wait CURSPL (EADC_STATUS3[4:0]) changes to 0x1, set SWTRG (EADC_SWTRG[8:0]) as 0x100 to trigger Module 8.
- Repeat Step 3~4 to continue the conversion.

6.45.5.6 EADC Sample Module End of Conversion Interrupt Operation

There are 4 EADC interrupts ADINT0~3, and each of these interrupts has its own interrupt vector address and can be configured to set multiple sample module EOC pulse (sample module 0~8 End of conversion pulses) as its interrupt trigger source. Figure 6.45-8 shows the control logic of interrupts. Take ADINT0 as an example, when EADCIEN0 (EADC_CTL[2]) = 1 and SPLIEn (EADC_INTSRC0[n]) = 1 (n=0~8), the specific module EOC (End of conversion) pulses will set flag ADIF0 (EADC_STATUS[0]) as 1 and interrupt (ADINT0) will be asserted either.

The interrupt pulses (ADINT0/1) are generated whenever the specific sample module EADC EOC pulse is generated. It also can be the sample module conversion trigger sources, and user can use it to do the EADC continuous scan conversion.

The example of continuous scan triggered by interrupt is as follows:

1. If EADC sample module 2 EOC2 pulse is selected as ADINT0 interrupt trigger SPLIE2 (EADC_INTSRC0[2]) = 1 and ADINT0 is selected as sample module 0, 1, 2 hardware conversion trigger.
2. Set software trigger SWTRG2 (EADC_SWTRG[2]) to 1 to start a sample module 2 EADC conversion, after the conversion completes, it generates an EOC2 pulse signal and ADINT0 interrupt pulse at end of sample module 2 EADC conversion, ADINT0 interrupt pulse will trigger the sample module 0, 1, 2 to start the EADC conversions.
3. ADINT0 interrupt pulse repeats to trigger sample module 0, 1, 2 EADC conversions automatically.
4. Clear TRGSEL (EADC_SCTL2[20:16]) to 0 to disable sample module 2 ADINT0 interrupt pulse hardware trigger, if needs to stop the continuous scan.

Note: Because the system takes 3 EADC_CLK to trigger the next module by interrupt pulse, the average conversion cycles of continuous scan triggered by interrupt is 19 EADC_CLK.

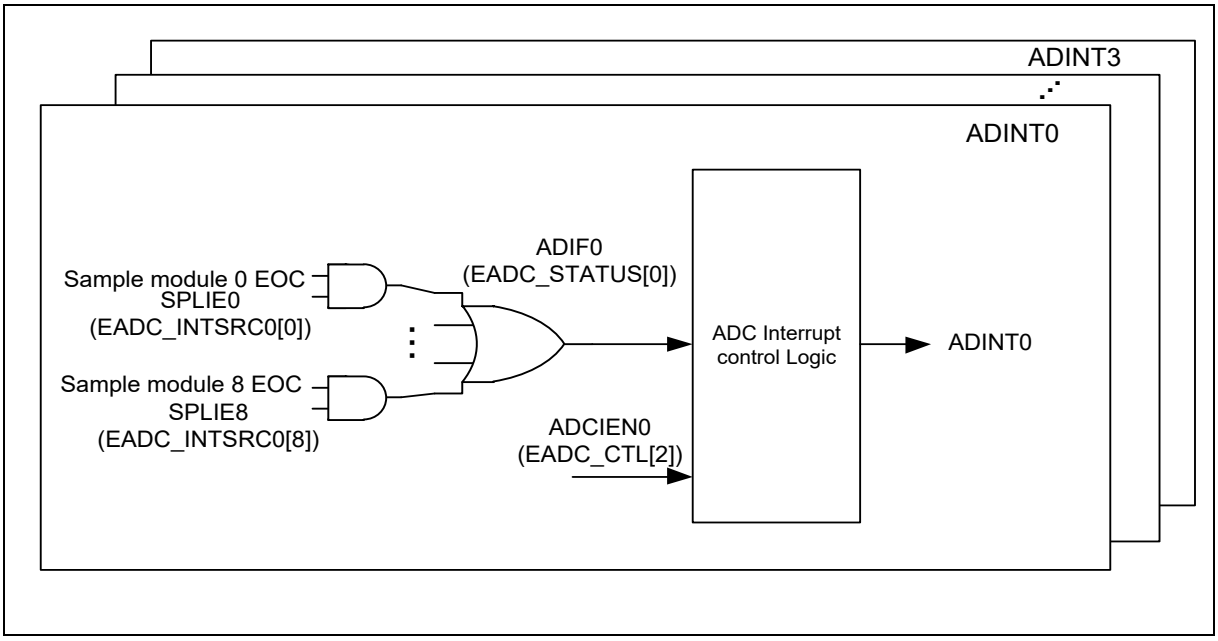


Figure 6.45-8 Specific Sample Module EADC EOC Signal for ADINT0~3 Interrupt

6.45.5.7 EADC Trigger by Timer Trigger and External Pin EADC0_ST

There are 12 Timer trigger source and an external pin EADC0_ST which can configure sample module 0~8 to trigger EADC start when timer overflow occurs.

6.45.5.8 EADC Trigger by External Pin EADC0_ST,Timer Trigger, EPWM Trigger

EADC conversion can be triggered by external pin EADC0_ST request. Setting the TRGSEL (EADC_SCTLn[20:16], n=0~8) to 0x01 is to select external trigger input from the EADC0_ST pin. User can set EXTFEN (EADC_SCTLn[5], n=0~8) and EXTREN (EADC_SCTLn[4], n=0~8) to enable pin EADC0_ST trigger condition is falling or rising edge. There is a de-bounce circuit to detect falling or rising edge. If rising edge trigger condition is selected, the low state must be kept at least 2 PCLK cycles and the following high state must be kept at least 3 PCLK cycles. If falling edge trigger condition is selected, the high state must be kept at least 2 PCLK cycles and the following low state must be kept at least 3 PCLK cycles. Pulse that is shorter than this specification will be ignored. The external trigger timing is shown in Figure 6.45-9.

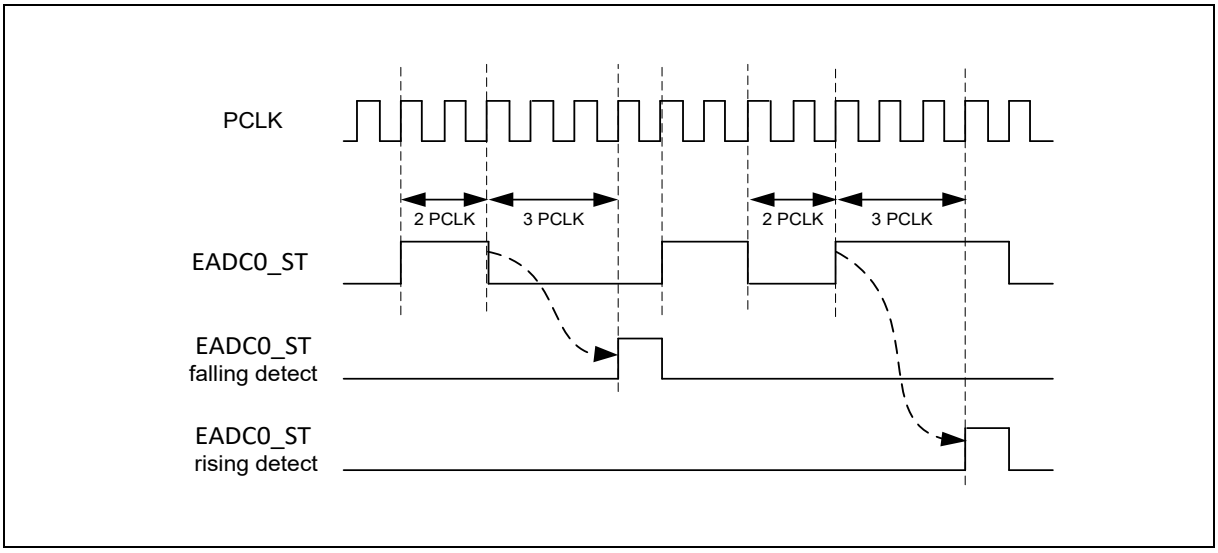


Figure 6.45-9 EADC0_ST De-bounce Timing Diagram

6.45.5.9 EADC Start Synchronous with EPWM Trigger

Besides user start, ADINT0/1 interrupt pulse, external pin EADC0_ST and Timer0~11 overflow pulse to start EADC conversion, this device has new feature to allow EPWM channels to trigger the EADC start. User may configure EPWM trigger types: rising, falling EPWM edge or center point of EPWM (center-aligned mode only) to trigger EADC start. The device also allows user to configure the amount of delay period to EADC start after hardware detected the external trigger. User can configure the trigger delay time by setting TRGDLYCNT (EADC_SCTLn[15:8], n=0~8) and TRGDLYDIV (EADC_SCTLn[7:6], n=0~8). Figure 6.45-10 shows the programmable delay time for EPWM-triggered EADC start conversion.

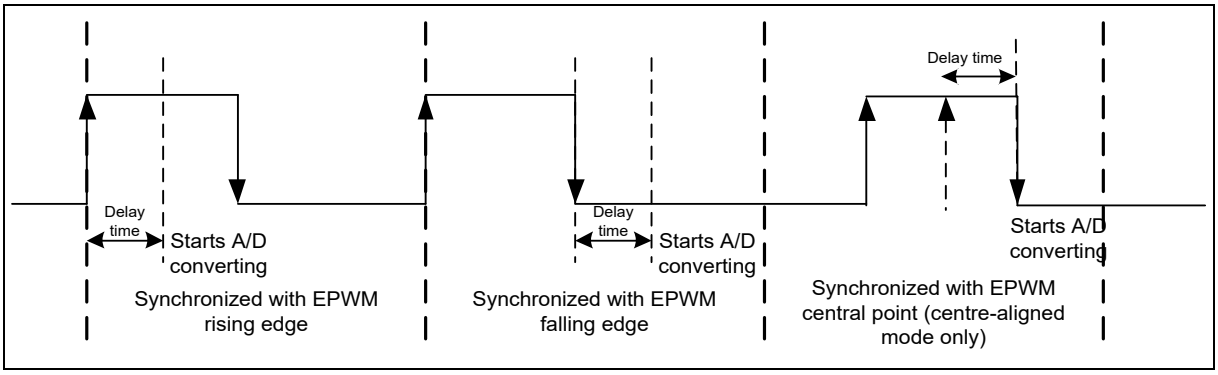


Figure 6.45-10 EPWM-triggered EADC Start Conversion

Figure 6.45-11 shows the programmable delay time for other trigger source.

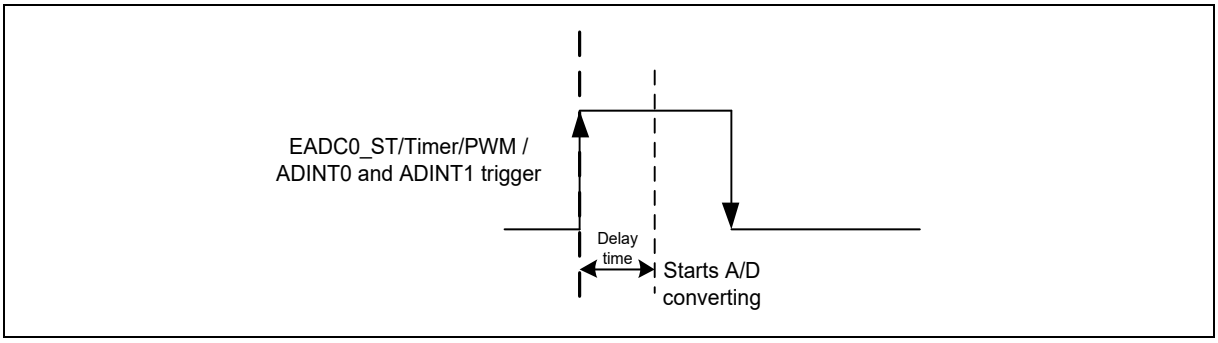


Figure 6.45-11 External triggered EADC Start Conversion

6.45.5.10 Input Sampling and EADC Conversion Time

The EADC converter sample the analog input when EADC conversion start delay time (Td) has passed after SWTRG (EADC_SWTRG[n], n=0~8) is set to 1, then start conversion. Due to EADC clock is generated by PCLK divided by (EADCDIV(2*(CLK_CLKDIV4 [3:0])+1)), the maximum delay time from user write SWTRG to EADC start sampling analog input time is two EADC clock cycles. The start delay time is shown in Figure 6.45-12.

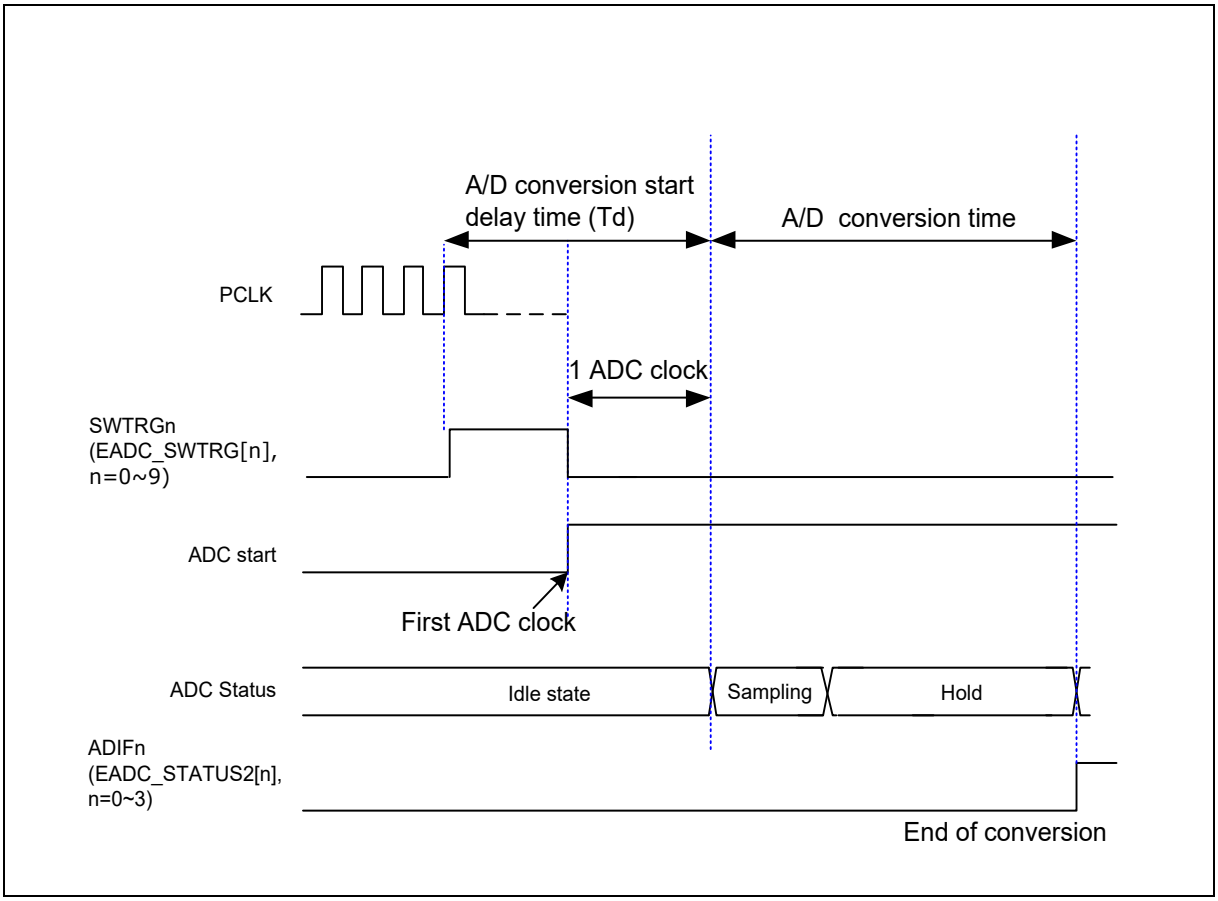
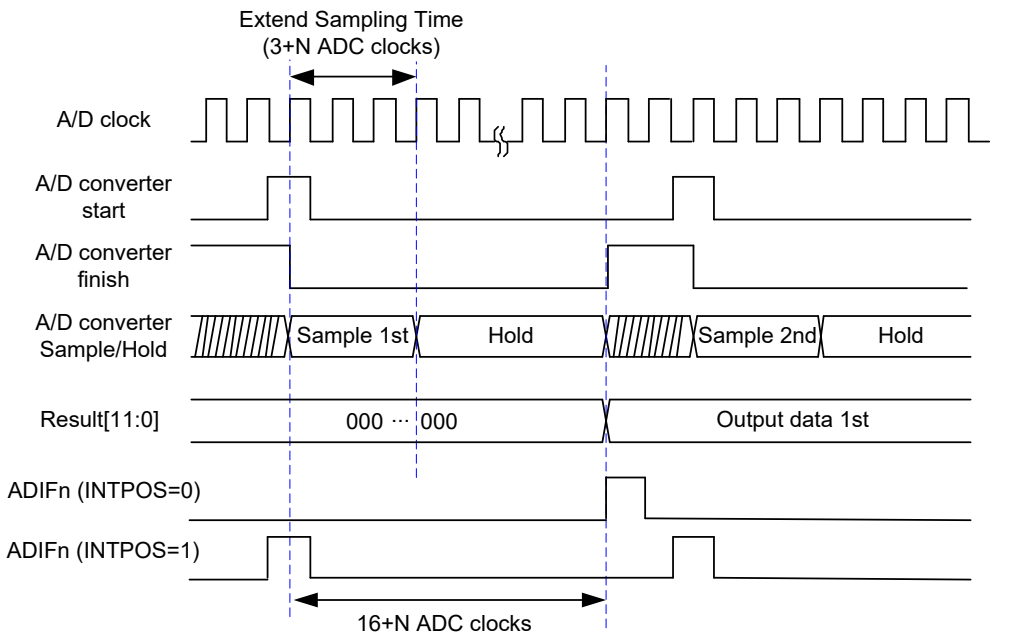


Figure 6.45-12 Conversion Start Delay Timing Diagram

6.45.5.11 EADC Extend Sampling Time

When EADC operates at high EADC clock rate, the sampling time of analog input voltage may not be enough if the analog channel has heavy loading to cause fully charge time is longer. User can set extend sampling time by writing SELSMPn (EADC_SELSMP0n and EADC_SELSMP1) for each sample module. The EADC extend sampling time is present between EADC controller judging which channel to be converted and EADC starting conversion. The extended sampling time is shown in Figure 6.45-13.



Note: N = SEL_SMPn (EADC_SELSMP0 and EADC_SELSMP1) (Extend sampling cycle for each sample module)
ADIFn (EADC_STATUS2[3:0], n=0~3)
INTPOS (EADC_SCTLn[22], n=0~8)

Figure 6.45-13 EADC Extend Sampling Timing Diagram

6.45.5.12 Conversion Result Monitor by Compare Mode

The EADC controller provides four sets of compare registers EADC_CMP0 ~ EADC_CMP3 to monitor a maximum of four specified sample module 0~8 conversion results from EADC conversion module, as shown in Figure 6.45-14. User can select which sample module result to be monitored by set CMPSPL (EADC_CMPn[7:3], n =0~3) and CMPCOND (EADC_CMPn[2], where n =0~3) is used to check conversion result is less than specify value or greater than (equal to) value specified in CMPDAT (EADC_CMPn[27:16], where n =0~3). When the conversion of the sample module specified by CMPSPL is completed, the comparing action will be triggered one time automatically. When the compare result meets the compare condition, the internal compare match counter will increase 1. If the compare result does not meet the condition, the compare match counter will reset to 0. When counter value reach the setting of (CMPMCNT (EADC_CMPn[11:8])+1, where n =0~3) then EADCMPFn (EADC_STATUS2[7:4], where n =0~3) bit will be set to 1, if EADCMPIE (EADC_CMPn[1], n =0~3) is set then an ADINT3 interrupt request is generated. User can use it to monitor the external analog input pin voltage transition. Detailed logics diagram is shown in Figure 6.45-14.

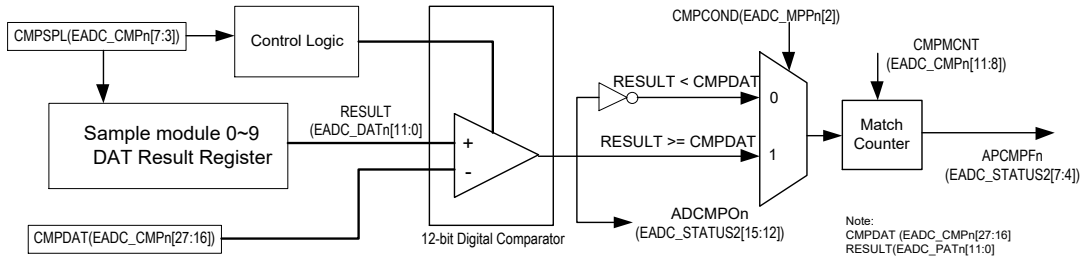


Figure 6.45-14 EADC Conversion Result Monitor Logics Diagram

The EADC controller supports a window compare mode. User can set CMPWEN (EADC_CMP0[15]/ EADC_CMP2[15]) to enable this function. If user enables this function, EADCMPF0 (EADC_STATUS2[4]) will be set when both EADC_CMP0 and EADC_CMP1 compared condition matched. EADCMPF2 (EADC_STATUS2[6]) will be set when both EADC_CMP2 and EADC_CMP3 compared condition matched.

6.45.5.13 Differential Mode

The EADC controller supports analog differential mode. If user enables DIFFEN (EADC_CTL[8]), the differential mode will enable. The pair of analog input channel is as Table 6.45-2.

Differential analog input voltage (Vdiff) = Vplus - Vminus, where Vplus is the analog input; Vminus is the inverted analog input.

Differential Analog Input Paired Channel	EADC Analog Input	
	V_{plus}	V_{minus}
0	EADC_CH0	EADC_CH4
1	EADC_CH1	EADC_CH5
2	EADC_CH2	EADC_CH6
3	EADC_CH3	EADC_CH7

Table 6.45-2 EADC Differential Model Channel Selection

In differential analog input mode, only the positive one of the two corresponding channels needs to be enabled in CHSEL (EADC_SCTLn[3:0]). The conversion result will be placed to the corresponding data register of the enabled channel. The conversion result will store with 2'complement format when DMOF (EADC_CTL[9]) = 1.

6.45.5.14 Double Buffer Mode

The EADC controller supports a double buffer mode in sample module 0~3. If user enable DBMEN (EADC_SCTLn[23], n=0~3), the double buffer mode will enable. In double buffer mode, after first time EADC convert finish, the VALID (EADC_DATn[17], n=0~3) will set to high, but VALID (EADC_DDATn[17], n=0~3) will keep low. And the second time EADC converts finish, VALID (EADC_DDATn[17], n=0~3) will set to high either. Then, user can get the EADC results from EADC_DATn and EADC_DDATn register.

6.45.5.15 PDMA Request

The EADC controller supports PDMA. User can enable PDMATEN (EADC_PDMACTL[8:0]) for each sample mpdule and configure PDMA channel's source address as EADC_CURDAT (EADC_BA+0x4C). After enabling PDMAEN and PDMA channel, if any VALID (EADC_DATn[17],n=0~8) is high, EADC controller will send request to PDMA and PDMA will read EADC_CURDAT to get result. The EADC_CURDAT register is a shadow register of highest priority EADC_DAT register. After PDMA reads EADC_CURDAT register, the VAILD of the shadow EADC_DAT register will be automatically cleared.

6.45.5.16 Interrupt Sources

The EADC converter generates ADIFn (EADC_STATUS2[3:0], n=0~3) at the start of conversion or the end of conversion decide by INTPOS (EADC_SCTLn[22], n=0~15). If EADCIENn (EADC_CTL[5:2], n=0~3) is set then conversion end interrupt request ADINTn (n=0~3) is generated. The controller of interrupts is shown as Figure 6.45-15.

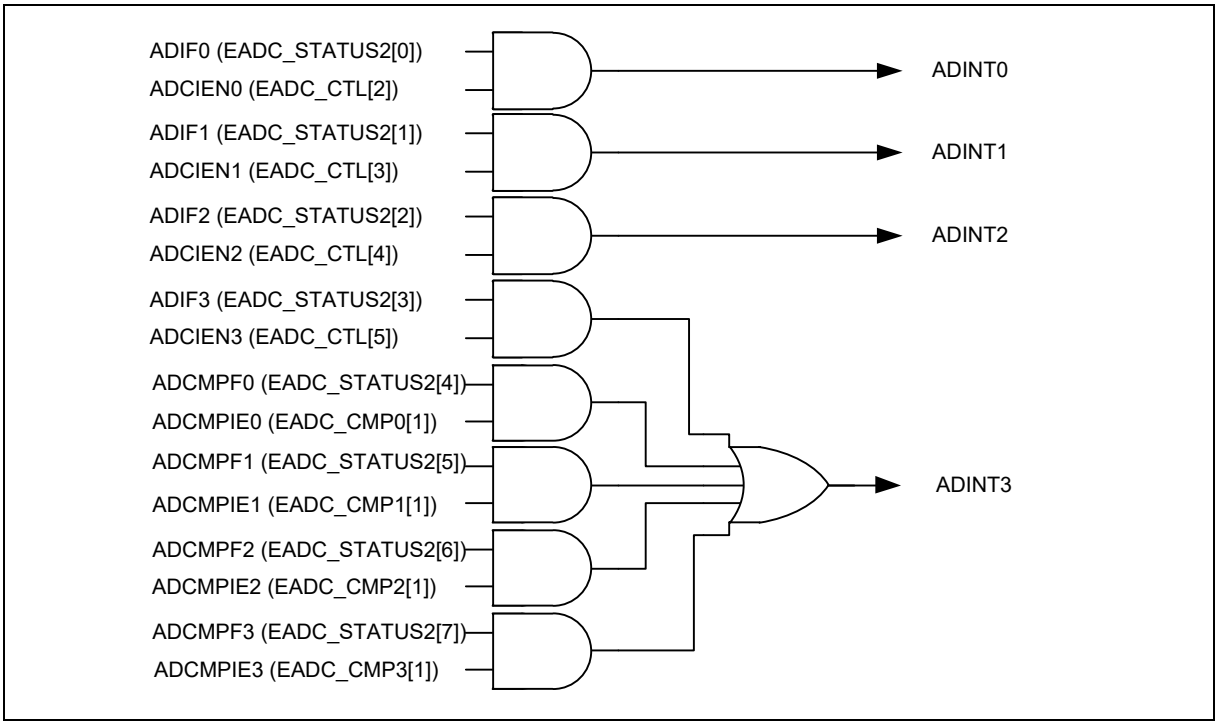


Figure 6.45-15 EADC Controller Interrupts

6.45.5.17 Power Management and Calibration

There are three kinds of power saving modes: Deep Power-down, Power-down, and Standby. User may set PWDMOD (EADC_PWRM[3:2]) to select which power saving mode EADC would enter when EADCEN (EADC_CTL[0]) is set as 0. The difference of these Power-down mode is shown as Table 6.45-3. Because the internal reference will be shut down in Deep Power-down mode, EADC needs to take extra time to resume. The interval of time to resume is set by IREFSUT (EADC_PWRM[27:8]) which must be longer than 1 millisecond.

Power Supplies	Deep Power-Down	Power-Down	Standby
Internal reference	Disable	Enable	Enable

Table 6.45-3 EADC Power Saving Mode

When EADC is activated by setting EADCEN(EADC_CTL[0]) to 1, the start-up sequence will execute automatically. After start-up sequence is finished, PWUPRDY (EADC_PWRM[0]) will be set to 1 by hardware which means ready to convert. EADCEN (EADC_CTL[0]) must be kept at 1 until PWUPRDY (EADC_PWRM[0]) is set to 1 during the start-up sequence. Changing EADCEN (EADC_CTL[0]) arbitrarily at start-up sequence will cause EADC function failure.

The conversion results of EADC will be more accurate with calibration. User may set PWUCALEN (EADC_PWRM[1]) as 1 to carry out calibration at start-up. This bit needs to cooperate with CALSEL (EADC_CALCTL [3]), the configuration of {PWUCALEN, CALSEL} is shown as Table 6.45-4. An example about start-up with calibration is shown as Figure 6.45-16.

PWUCALEN	CALSEL	Configuration
0	0	Start-up without calibration
0	1	Start-up without calibration
1	1	Start-up with calibration

Table 6.45-4 EADC Start-up with Calibration

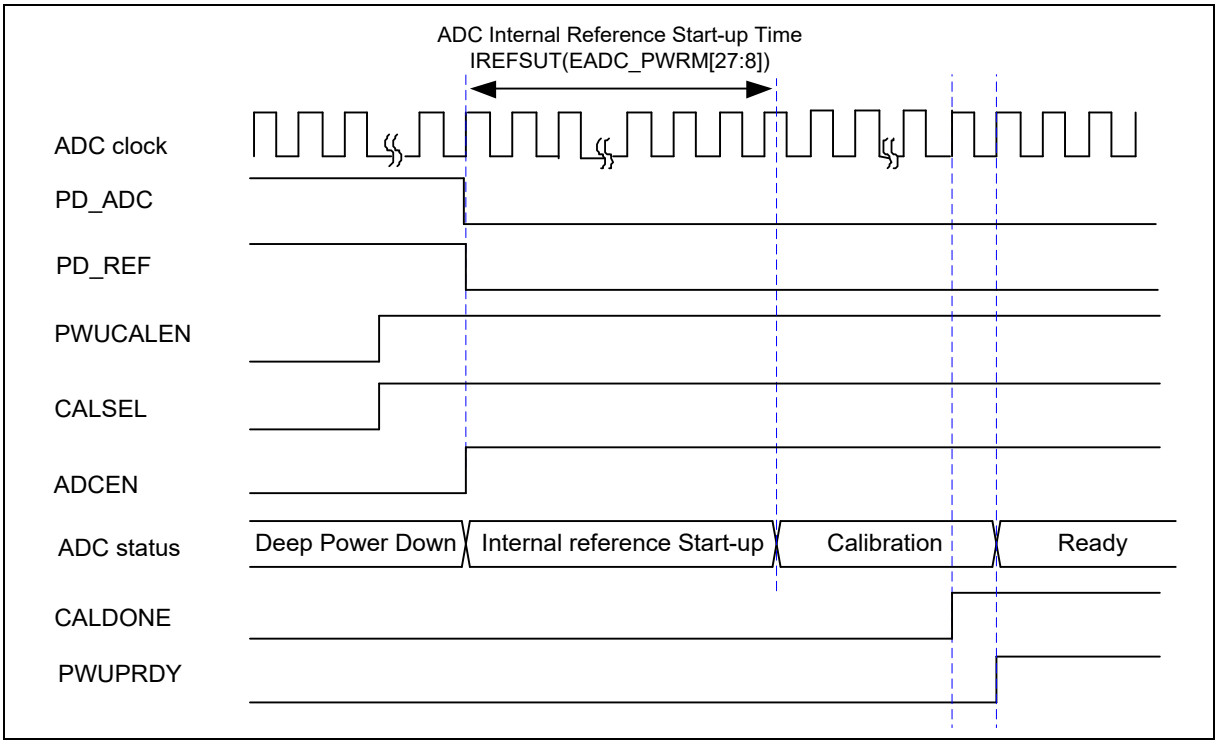


Figure 6.45-16 EADC Start up Sequence with Calibration

To get precise result, user may calibrate again after a few conversion. Set CALSTART (EADC_CALCTL[1]) as 1 could enable calibration again, but this bit needs to work with CALSEL (EADC_CALCTL[3]). Set CALSTART (EADC_CALCTL[1]) as 1 and CALSEL (EADC_CALCTL[3]) as 1 will execute calibration again. The re-calibration sequence should be as follows:

1. Set CALSEL (EADC_CALCTL[3]) as 1 to enable calibration function.
2. Set CALSTART (EADC_CALCTL[1]) as 1 to active calibration.
3. CALDONE (EADC_CALCTL[2]) will be set as 0 by hardware during calibration.

6.45.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
EADC Base Address: **EADC_BA = 0x4043_0000**				
EADC_DAT0	EADC_BA+0x00	R	EADC Data Register 0 for Sample Module 0	0x0000_0000
EADC_DAT1	EADC_BA+0x04	R	EADC Data Register 1 for Sample Module 1	0x0000_0000
EADC_DAT2	EADC_BA+0x08	R	EADC Data Register 2 for Sample Module 2	0x0000_0000
EADC_DAT3	EADC_BA+0x0C	R	EADC Data Register 3 for Sample Module 3	0x0000_0000
EADC_DAT4	EADC_BA+0x10	R	EADC Data Register 4 for Sample Module 4	0x0000_0000
EADC_DAT5	EADC_BA+0x14	R	EADC Data Register 5 for Sample Module 5	0x0000_0000
EADC_DAT6	EADC_BA+0x18	R	EADC Data Register 6 for Sample Module 6	0x0000_0000
EADC_DAT7	EADC_BA+0x1C	R	EADC Data Register 7 for Sample Module 7	0x0000_0000
EADC_DAT8	EADC_BA+0x20	R	EADC Data Register 8 for Sample Module 8	0x0000_0000
EADC_CURDAT	EADC_BA+0x4C	R	EADC PDMA Current Transfer Data Register	0x0000_0000
EADC_CTL	EADC_BA+0x50	R/W	EADC Control Register	0x0000_10C0
EADC_SWTRG	EADC_BA+0x54	W	EADC Sample Module Software Start Register	0x0000_0000
EADC_PENDSTS	EADC_BA+0x58	R/W	EADC Start of Conversion Pending Flag Register	0x0000_0000
EADC_OVSTS	EADC_BA+0x5C	R/W	EADC Sample Module Start of Conversion Overrun Flag Register	0x0000_0000
EADC_SCTL0	EADC_BA+0x80	R/W	EADC Sample Module 0 Control Register	0x0000_0000
EADC_SCTL1	EADC_BA+0x84	R/W	EADC Sample Module 1 Control Register	0x0000_0000
EADC_SCTL2	EADC_BA+0x88	R/W	EADC Sample Module 2 Control Register	0x0000_0000
EADC_SCTL3	EADC_BA+0x8C	R/W	EADC Sample Module 3 Control Register	0x0000_0000
EADC_SCTL4	EADC_BA+0x90	R/W	EADC Sample Module 4 Control Register	0x0000_0000
EADC_SCTL5	EADC_BA+0x94	R/W	EADC Sample Module 5 Control Register	0x0000_0000
EADC_SCTL6	EADC_BA+0x98	R/W	EADC Sample Module 6 Control Register	0x0000_0000
EADC_SCTL7	EADC_BA+0x9C	R/W	EADC Sample Module 7 Control Register	0x0000_0000
EADC_INTSRC0	EADC_BA+0xD0	R/W	EADC Interrupt 0 Source Enable Control Register.	0x0000_0000
EADC_INTSRC1	EADC_BA+0xD4	R/W	EADC Interrupt 1 Source Enable Control Register.	0x0000_0000
EADC_INTSRC2	EADC_BA+0xD8	R/W	EADC Interrupt 2 Source Enable Control Register.	0x0000_0000
EADC_INTSRC3	EADC_BA+0xDC	R/W	EADC Interrupt 3 Source Enable Control Register.	0x0000_0000
EADC_CMP0	EADC_BA+0xE0	R/W	EADC Result Compare Register 0	0x0000_0000

Register	Offset	R/W	Description	Reset Value
EADC Base Address: **EADC_BA = 0x4043_0000**				
EADC_CMP1	EADC_BA+0xE4	R/W	EADC Result Compare Register 1	0x0000_0000
EADC_CMP2	EADC_BA+0xE8	R/W	EADC Result Compare Register 2	0x0000_0000
EADC_CMP3	EADC_BA+0xEC	R/W	EADC Result Compare Register 3	0x0000_0000
EADC_STATUS0	EADC_BA+0xF0	R	EADC Status Register 0	0x0000_0000
EADC_STATUS2	EADC_BA+0xF8	R/W	EADC Status Register 2	0x0011_0000
EADC_STATUS3	EADC_BA+0xFC	R	EADC Status Register 3	0x0000_001F
EADC_DDAT0	EADC_BA+0x100	R	EADC Double Data Register 0 for Sample Module 0	0x0000_0000
EADC_DDAT1	EADC_BA+0x104	R	EADC Double Data Register 1 for Sample Module 1	0x0000_0000
EADC_DDAT2	EADC_BA+0x108	R	EADC Double Data Register 2 for Sample Module 2	0x0000_0000
EADC_DDAT3	EADC_BA+0x10C	R	EADC Double Data Register 3 for Sample Module 3	0x0000_0000
EADC_PWRM	EADC_BA+0x110	R/W	EADC Power Management Register	0x0138_8002
EADC_CALCTL	EADC_BA+0x114	R/W	EADC Calibration Control Register	0x0000_0008
EADC_PDMACTL	EADC_BA+0x130	R/W	EADC PDMA Control Register	0x0000_0000
EADC_SELSMP0	EADC_BA+0x140	R/W	EADC Select Sampling Time Register 0	0x0000_0000
EADC_SELSMP1	EADC_BA+0x144	R/W	EADC Select Sampling Time Register 1	0x0000_0000
EADC_REFADJCTL	EADC_BA+0x150	R/W	EADC Reference Voltage Adjust Control Register	0x0000_0000

6.45.7 Register Description

EADC Data Registers (EADC_DAT0~ EADC_DAT8)

Register	Offset	R/W	Description	Reset Value
EADC_DAT0	EADC_BA+0x00	R	EADC Data Register 0 for Sample Module 0	0x0000_0000
EADC_DAT1	EADC_BA+0x04	R	EADC Data Register 1 for Sample Module 1	0x0000_0000
EADC_DAT2	EADC_BA+0x08	R	EADC Data Register 2 for Sample Module 2	0x0000_0000
EADC_DAT3	EADC_BA+0x0C	R	EADC Data Register 3 for Sample Module 3	0x0000_0000
EADC_DAT4	EADC_BA+0x10	R	EADC Data Register 4 for Sample Module 4	0x0000_0000
EADC_DAT5	EADC_BA+0x14	R	EADC Data Register 5 for Sample Module 5	0x0000_0000
EADC_DAT6	EADC_BA+0x18	R	EADC Data Register 6 for Sample Module 6	0x0000_0000
EADC_DAT7	EADC_BA+0x1C	R	EADC Data Register 7 for Sample Module 7	0x0000_0000
EADC_DAT8	EADC_BA+0x20	R	EADC Data Register 8 for Sample Module 8	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						VALID	OV
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESULT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESULT							

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17]	**VALID**	**Valid Flag** This bit is set to 1 when corresponding sample module channel analog input conversion is completed and cleared by hardware after EADC_DAT register is read. 0 = Data in RESULT[11:0] bits is not valid. 1 = Data in RESULT[11:0] bits is valid.
[16]	**OV**	**Overrun Flag** If converted data in RESULT[11:0] has not been read before new conversion result is loaded to this register, OV is set to 1. 0 = Data in RESULT[11:0] is recent conversion result. 1 = Data in RESULT[11:0] is overwrite. **Note:** It is cleared by hardware after EADC_DAT register is read.
[15:0]	**RESULT**	**EADC Conversion Result** This field contains 12 bits conversion result. When DMOF (EADC_CTL[9]) is set to 0, 12-bit EADC conversion result with unsigned format will be filled in RESULT[11:0] and zero will be filled in RESULT[15:12]. When DMOF (EADC_CTL[9]) set to 1, 12-bit EADC conversion result with 2'complement format will be filled in RESULT[11:0] and signed bits to will be filled in RESULT[15:12].

EADC PDMA Current Transfer Data Register (EADC_CURDAT)

Register	Offset	R/W	Description	Reset Value
EADC_CURDAT	EADC_BA+0x4C	R	EADC PDMA Current Transfer Data Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						CURDAT	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CURDAT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CURDAT							

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17:0]	**CURDAT**	**EADC PDMA Current Transfer Data (Read Only)** This register is a shadow register of EADC_DATn (n=0~8) for PDMA support.

EADC Control Register (EADC_CTL)

Register	Offset	R/W	Description	Reset Value
EADC_CTL	EADC_BA+0x50	R/W	EADC Control Register	0x0000_10C0

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			SPEED	VREFSEL		DMOF	DIFFEN
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RES		EADCIEN3	EADCIEN2	EADCIEN1	EADCIEN0	EADCRST	EADCEN

Bits	Description	
[31:13]	**Reserved**	Reserved.
[12]	**SPEED**	**Speed Mode Select Bit** Control signal for EADC conversion speed 0 = Low speed mode. 1 = High speed mode.
[11:10]	**VREFSEL**	**Internal Voltage Reference Select Bit** EADC have internal voltage reference, user can control this bit to select reference voltage for EADC 00 = 1.6V. 01 = 2.0V. 10 = 2.5V ($AV_{DD}33 \geq 2.8V$). 11 = 3.0V ($AV_{DD}33 \geq 3.3V$). **Note:** Please wait 1ms for the internal reference voltage to stabilize after setting VREFSEL (EADC_CTL[11:10]).
[9]	**DMOF**	**EADC Differential Input Mode Output Format** 0 = EADC conversion result will be filled in RESULT (EADC_DATn[15:0], where n= 0 ~ 8) with unsigned format. 1 = EADC conversion result will be filled in RESULT (EADC_DATn[15:0], where n= 0 ~ 8) with 2'complement format.
[8]	**DIFFEN**	**Differential Analog Input Mode Enable Bit** 0 = Single-end analog input mode. 1 = Differential analog input mode.
[7:6]	**RES**	**Resolution (Read Only)** 11 = 12-bit EADC result will be put at RESULT (EADC_DATn[11:0]).

Bits	Description	
[5]	**EADCIEN3**	**Specific Sample Module EADC ADINT3 Interrupt Enable Bit** The EADC converter generates a conversion end ADIF3 (EADC_STATUS2[3]) upon the end of specific sample module EADC conversion. If EADCIEN3 bit is set then conversion end interrupt request ADINT3 is generated. 0 = Specific sample module EADC ADINT3 interrupt function Disabled. 1 = Specific sample module EADC ADINT3 interrupt function Enabled.
[4]	**EADCIEN2**	**Specific Sample Module EADC ADINT2 Interrupt Enable Bit** The EADC converter generates a conversion end ADIF2 (EADC_STATUS2[2]) upon the end of specific sample module EADC conversion. If EADCIEN2 bit is set then conversion end interrupt request ADINT2 is generated. 0 = Specific sample module EADC ADINT2 interrupt function Disabled. 1 = Specific sample module EADC ADINT2 interrupt function Enabled.
[3]	**EADCIEN1**	**Specific Sample Module EADC ADINT1 Interrupt Enable Bit** The EADC converter generates a conversion end ADIF1 (EADC_STATUS2[1]) upon the end of specific sample module EADC conversion. If EADCIEN1 bit is set then conversion end interrupt request ADINT1 is generated. 0 = Specific sample module EADC ADINT1 interrupt function Disabled. 1 = Specific sample module EADC ADINT1 interrupt function Enabled.
[2]	**EADCIEN0**	**Specific Sample Module EADC ADINT0 Interrupt Enable Bit** The EADC converter generates a conversion end ADIF0 (EADC_STATUS2[0]) upon the end of specific sample module EADC conversion. If EADCIEN0 bit is set then conversion end interrupt request ADINT0 is generated. 0 = Specific sample module EADC ADINT0 interrupt function Disabled. 1 = Specific sample module EADC ADINT0 interrupt function Enabled.
[1]	**EADCRST**	**EADC Converter Control Circuits Reset** 0 = No effect. 1 = Cause EADC control circuits reset to initial state, but not change the EADC registers value. **Note: EADC**RST bit remains 1 during EADC reset, when EADC reset end, the EADCRST bit is automatically cleared to 0.
[0]	**EADCEN**	**EADC Converter Enable Bit** 0 = EADC Disabled. 1 = EADC Enabled. **Note:** Before starting EADC conversion function, this bit should be set to 1. Clear it to 0 to disable EADC converter analog circuit power consumption.

EADC Sample Module Software Start Register (EADC_SWTRG)

Register	Offset	R/W	Description	Reset Value
EADC_SWTRG	EADC_BA+0x54	W	EADC Sample Module Software Start Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							SWTRG
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SWTRG							

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8:0]	**SWTRG**	**EADC Sample Module 0~8 Software Force to Start EADC Conversion** 0 = No effect. 1 = Cause an EADC conversion when the priority is given to sample module. **Note:** After writing this register to start EADC conversion, the EADC_PENDSTS register will show which sample module will conversion. If user want to disable the conversion of the sample module, user can write EADC_PENDSTS register to clear it.

EADC Sample Module Start of Conversion Pending Flag Register (EADC_PENDSTS)

Register	Offset	R/W	Description	Reset Value
EADC_PENDSTS	EADC_BA+0x58	R/W	EADC Start of Conversion Pending Flag Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							STPF
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
STPF							

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8:0]	**STPF**	**EADC Sample Module 0~8 Start of Conversion Pending Flag** **Read Operation:** 0 = There is no pending conversion for sample module. 1 = Sample module EADC start of conversion is pending. **Write Operation:** 1 = Clear pending flag and cancel the conversion for sample module. **Note:** This bit remains 1 during pending state. When the respective EADC conversion is ended, the STPFn (n=0~8) bit is automatically cleared to 0.

EADC Sample Module Overrun Flag Register (EADC_OVSTS)

Register	Offset	R/W	Description	Reset Value
EADC_OVSTS	EADC_BA+0x5C	R/W	EADC Sample Module Start of Conversion Overrun Flag Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							SPOVF
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SPOVF							

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8:0]	**SPOVF**	**EADC SAMPLE0~8 Overrun Flag** 0 = No sample module event overrun. 1 = Indicates a new sample module event is generated while an old one event is pending. **Note:** This bit is cleared by writing 1 to it.

EADC Sample Module 0~3 Control Registers (EADC_SCTL0~EADC_SCTL3)

Register	Offset	R/W	Description	Reset Value
EADC_SCTL0	EADC_BA+0x80	R/W	EADC Sample Module 0 Control Register	0x0000_0000
EADC_SCTL1	EADC_BA+0x84	R/W	EADC Sample Module 1 Control Register	0x0000_0000
EADC_SCTL2	EADC_BA+0x88	R/W	EADC Sample Module 2 Control Register	0x0000_0000
EADC_SCTL3	EADC_BA+0x8C	R/W	EADC Sample Module 3 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
DBMEN	INTPOS	TRGSEL					
15	14	13	12	11	10	9	8
TRGDLYCNT							
7	6	5	4	3	2	1	0
TRGDLYDIV		EXTFEN	EXTREN	CHSEL			

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23]	**DBMEN**	**Double Buffer Mode Enable Bit** 0 = Sample has one sample result register (default). 1 = Sample has two sample result registers.
[22]	**INTPOS**	**Interrupt Flag Position Select** 0 = Set ADIFn (EADC_STATUS2[n], n=0~3) at EADC end of conversion. 1 = Set ADIFn (EADC_STATUS2[n], n=0~3) at EADC start of conversion.

Bits	Description	
[21:16]	**TRGSEL**	**EADC Sample Module Start of Conversion Trigger Source Selection** 0H = Disable trigger. 1H = External trigger from EADC0_ST pin input. 2H = EADC ADINT0 interrupt EOC (End of conversion) pulse trigger. 3H = EADC ADINT1 interrupt EOC (End of conversion) pulse trigger. 4H = Timer0 overflow pulse trigger. 5H = Timer1 overflow pulse trigger. 6H = Timer2 overflow pulse trigger. 7H = Timer3 overflow pulse trigger. 8H = Timer4 overflow pulse trigger. 9H = Timer5 overflow pulse trigger. AH = Timer6 overflow pulse trigger. BH = Timer7 overflow pulse trigger. CH = Timer8 overflow pulse trigger. DH = Timer9 overflow pulse trigger. EH = Timer10 overflow pulse trigger. FH = Timer11 overflow pulse trigger. 10H = EPWM0TG0. 11H = EPWM0TG1. 12H = EPWM0TG2. 13H = EPWM0TG3. 14H = EPWM0TG4. 15H = EPWM0TG5. 16H = EPWM1TG0. 17H = EPWM1TG1. 18H = EPWM1TG2. 19H = EPWM1TG3. 1AH = EPWM1TG4. 1BH = EPWM1TG5. 1CH = EPWM2TG0. 1DH = EPWM2TG1. 1EH = EPWM2TG2. 1FH = EPWM2TG3. 20H = EPWM2TG4. 21H = EPWM2TG5. other = Reserved.
[15:8]	**TRGDLYCNT**	**EADC Sample Module Start of Conversion Trigger Delay Time** Trigger delay time = TRGDLYCNT x EADC_CLK period x n (n=1,2,4,16 from TRGDLYDIV setting). **Note:** If TRGDLYCNT is set to 1, trigger delay time is actually the same as TRGDLYCNT is set to 2 for hardware operation.
[7:6]	**TRGDLYDIV**	**EADC Sample Module Start of Conversion Trigger Delay Clock Divider Selection** Trigger delay clock frequency: 00 = EADC_CLK/1. 01 = EADC_CLK/2. 10 = EADC_CLK/4. 11 = EADC_CLK/16.

Bits	Description	
[5]	**EXTFEN**	**EADC External Trigger Falling Edge Enable Bit** 0 = Falling edge Disabled when EADC selects EADC0_ST as trigger source. 1 = Falling edge Enabled when EADC selects EADC0_ST as trigger source.
[4]	**EXTREN**	**EADC External Trigger Rising Edge Enable Bit** 0 = Rising edge Disabled when EADC selects EADC0_ST as trigger source. 1 = Rising edge Enabled when EADC selects EADC0_ST as trigger source.
[3:0]	**CHSEL**	**EADC Sample Module Channel Selection** 00H = EADC_CH0. 01H = EADC_CH1. 02H = EADC_CH2. 03H = EADC_CH3. 04H = EADC_CH4. 05H = EADC_CH5. 06H = EADC_CH6. 07H = EADC_CH7. 08H = V_{BAT}/4.

EADC Sample Module 4~7 Control Registers (EADC_SCTL4~EADC_SCTL7)

Register	Offset	R/W	Description	Reset Value
EADC_SCTL4	EADC_BA+0x90	R/W	EADC Sample Module 4 Control Register	0x0000_0000
EADC_SCTL5	EADC_BA+0x94	R/W	EADC Sample Module 5 Control Register	0x0000_0000
EADC_SCTL6	EADC_BA+0x98	R/W	EADC Sample Module 6 Control Register	0x0000_0000
EADC_SCTL7	EADC_BA+0x9C	R/W	EADC Sample Module 7 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved	INTPOS	TRGSEL					
15	14	13	12	11	10	9	8
TRGDLYCNT							
7	6	5	4	3	2	1	0
TRGDLYDIV		EXTFEN	EXTREN	CHSEL			

Bits	Description	
[31:23]	**Reserved**	Reserved.
[22]	**INTPOS**	**Interrupt Flag Position Select** 0 = Set ADIFn (EADC_STATUS2[n], n=0~3) at EADC end of conversion. 1 = Set ADIFn (EADC_STATUS2[n], n=0~3) at EADC start of conversion.

Bits	Description	
[21:16]	**TRGSEL**	**EADC Sample Module Start of Conversion Trigger Source Selection** 0H = Disable trigger. 1H = External trigger from EADC0_ST pin input. 2H = EADC ADINT0 interrupt EOC (End of conversion) pulse trigger. 3H = EADC ADINT1 interrupt EOC (End of conversion) pulse trigger. 4H = Timer0 overflow pulse trigger. 5H = Timer1 overflow pulse trigger. 6H = Timer2 overflow pulse trigger. 7H = Timer3 overflow pulse trigger. 8H = Timer4 overflow pulse trigger. 9H = Timer5 overflow pulse trigger. AH = Timer6 overflow pulse trigger. BH = Timer7 overflow pulse trigger. CH = Timer8 overflow pulse trigger. DH = Timer9 overflow pulse trigger. EH = Timer10 overflow pulse trigger. FH = Timer11 overflow pulse trigger. 10H = EPWM0TG0. 11H = EPWM0TG1. 12H = EPWM0TG2. 13H = EPWM0TG3. 14H = EPWM0TG4. 15H = EPWM0TG5. 16H = EPWM1TG0. 17H = EPWM1TG1. 18H = EPWM1TG2. 19H = EPWM1TG3. 1AH = EPWM1TG4. 1BH = EPWM1TG5. 1CH = EPWM2TG0. 1DH = EPWM2TG1. 1EH = EPWM2TG2. 1FH = EPWM2TG3. 20H = EPWM2TG4. 21H = EPWM2TG5. other = Reserved.
[15:8]	**TRGDLYCNT**	**EADC Sample Module Start of Conversion Trigger Delay Time** Trigger delay time = TRGDLYCNT x EADC_CLK period x n (n=1,2,4,16 from TRGDLYDIV setting). **Note:** If TRGDLYCNT is set to 1, trigger delay time is actually the same as TRGDLYCNT is set to 2 for hardware operation.
[7:6]	**TRGDLYDIV**	**EADC Sample Module Start of Conversion Trigger Delay Clock Divider Selection** Trigger delay clock frequency: 00 = EADC_CLK/1. 01 = EADC_CLK/2. 10 = EADC_CLK/4. 11 = EADC_CLK/16.

Bits	Description	
[5]	**EXTFEN**	**EADC External Trigger Falling Edge Enable Bit** 0 = Falling edge Disabled when EADC selects EADC0_ST as trigger source. 1 = Falling edge Enabled when EADC selects EADC0_ST as trigger source.
[4]	**EXTREN**	**EADC External Trigger Rising Edge Enable Bit** 0 = Rising edge Disabled when EADC selects EADC0_ST as trigger source. 1 = Rising edge Enabled when EADC selects EADC0_ST as trigger source.
[3:0]	**CHSEL**	**EADC Sample Module Channel Selection** 00H = EADC_CH0. 01H = EADC_CH1. 02H = EADC_CH2.. 03H = EADC_CH3. 04H = EADC_CH4. 05H = EADC_CH5. 06H = EADC_CH6. 07H = EADC_CH7. 08H = V_{BAT}/4.

EADC Interrupt Source Enable Control Registers (EADC_INTSRC0~EADC_INTSRC3)

Register	Offset	R/W	Description	Reset Value
EADC_INTSRC0	EADC_BA+0xD0	R/W	EADC Interrupt 0 Source Enable Control Register.	0x0000_0000
EADC_INTSRC1	EADC_BA+0xD4	R/W	EADC Interrupt 1 Source Enable Control Register.	0x0000_0000
EADC_INTSRC2	EADC_BA+0xD8	R/W	EADC Interrupt 2 Source Enable Control Register.	0x0000_0000
EADC_INTSRC3	EADC_BA+0xDC	R/W	EADC Interrupt 3 Source Enable Control Register.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							SPLIE8
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SPLIE7	SPLIE6	SPLIE5	SPLIE4	SPLIE3	SPLIE2	SPLIE1	SPLIE0

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8]	**SPLIE8**	**Sample Module 8 Interrupt Enable Bit** 0 = Sample Module 8 interrupt Disabled. 1 = Sample Module 8 interrupt Enabled.
[7]	**SPLIE7**	**Sample Module 7 Interrupt Enable Bit** 0 = Sample Module 7 interrupt Disabled. 1 = Sample Module 7 interrupt Enabled.
[6]	**SPLIE6**	**Sample Module 6 Interrupt Enable Bit** 0 = Sample Module 6 interrupt Disabled. 1 = Sample Module 6 interrupt Enabled.
[5]	**SPLIE5**	**Sample Module 5 Interrupt Enable Bit** 0 = Sample Module 5 interrupt Disabled. 1 = Sample Module 5 interrupt Enabled.
[4]	**SPLIE4**	**Sample Module 4 Interrupt Enable Bit** 0 = Sample Module 4 interrupt Disabled. 1 = Sample Module 4 interrupt Enabled.
[3]	**SPLIE3**	**Sample Module 3 Interrupt Enable Bit** 0 = Sample Module 3 interrupt Disabled. 1 = Sample Module 3 interrupt Enabled.
[2]	**SPLIE2**	**Sample Module 2 Interrupt Enable Bit** 0 = Sample Module 2 interrupt Disabled. 1 = Sample Module 2 interrupt Enabled.

Bits	Description	
[1]	**SPLIE1**	**Sample Module 1 Interrupt Enable Bit** 0 = Sample Module 1 interrupt Disabled. 1 = Sample Module 1 interrupt Enabled.
[0]	**SPLIE0**	**Sample Module 0 Interrupt Enable Bit** 0 = Sample Module 0 interrupt Disabled. 1 = Sample Module 0 interrupt Enabled.

EADC Result Compare Register 0/1/2/3 (EADC_CMP0/1/2/3)

Register	Offset	R/W	Description	Reset Value
EADC_CMP0	EADC_BA+0xE0	R/W	EADC Result Compare Register 0	0x0000_0000
EADC_CMP1	EADC_BA+0xE4	R/W	EADC Result Compare Register 1	0x0000_0000
EADC_CMP2	EADC_BA+0xE8	R/W	EADC Result Compare Register 2	0x0000_0000
EADC_CMP3	EADC_BA+0xEC	R/W	EADC Result Compare Register 3	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				CMPDAT			
23	22	21	20	19	18	17	16
CMPDAT							
15	14	13	12	11	10	9	8
CMPWEN	Reserved			CMPMCNT			
7	6	5	4	3	2	1	0
Reserved	CMPSPL				CMPCOND	EADCMPIE	EADCMPEN

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:16]	**CMPDAT**	**Comparison Data** The 12 bits data is used to compare with conversion result of specified sample module. User can use it to monitor the external analog input pin voltage transition without imposing a load on software.
[15]	**CMPWEN**	**Compare Window Mode Enable Bit** 0 = EADCMPF0 (EADC_STATUS2[4]) will be set when EADC_CMP0 compared condition matched. EADCMPF2 (EADC_STATUS2[6]) will be set when EADC_CMP2 compared condition matched 1 = EADCMPF0 (EADC_STATUS2[4]) will be set when both EADC_CMP0 and EADC_CMP1 compared condition matched. EADCMPF2 (EADC_STATUS2[6]) will be set when both EADC_CMP2 and EADC_CMP3 compared condition matched. **Note:** This bit is only present in EADC_CMP0 and EADC_CMP2 register.
[14:12]	**Reserved**	Reserved.
[11:8]	**CMPMCNT**	**Compare Match Count** When the specified EADC sample module analog conversion result matches the compare condition defined by CMPCOND (EADC_CMPn[2], n=0~3), the internal match counter will increase 1. If the compare result does not meet the compare condition, the internal compare match counter will reset to 0. When the internal counter reaches the value to (CMPMCNT +1), the EADCMPFn (EADC_STATUS2[7:4], n=0~3) will be set.
[7]	**Reserved**	Reserved.

Bits	Description	
[6:3]	**CMPSPL**	**Compare Sample Module Selection** 00000 = Sample Module 0 conversion result EADC_DAT0 is selected to be compared. 00001 = Sample Module 1 conversion result EADC_DAT1 is selected to be compared. 00010 = Sample Module 2 conversion result EADC_DAT2 is selected to be compared. 00011 = Sample Module 3 conversion result EADC_DAT3 is selected to be compared. 00100 = Sample Module 4 conversion result EADC_DAT4 is selected to be compared. 00101 = Sample Module 5 conversion result EADC_DAT5 is selected to be compared. 00110 = Sample Module 6 conversion result EADC_DAT6 is selected to be compared. 00111 = Sample Module 7 conversion result EADC_DAT7 is selected to be compared. 01000 = Sample Module 8 conversion result EADC_DAT8 is selected to be compared.
[2]	**CMPCOND**	**Compare Condition** 0= Set the compare condition as that when a 12-bit EADC conversion result is less than the 12-bit CMPDAT (EADC_CMPn [27:16]), the internal match counter will increase one. 1= Set the compare condition as that when a 12-bit EADC conversion result is greater or equal to the 12-bit CMPDAT (EADC_CMPn [27:16]), the internal match counter will increase one. **Note:** When the internal counter reaches the value to (CMPMCNT (EADC_CMPn[11:8], n=0~3) +1), the CMPF bit will be set.
[1]	**EADCMPIE**	**EADC Result Compare Interrupt Enable Bit** 0 = Compare function interrupt Disabled. 1 = Compare function interrupt Enabled. If the compare function is enabled and the compare condition matches the setting of CMPCOND (EADC_CMPn[2], n=0~3) and CMPMCNT (EADC_CMPn[11:8], n=0~3), EADCMPFn (EADC_STATUS2[7:4], n=0~3) will be asserted, in the meanwhile, if EADCMPIE is set to 1, a compare interrupt request is generated.
[0]	**EADCMPEN**	**EADC Result Compare Enable Bit** 0 = Compare Disabled. 1 = Compare Enabled. Set this bit to 1 to enable compare CMPDAT (EADC_CMPn[27:16], n=0~3) with specified sample module conversion result when converted data is loaded into EADC_DAT register.

EADC Status Register 0 (EADC_STATUS0)

Register	Offset	R/W	Description	Reset Value
EADC_STATUS0	EADC_BA+0xF0	R	EADC Status Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							OV
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
OV							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							VALID
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VALID							

Bits	Description	
[31:25]	**Reserved**	Reserved.
[24:16]	**OV**	**EADC_DAT0~8 Overrun Flag** It is a mirror to OV bit in sample module EADC result data register EADC_DATn. (n=0~8).
[15:9]	**Reserved**	Reserved.
[8:0]	**VALID**	**EADC_DAT0~8 Data Valid Flag** It is a mirror of VALID bit in sample module EADC result data register EADC_DATn. (n=0~8).

EADC Status Register 2 (EADC_STATUS2)

Register	Offset	R/W	Description	Reset Value
EADC_STATUS2	EADC_BA+0xF8	R/W	EADC Status Register 2	0x0011_0000

31	30	29	28	27	26	25	24
Reserved				AOV	AVALID	STOVF	ADOVIF
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
BUSY	Reserved	Reserved	CHANNEL				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
EADCMPO3	EADCMPO2	EADCMPO1	EADCMPO0	ADOVIF3	ADOVIF2	ADOVIF1	ADOVIF0
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EADCMPF3	EADCMPF2	EADCMPF1	EADCMPF0	ADIF3	ADIF2	ADIF1	ADIF0

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27]	**AOV**	**All Sample Module EADC Result Data Register Overrun Flags Check (Read Only)** n=0~8. 0 = None of sample module data register overrun flag OV (EADC_DATn[16]) is set to 1. 1 = Any one of sample module data register overrun flag OV (EADC_DATn[16]) is set to 1. **Note:** This bit will keep 1 when any OV Flag is equal to 1.
[26]	**AVALID**	**All Sample Module EADC Result Data Register EADC_DAT Data Valid Flag Check (Read Only)** n=0~8. 0 = None of sample module data register valid flag VALID (EADC_DATn[17]) is set to 1. 1 = Any one of sample module data register valid flag VALID (EADC_DATn[17]) is set to 1. **Note:** This bit will keep 1 when any VALID Flag is equal to 1.
[25]	**STOVF**	**All EADC Sample Module Start of Conversion Overrun Flags Check (Read Only)** n=0~8. 0 = None of sample module event overrun flag SPOVF (EADC_OVSTS[n]) is set to 1. 1 = Any one of sample module event overrun flag SPOVF (EADC_OVSTS[n]) is set to 1. **Note:** This bit will keep 1 when any SPOVF Flag is equal to 1.
[24]	**ADOVIF**	**All EADC Interrupt Flag Overrun Bits Check (Read Only)** n=0~3. 0 = None of ADINT interrupt flag ADOVIFn (EADC_STATUS2[11:8]) is overwritten to 1. 1 = Any one of ADINT interrupt flag ADOVIFn (EADC_STATUS2[11:8]) is overwritten to 1. **Note:** This bit will keep 1 when any ADOVIFn Flag is equal to 1.
[23]	**BUSY**	**Busy/Idle (Read Only)** 0 = EADC is in idle state. 1 = EADC is busy at conversion.
[22:21]	**Reserved**	Reserved.

Bits	Description	
[20:16]	**CHANNEL**	**Current Conversion Channel (Read Only)** This filed reflects EADC current conversion channel when BUSY=1. 00H = EADC_CH0. 01H = EADC_CH1. 02H = EADC_CH2. 03H = EADC_CH3. 04H = EADC_CH4. 05H = EADC_CH5. 06H = EADC_CH6. 07H = EADC_CH7. 08H = V_{BAT}/4.
[15]	**EADCMPO3**	**EADC Compare 3 Output Status (Read Only)** The 12 bits compare3 data CMPDAT3 (EADC_CMP3[27:16]) is used to compare with conversion result of specified sample module. User can use it to monitor the external analog input pin voltage status. 0 = Conversion result in EADC_DAT is less than CMPDAT3 setting. 1 = Conversion result in EADC_DAT is greater than or equal to CMPDAT3 setting.
[14]	**EADCMPO2**	**EADC Compare 2 Output Status (Read Only)** The 12 bits compare2 data CMPDAT2 (EADC_CMP2[27:16]) is used to compare with conversion result of specified sample module. User can use it to monitor the external analog input pin voltage status. 0 = Conversion result in EADC_DAT is less than CMPDAT2 setting. 1 = Conversion result in EADC_DAT is greater than or equal to CMPDAT2 setting.
[13]	**EADCMPO1**	**EADC Compare 1 Output Status (Read Only)** The 12 bits compare1 data CMPDAT1 (EADC_CMP1[27:16]) is used to compare with conversion result of specified sample module. User can use it to monitor the external analog input pin voltage status. 0 = Conversion result in EADC_DAT is less than CMPDAT1 setting. 1 = Conversion result in EADC_DAT is greater than or equal to CMPDAT1 setting.
[12]	**EADCMPO0**	**EADC Compare 0 Output Status (Read Only)** The 12 bits compare0 data CMPDAT0 (EADC_CMP0[27:16]) is used to compare with conversion result of specified sample module. User can use it to monitor the external analog input pin voltage status. 0 = Conversion result in EADC_DAT is less than CMPDAT0 setting. 1 = Conversion result in EADC_DAT is greater than or equal to CMPDAT0 setting.
[11]	**ADOVIF3**	**EADC ADINT3 Interrupt Flag Overrun** 0 = ADINT3 interrupt flag is not overwritten to 1. 1 = ADINT3 interrupt flag is overwritten to 1. **Note:** This bit is cleared by writing 1 to it.
[10]	**ADOVIF2**	**EADC ADINT2 Interrupt Flag Overrun** 0 = ADINT2 interrupt flag is not overwritten to 1. 1 = ADINT2 interrupt flag is overwritten to 1. **Note:** This bit is cleared by writing 1 to it.
[9]	**ADOVIF1**	**EADC ADINT1 Interrupt Flag Overrun** 0 = ADINT1 interrupt flag is not overwritten to 1. 1 = ADINT1 interrupt flag is overwritten to 1. **Note:** This bit is cleared by writing 1 to it.

Bits	Description	
[8]	**ADOVIF0**	**EADC ADINT0 Interrupt Flag Overrun** 0 = ADINT0 interrupt flag is not overwritten to 1. 1 = ADINT0 interrupt flag is overwritten to 1. **Note:** This bit is cleared by writing 1 to it.
[7]	**EADCMPF3**	**EADC Compare 3 Flag** When the specific sample module EADC conversion result meets setting condition in EADC_CMP3 then this bit is set to 1. 0 = Conversion result in EADC_DAT does not meet EADC_CMP3 register setting. 1 = Conversion result in EADC_DAT meets EADC_CMP3 register setting. **Note:** This bit is cleared by writing 1 to it.
[6]	**EADCMPF2**	**EADC Compare 2 Flag** When the specific sample module EADC conversion result meets setting condition in EADC_CMP2 then this bit is set to 1. 0 = Conversion result in EADC_DAT does not meet EADC_CMP2 register setting. 1 = Conversion result in EADC_DAT meets EADC_CMP2 register setting. **Note:** This bit is cleared by writing 1 to it.
[5]	**EADCMPF1**	**EADC Compare 1 Flag** When the specific sample module EADC conversion result meets setting condition in EADC_CMP1 then this bit is set to 1. 0 = Conversion result in EADC_DAT does not meet EADC_CMP1 register setting. 1 = Conversion result in EADC_DAT meets EADC_CMP1 register setting. **Note:** This bit is cleared by writing 1 to it.
[4]	**EADCMPF0**	**EADC Compare 0 Flag** When the specific sample module EADC conversion result meets setting condition in EADC_CMP0 then this bit is set to 1. 0 = Conversion result in EADC_DAT does not meet EADC_CMP0 register setting. 1 = Conversion result in EADC_DAT meets EADC_CMP0 register setting. **Note:** This bit is cleared by writing 1 to it.
[3]	**ADIF3**	**EADC ADINT3 Interrupt Flag** 0 = No ADINT3 interrupt pulse received. 1 = ADINT3 interrupt pulse has been received. **Note 1:** This bit is cleared by writing 1 to it. **Note 2:**This bit indicates whether an EADC conversion of specific sample module has been completed
[2]	**ADIF2**	**EADC ADINT2 Interrupt Flag** 0 = No ADINT2 interrupt pulse received. 1 = ADINT2 interrupt pulse has been received. **Note 1:** This bit is cleared by writing 1 to it. **Note 2:**This bit indicates whether an EADC conversion of specific sample module has been completed
[1]	**ADIF1**	**EADC ADINT1 Interrupt Flag** 0 = No ADINT1 interrupt pulse received. 1 = ADINT1 interrupt pulse has been received. **Note 1:** This bit is cleared by writing 1 to it. **Note 2:**This bit indicates whether an EADC conversion of specific sample module has been completed

Bits	Description	
[0]	**ADIF0**	**EADC ADINT0 Interrupt Flag** 0 = No ADINT0 interrupt pulse received. 1 = ADINT0 interrupt pulse has been received. **Note 1:** This bit is cleared by writing 1 to it. **Note 2:**This bit indicates whether an EADC conversion of specific sample module has been completed

EADC Status Register 3 (EADC_STATUS3)

Register	Offset	R/W	Description	Reset Value
EADC_STATUS3	EADC_BA+0xFC	R	EADC Status Register 3	0x0000_001F

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			CURSPL				

Bits	Description	
[31:5]	**Reserved**	Reserved.
[4:0]	**CURSPL**	**EADC Current Sample Module (Read Only)** This register shows the current EADC is controlled by which sample module control logic modules. If the EADC is Idle, the bit filed will be set to 0x1F.

EADC Double Data Register n for Sample Module n (EADC_DDAT0~3)

Register	Offset	R/W	Description	Reset Value
EADC_DDAT0	EADC_BA+0x100	R	EADC Double Data Register 0 for Sample Module 0	0x0000_0000
EADC_DDAT1	EADC_BA+0x104	R	EADC Double Data Register 1 for Sample Module 1	0x0000_0000
EADC_DDAT2	EADC_BA+0x108	R	EADC Double Data Register 2 for Sample Module 2	0x0000_0000
EADC_DDAT3	EADC_BA+0x10C	R	EADC Double Data Register 3 for Sample Module 3	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						VALID	OV
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESULT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESULT							

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17]	**VALID**	**Valid Flag** 0 = Double data in RESULT (EADC_DDATn[15:0]) is not valid. 1 = Double data in RESULT (EADC_DDATn[15:0]) is valid. **Note:** This bit is set to 1 when corresponding sample module channel analog input conversion is completed and cleared by hardware after EADC_DDATn register is read (n=0~3).
[16]	**OV**	**Overrun Flag** 0 = Data in RESULT (EADC_DATn[15:0], n=0~3) is recent conversion result. 1 = Data in RESULT (EADC_DATn[15:0], n=0~3) is overwrite. **Note:** If converted data in RESULT[15:0] has not been read before new conversion result is loaded to this register, OV is set to 1. It is cleared by hardware after EADC_DDAT register is read.
[15:0]	**RESULT**	**EADC Conversion Results** This field contains 12 bits conversion results. When the DMOF (EADC_CTL[9]) is set to 0, 12-bit EADC conversion result with unsigned format will be filled in RESULT [11:0] and zero will be filled in RESULT [15:12]. When DMOF (EADC_CTL[9]) is set to 1, 12-bit EADC conversion result with 2'complement format will be filled in RESULT [11:0] and signed bits to will be filled in RESULT [15:12].

EADC Power Management Register (EADC_PWRM)

Register	Offset	R/W	Description	Reset Value
EADC_PWRM	EADC_BA+0x110	R/W	EADC Power Management Register	0x0138_8002

31	30	29	28	27	26	25	24
Reserved				IREFSUT			
23	22	21	20	19	18	17	16
IREFSUT							
15	14	13	12	11	10	9	8
IREFSUT							
7	6	5	4	3	2	1	0
Reserved				PWDMOD		PWUCALEN	PWUPRDY

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:8]	**IREFSUT**	**EADC Internal REF Start-up Time** Set this bit field to control internal reference start-up time. The typical required internal reference start-up time is 1ms when EADC fully power down. Internal reference start-up time = (1/EADC_CLK) x IREFSUT. **Note 1:** EADC fully Power-down mode means both EADC macro and internal reference enter Power-down mode. **Note 2:** EADC macro enters Power-down mode when EADCEN(EADC_CTL[0] = 0. **Note 3:** EADC internal reference enters Power-down mode when PDREF(EADC_REFADJCT[0] = 1.
[7:4]	**Reserved**	Reserved.
[3:2]	**PWDMOD**	**EADC Power-down Mode** Set this bit field to select EADC Power-down mode when system power-down. 00 = EADC Deep Power-down mode. 01 = EADC Power down. 10 = EADC Standby mode. 11 = EADC Deep Power-down mode. **Note:** Different PWDMOD has different power down/up sequence; in order to avoid EADC powering up with wrong sequence, user must keep PWMOD consistent each time in power down and start-up.
[1]	**PWUCALEN**	**Power Up Calibration Function Enable Bit** 0 = Calibration function Disabled at power up. 1 = Calibration function Enabled at power up. **Note:** This bit works together with CALSEL (EADC_CALCTL [3]), see the following {PWUCALEN, CALSEL } Description: PWUCALEN is 0 and CALSEL is 0: No need to calibrate. PWUCALEN is 0 and CALSEL is 1: No need to calibrate. PWUCALEN is 1 and CALSEL is 1: Calibrate when power up.
[0]	**PWUPRDY**	**EADC Power-up Sequence Completed and Ready for Conversion (Read Only)** 0 = EADC is not ready for conversion may be in power down state or in the progress of start-up.

Bits	Description	
		1 = EADC is ready for conversion.

EADC Calibration Control Register (EADC_CALCTL)

Register	Offset	R/W	Description	Reset Value
EADC_CALCTL	EADC_BA+0x114	R/W	EADC Calibration Control Register	0x0000_0008

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**CALSEL**	**CALDONE**	**CALSTART**	**Reserved**

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3]	**CALSEL**	**Select Calibration Functional Block Enable Bit** 0 = Calibration functional block Disabled. 1 = Calibration functional block Enabled.
[2]	**CALDONE**	**Calibration Functional Block Complete (Read Only)** 0 = During a calibration. 1 = Calibration is completed.
[1]	**CALSTART**	**Calibration Functional Block Start** 0 = Stop calibration functional block. 1 = Start calibration functional block. **Note:** This bit is set by software and cleared by hardware after re-calibration is finished.
[0]	**Reserved**	Reserved.

EADC PDMA Control Register (EADC_PDMACTL)

Register	Offset	R/W	Description	Reset Value
EADC_PDMACTL	EADC_BA+0x130	R/W	EADC PDMA Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PDMABUSY	Reserved						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							PDMATEN
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PDMATEN							

Bits	Description	
[31]	**PDMABUSY**	**PDMA Busy Bit** When EADC conversion is completed, the converted data is loaded into EADC_DATn (n: 0 ~ 8) register. User can trigger PDMA to read converted data. When PDMA is too busy to read converted data, this bit field to 1. **Note:** This bit is cleared by writing 1 to it.
[30:9]	**Reserved**	Reserved.
[8:0]	**PDMATEN**	**PDMA Transfer Enable Bit** When EADC conversion is completed, the converted data is loaded into EADC_DATn (n: 0 ~ 8) register, user can enable this bit to generate a PDMA data transfer request. 0 = PDMA data transfer Disabled. 1 = PDMA data transfer Enabled. **Note:** When this bit field is set to 1, user must set EADCIENn (EADC_CTL[5:2], n=0~3) = 0 to disable interrupt.

EADC Select Sampling Time Register 0 (EADC_SELSMP0)

Register	Offset	R/W	Description	Reset Value
EADC_SELSMP0	EADC_BA+0x140	R/W	EADC Select Sampling Time Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		SELSMP7		Reserved		SELSMP6	
23	22	21	20	19	18	17	16
Reserved		SELSMP5		Reserved		SELSMP4	
15	14	13	12	11	10	9	8
Reserved		SELSMP3		Reserved		SELSMP2	
7	6	5	4	3	2	1	0
Reserved		SELSMP1		Reserved		SELSMP0	

Bits	Description	
[4n+3:4n+2] n=0,1..7	**Reserved**	Reserved.
[4n+1:4n] n=0,1..7	**SELSMPn**	**Sample Module n Select Sampling Time Option** EADC sampling time selection bit, in case more sampling time is needed when the input signal resistance is higher. 00 = 3+0 EADC_CLK. 01 = 3+2 EADC_CLK. 10 = 3+4 EADC_CLK. 11 = 3+6 EADC_CLK.

EADC Select Sampling Time Register 1 (EADC_SELSMP1)

Register	Offset	R/W	Description	Reset Value
EADC_SELSMP1	EADC_BA+0x144	R/W	EADC Select Sampling Time Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						SELSMP8	

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1:0]	**SELSMP8**	**Sample Module 8 Select Sampling Time Option** EADC sampling time selection bit, in case more sampling time is needed when the input signal resistance is higher. 00 = 3+0 EADC_CLK. 01 = 3+2 EADC_CLK. 10 = 3+4 EADC_CLK. 11 = 3+6 EADC_CLK.

EADC Reference Voltage Adjust Control Register (EADC_REFADJCTL)

Register	Offset	R/W	Description	Reset Value
EADC_REFADJCTL	EADC_BA+0x150	R/W	EADC Reference Voltage Adjust Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			REFADJ				PDREF

Bits	Description	
[31:5]	**Reserved**	Reserved.
[4:1]	**REFADJ**	**EADC Internal Band-gap Voltage Option** By setting REFADJ according to the following formula and configuration table, one can adjust reference voltage V_{REF} in a small range: V_{REF} = GAIN x VBG. GAIN = 1.6, 2.0, 2.5, 3.0(by setting VREFSEL(EADC_CTL[11:10]) = 00, 01, 10, 11 respectively). VBG is internal band-gap voltage, typical 1.0V, VBG can be configured by the following: 0000 = VBG_type. 0001 = VBG_type + 20 mV. 0010 = VBG_type + 40 mV. 0011 = VBG_type + 60 mV. 0100 = VBG_type + 80 mV. 0101 = VBG_type + 100 mV. 0110 = VBG_type + 120 mV. 0111 = VBG_type + 160 mV. 1000 = VBG_type – 20 mV. 1001 = VBG_type – 40 mV. 1010 = VBG_type – 60 mV. 1011 = VBG_type – 80 mV. 1100 = VBG_type – 100 mV. 1101 = VBG_type – 120 mV. 1110 = VBG_type – 140 mV. 1111 = VBG_type – 160 mV. **Note:** VBG_type = 1.0 V. **Note:** Please wait 1ms for the internal reference voltage to stabilize after setting REFADJ (EADC_REFADJCTL[4:1]).

Bits	Description	
[0]	**PDREF**	**EADC Power Down Internal Reference Control Bit** 0 = Normal operating mode. 1 = Power-down mode. **Note:** If internal reference is in Power-down mode, EADC's reference voltage must come from external V_{REF}.

6.46 System Debug (CoreSight SoC-400)

6.46.1 Overview

The system debug component is based on the Arm CoreSight SoC-400 and provides the function to debug Cortex-A35 and Cortex-M4 at the same time.

6.46.2 Features

- Access to debug features and AHB matrix through a JTAG or Serial Wire Debug (SWD) interface.
- Merging of multiple trace sources (Cortex-A35 Core0 ETM, Cortex-A35 Core1 ETM, and Cortex-M4 ETM) into a single trace stream
- Capture of trace streams on-chip (4 Kbytes ETB) or off-chip (TPIU)
- Cross-triggering between different debug and trace components
- Timestamp generation and system-wide compressed timestamp distribution
- Supports debug authentication for TrustZone and resource isolation

6.46.3 Block Diagram

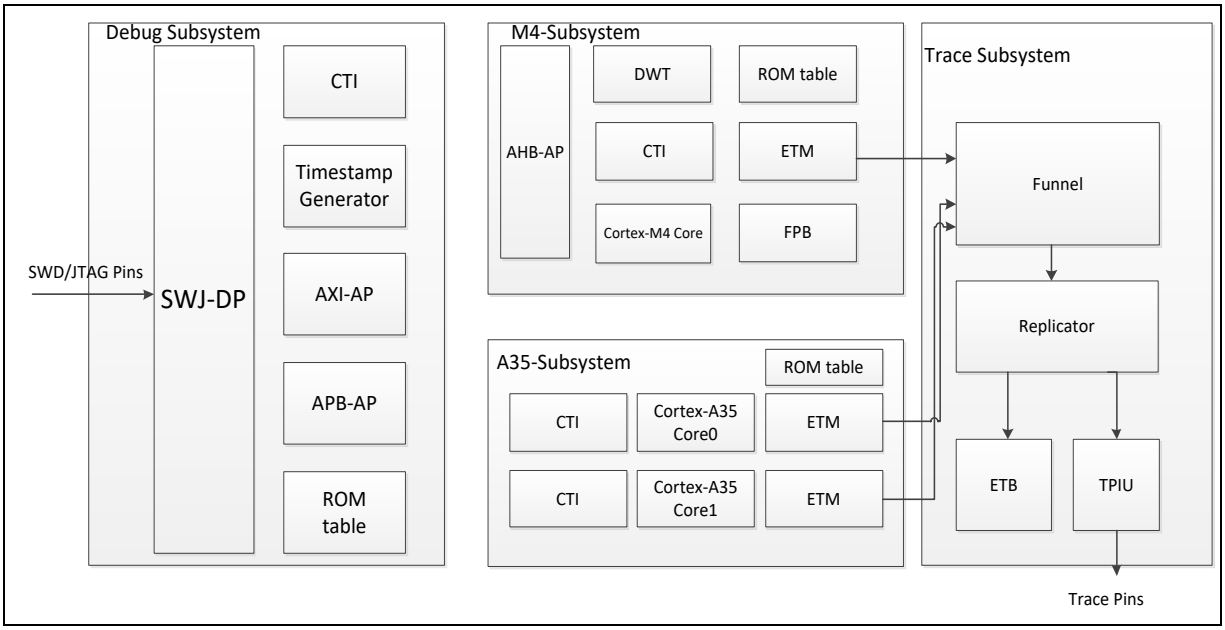


Figure 6.46-1 CoreSight Block Diagram

6.46.4 Basic Configuration

- Clock Source Configuration
  - CoreSight clock source has four clock domains:
    - SWD/JTAG clock: this clock is decided by the external interface pin.
    - DBG clock: this clock is decided by the HCLK and user can divide the frequency by DBGDIV(CLK_CLKDIV3[31:28]).
    - AXI-AP clock: this clock is decided by the Cortex-A35 clock and user can divide the frequency by ACLK0DIV(CLK_CLKDIV0[16]).
    - Trace clock: the clock is decided by the HCLK and user can divide the frequency

by TRACEDIV(CLK_CLKDIV3[27:24]).

- Clock can be enabled in two ways:
  - External debugger: Handshake with csyspwrupreq and csyspwrupack signal, it will auto enable the trace and debug clock when external debugger(ICE) is inserted.
  - Self-hosted debug: Software writes this DBGCKEN(CLK_SYSCLK1[12]) to enable debug clock (DBG clock and AXI-AP clock) and TRACKEN(CLK_SYSCLK1[11]) to enable trace clock separately.
- Reset Configuration
  - CoreSight reset source has one reset source:
    - When Chip reset / WDT reset / POR / BOD is detected, it will reset all the CoreSight component including SWJ-DP
- Pin Configuration
  - SWD/JTAG related pins

Name	GPIO	MFP	BITs	SWD Mode		JTAG Mode	
				I/O type	Description	I/O type	Description
JTMS/SWDIO	GPG[13]	MFP3	1	Input / Output	Serial wire data in/out	Input	JTAG test mode select
JTCK/SWCLK	GPG[12]	MFP3	1	Input	Serial wire clock	Input	JTAG test clock
JTDI	GPG[14]	MFP3	1			Input	JTAG test data input
JTDO	GPG[11]	MFP3	1			Output	JTAG test data output
nJTRST	GPG[15]	MFP3	1			Input	JTAG test reset

Table 6.46-1 SWD/JTAG Pin Table

  - Trace related pins

Name	GPIO	MFP	BITs	I/O Type	Pin Description
Trace Data 0	GPM[12]	MFP3	1	Output	Trace synchronous data out 0
Trace Data 1	GPM[13]	MFP3	1	Output	Trace synchronous data out 1
Trace Data 2	GPM[14]	MFP3	1	Output	Trace synchronous data out 2
Trace Data 3	GPM[15]	MFP3	1	Output	Trace synchronous data out 3
Trace Clock	GPL[15]	MFP3	1	Output	Trace clock

Table 6.46-2 Trace Pin Table

6.46.5 Functional Description

6.46.5.1 Peripheral Stop in Debug Mode

User can decide to stop the peripherals or not when system is in debug mode. These peripherals will be stopped or not by its owner CPU (Cortex-A35 or Cortex-M4) state. For more detailed information, please refer to the "*Arm Cortex-A35 Processor Technical Reference Manual*" and "*Arm Cortex-M4 Technical Reference Manual*".

6.46.5.2 Low Power Mode Emulation

User can enable LPEMU(SYS_CSDBGCTL[3]) to ensure the external debugger connection will not be

lost when system is in low power mode. When this bit is on, the CPU clock will not be auto disabled in low power mode.

6.46.5.3 System ROM Table

This chip has two ROM tables on the APB-AP bus. One is for CoreSight ROM table and the other is for Cortex-A35 ROM table. For detailed function view and programmers model, please refer to Section 2.12 in the "*CoreSight Components Technical Reference Manual (Arm DDI 0314H)*".

- CoreSight ROM table: (Base address: 0xe0800000)

Address Offset In ROM Table	Component Name	Component Base Address (External Debugger View)	Component Base Address (CPU View)	Component Address Offset	Size	Data
0x000	System CTI	0xe0801000	0x60801000	0x1000	4 Kbytes	0x00001003
0x004	Timestamp Generator	0xe0802000	0x60802000	0x2000	4 Kbytes	0x00002003
0x008	ATB replicator	0xe0803000	0x60803000	0x3000	4 Kbytes	0x00003003
0x00C	TPIU	0xe0804000	0x60804000	0x4000	4 Kbytes	0x00004003
0x010	ETB	0xe0805000	0x60805000	0x5000	4 Kbytes	0x00005003
0x014	ATB funnel	0xe0806000	0x60806000	0x6000	4 Kbytes	0x00006003
0x018	Cortex-A35 ROM table	0xe0c00000	0x60c00000	0x400000	4 Kbytes	0x00400003

Table 6.46-3 CoreSight ROM Table

- Cortex-A35 ROM table (Base address: 0xe0c00000)

Address Offset In ROM Table	Component Name	Component Base Address (Debugger View)	Component Base Address (CPU View)	Component Address Offset	Size	Data
0x000	Core0 Debug	0xe0c10000	0x60c10000	0x10000	4 Kbytes	0x00010003
0x004	Core0 CTI	0xe0c20000	0x60c20000	0x20000	4 Kbytes	0x00020003
0x008	Core0 PMU	0xe0c30000	0x60c30000	0x30000	4 Kbytes	0x00030003
0x00c	Core0 ETM	0xe0c40000	0x60c40000	0x40000	4 Kbytes	0x00040003
0x000	Core1 Debug	0xe0d10000	0x60d10000	0x110000	4 Kbytes	0x00110003
0x004	Core1 CTI	0xe0d20000	0x60d20000	0x120000	4 Kbytes	0x00120003
0x008	Core1 PMU	0xe0d30000	0x60d30000	0x130000	4 Kbytes	0x00130003
0x00c	Core1 ETM	0xe0d40000	0x60d40000	0x140000	4 Kbytes	0x00140003

Table 6.46-4 Cortex-A35 ROM Table

6.46.5.4 Debug Access Port (DAP)

This chip provides one DP(SWJ-DP) and three APs (AXI-AP, AHB-AP, APB-AP). SWJ-DP can support external interface whose protocol is SWD or JTAG.AHB-AP(AP2) for Cortex-M4 view, the AXI-AP(AP0) for Cortex-A35 view and the APB-AP(AP1) for CoreSight debug component. For detailed function view and programmers model, please refer to Chapter 2 in the "*CoreSight Components Technical Reference Manual (Arm DDI 0314H)*".

6.46.5.5 Embedded Cross Trigger (ECT)

The ECT can communication with CTI debug events to debug and synchronous with multi-core system. This chip has four CTI components including Cortex-A35 Core0 CTI, Cortex-A35 Core1 CTI, Cortex-M4 CTI and trace subsystem CTI.

For more details about Cortex-A35 CTI, please refer to Chapter C4 in the "*Arm Cortex -A35 Processor Technical Reference Manual r0p1*".

For more details about Cortex-M4 CTI, please refer to "*Cortex-M Series Processors and CoreSight SoC-400 Integration Manual*".

About System CTI detailed function view and programmers model, please refer to Chapter 4 in the "*CoreSight Components Technical Reference Manual (Arm DDI 0314H)*".

6.46.5.6 ATB Replicator

The ATB replicator enables two trace sinks to be wired together and to operate from the same incoming trace stream. This chip provides two trace sinks including Trace port interface unit and Embedded Trace Buffer. For detailed function view and programmers model, please refer to Chapter 6 in the "*CoreSight Components Technical Reference Manual (Arm DDI 0314H)*".

6.46.5.7 CoreSight Trace Funnel (CSTF)

The CSTF is used when there is more than one trace source. In this chip, the CSTF combines three trace streams including Cortex-A35 Core0 ETM(S0),Cortex-A35 Core1 ETM(S1) and Cortex-M4 ETM(S2) onto a single ATB bus. For detailed function view and programmers model, please refer to Chapter 7 in the "*CoreSight Components Technical Reference Manual (Arm DDI 0314H)*".

6.46.5.8 Trace Port Interface Unit (TPIU)

The TPIU acts as a bridge between the on-chip trace data, with separate IDs, to a data stream, encapsulating IDs where required, that is then captured by a Trace Port Analyzer (TPA). For detailed function view and programmers model, please refer to Chapter 8 in the "*CoreSight Components Technical Reference Manual (Arm DDI 0314H)*".

6.46.5.9 Embedded Trace Buffer (ETB)

The ETB provides on-chip storage of trace data using 32-bit RAM. This chip provides 4 Kbytes ETB. For detailed function view and programmers model, please refer to Chapter 9 in the "*CoreSight Components Technical Reference Manual (Arm DDI 0314H)*".

6.46.5.10 Timestamp Generator

The timestamp generator generates a timestamp value that provides a consistent view of time for three blocks including Cortex-A35 Core0 ETM, Cortex-A35 Core1 ETM and Cortex-M4 ETM.

For detailed function view and programmers model, please refer to Section 2.4 and 3.10 and Chapter 7 in the "*Arm CoreSight SoC-400 Technical Reference Manual r3p2*".

6.46.5.11 Cortex-A35 Debug Components

For detailed function view and programmers model, please refer to Part C in the "*Arm Cortex -A35 Processor Technical Reference Manual r0p1*".

6.46.5.12 Cortex-M4 ROM Table

The ROM table is placed on the Cortex-M4 AHB-AP. For detailed function view and programmers model, please refer to Chapter 8 in the "*Cortex-M4 Technical Reference Manual r0p0*".

- Cortex-M4 ROM table (Base address: 0xe00ff000)

Address Offset In ROM Table	Component Name	Component Base Address (Debugger And Corte-M4 View)	Component Address Offset	Size	Data
0x000	SCS	0xe000e000	0xfff0f000	4 Kbytes	0xfff0f003
0x004	DWT	0xe0001000	0xfff02000	4 Kbytes	0xfff02003
0x008	FPB	0xe0002000	0Xfff03000	4 Kbytes	0xfff03003
0x00c	ITM	0xe0000000	0xfff01000	4 Kbytes	0xfff01002
0x010	TPIU	0xe0040000	0xfff41000	4 Kbytes	0xfff41002
0x014	ETM	0xe0041000	0xfff42000	4 Kbytes	0xfff42003
0x018	CTI	0xe0042000	0xfff43000	4 Kbytes	0xfff43003

Table 6.46-5 Cortex-M4 ROM Table

6.47 Peripherals Interconnection

6.47.1 Overview

Some peripherals have interconnections which allow autonomous communication or synchronous action between peripherals without needing to involve the CPU. Peripherals interact without CPU saves CPU resources, reduces power consumption, operates with no software latency and fast responds.

6.47.2 Peripherals Interconnect Matrix table

Source	Destination				
	EADC	ECAP	EPWM	TIMER	QEI
LVD	-	-	2	2	-
Clock Fail	-	-	2	2	-
RTP-M4 Lockup	-	-	2	2	-
EADC	-	-	2	-	-
EPWM	1	-	3	-	-
HIRC	-	-	-	8	-
LIRC	-	-	-	8	-
HXT	-	-	-	8	-
LXT	-	-	-	8	-
QEI	-	6	-	-	-
TIMER	1	-	4	5	7

Table 6.47-1 Peripherals Interconnect Matrix table

6.47.3 Functional Description

1. From EPWM, TIMER to EADC
   - EPWM Trigger EADC Conversion

     EPWM can be one of the EADC conversion trigger source.

     Setting the EADC external hardware trigger input source from EPWM trigger described in section 6.45.5.9.

   - Timer Trigger EADC Conversion

     Timer0 ~ Timer11 can be one of the EADC conversion trigger source. When timer counter value matches the timer compared value or when the TMx_EXT pin edge transition meets setting, timer will trigger the ADC to start the conversion.

     Setting the EADC external hardware trigger input source from timer trigger described in section 6.45.5.9.

2. From EADC, LVD, Clock Fail and RTP-M4 Lockup to EPWM or TIMER PWM Mode
   - EPWM Brake Source

     EPWM brake source can be EADC compare mode result or several different system fail conditions include clock fail, Low-voltage detect and RTP-M4 core lockup. When EADC

compare result matches or system fault, EPWM brake signal generated, EPWM output will be set to protect the power switch controlled by EPWM.

The detail setting of EPWM brake function is described in section 6.18.5.23.

- Timer PWM Mode Brake Source

  TIMER PWM mode brake source can be several different system fail conditions include clock fail, Low-voltage detect and RTP-M4 core lockup. When system fault, TIMER PWM mode brake signal generated, TIMER PWM mode output will be set to protect the power switch controlled by TIMER PWM mode.

  The detail setting of TIMER PWM Mode brake function is described in section 6.17.6.17.

3. From EPWM to EPWM

- EPWM Synchronous Start Function

  Select synchronous source from EPWM0, EPWM1 or EPWM2, and select EPWM channels. The chosen EPWM channels will start counting at the same time once the synchronous start function is enabled and CNTSEN(EPWM_SSTRG[0]) is set.

  The detail setting of EPWM synchronous start function is described in section 6.18.5.19.

4. From TIMER to EPWM

- Timer Generates Trigger Pulses as EPWM External Clock Source

  Timer0 ~ Timer11 can generates trigger pules as EPWM external clock source.

  When timer counter value matches the timer compared value or when the TMx_EXT pin edge transition meets setting, timer can generate a trigger pulse by setting described in section 6.17.5.10.

  The setting of EPWM clock source are described in section 6.18.3.

5. From TIMER0/2/4/6/8/10 to TIMER1/3/5/7/9/11

- Inter-Timer Trigger Capture Mode

  Timer0/2/4/6/8/10 will be forced in event counting mode, counting with external event, and will generate an internal signal (INTR_TMR_TRG) to trigger Timer1/3/5/7/9/11 start or stop counting. The Timer1/3/5/7/9/11 will be forced in capture mode and start/stop trigger-counting by Timer0/2/4/6/8/10 counter status.

  The detail of inter-timer trigger capture mode are described in section 6.17.5.11.

6. From QEI to ECAP

- ECAP Input noisc filter

  The architecture of ECAP input noise filter is similar to that one used for QEI. With 6 sampling-rate options, it supports a wide range of filtering noise, the duration of filtered noise and the duration of the signal that is guaranteed to be sampled.

  The detail setting of modulation is described in section 6.20.5.1.

7. From TIMER to QEI

- TIMER TIF event o QEI

  When QEI bit HOLDCNT(QEI_CTL[24]) set, the CNT(QEI_CNT[31:0]) content will be captured into QEI Counter Hold Register CNTHOLD(QEI_CNTHOLD[31:0]), the data will be held until the next HOLDCNT (QEI_CTL[24]) trigger comes. The bit HOLDCNT can be set by writing 1 to it or the rising edge of timers interrupt flags TIF (TIMERx_INTSTS[0])The detail setting of modulation is described in section 6.20.5.11.

  The detail setting of modulation is described in section 6.17.5.1.

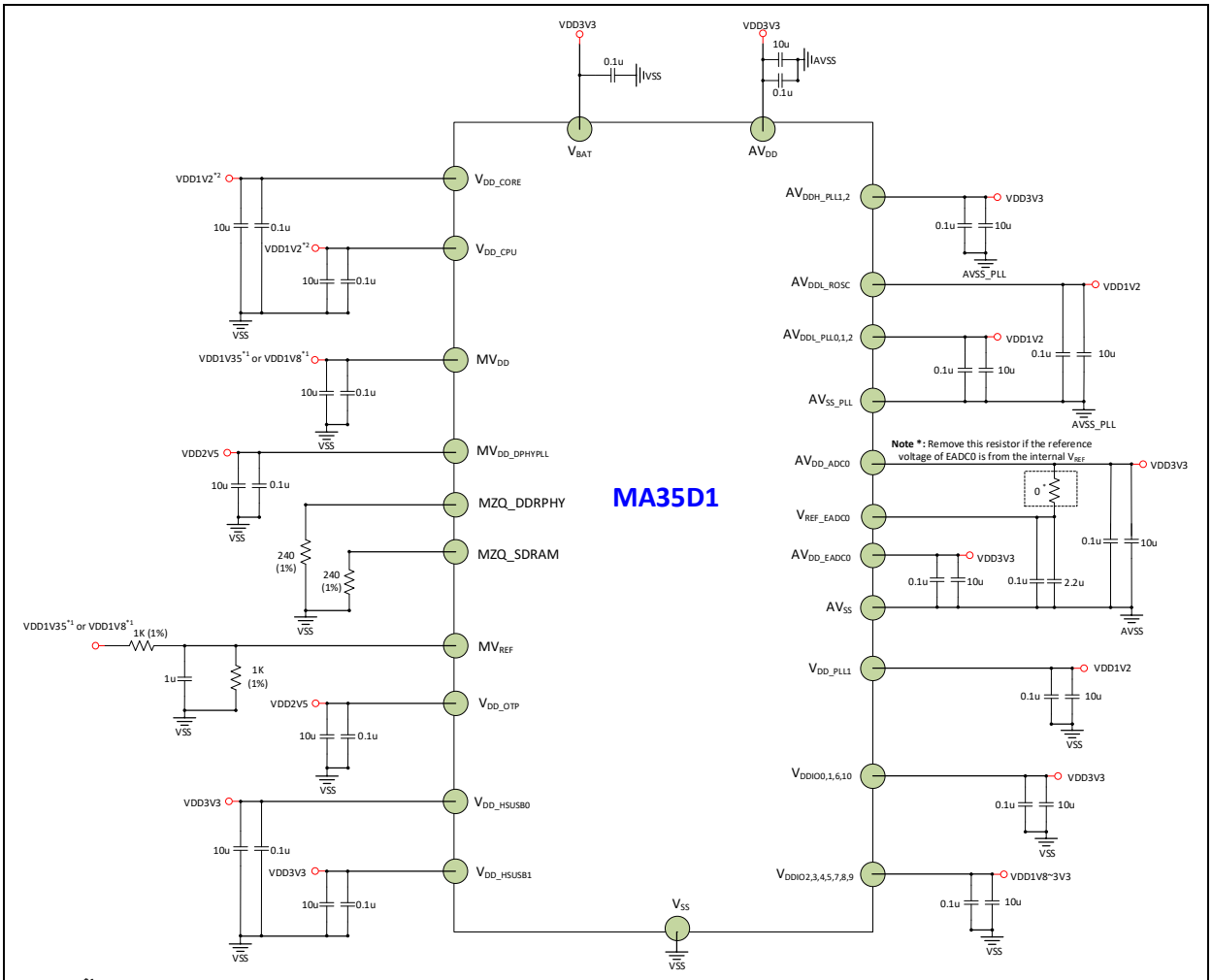
8. From HIRC, LIRC, HXT or LXT to Timer Capture Function

- Measure the Time Interval of HIRC, LIRC, HXT or LXT clock Speed

  Sets the timer capture source from HIRC, LIRC, HXT or LIRC clock and measures the time interval of the signal by using timer capture function. Users can use the results of time interval to trim HIRC through software.

  The detail of time capture function setting are described in section 6.17.5.8 and 6.17.5.9.

7 APPLICATION CIRCUIT



Note *1: For the actual input voltage of MV_{DD}, please refer to the part number defining the DDR type.
Note *2: For the actual input voltage ranges of V_{DD_CORE} and V_{DD_CPU}, please refer to the Minimum/Typical/Maximum values in section 8.2.1 of Datasheet document.

Figure 7-1 Application Circuit

Note: MV_{DD} is the power supply for internal DDR2/DDR3L-type SDRAM. Please refer to the part number from MA35D1 selection guide to supply the correct voltage on these pins. It is 1.8V for internal DDR2-type SDRAM or 1.35V for internal DDR3L-type SDRAM. $MV_{REF} = 1/2MV_{DD}$.

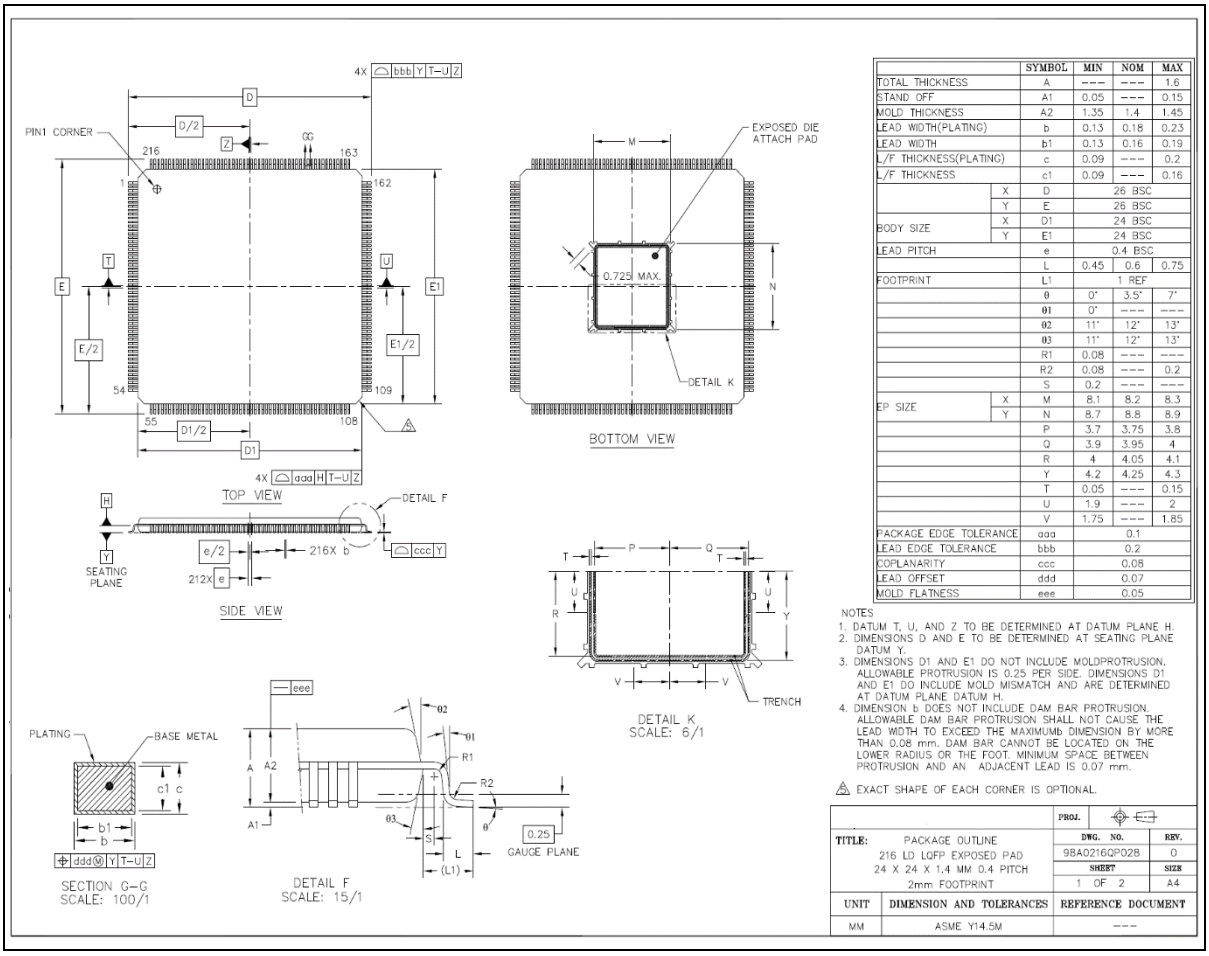
8 ELECTRICAL CHARACTERISTICS

For information on the MA35D1 Series Electrical characteristics, please refer to MA35D1 Series Datasheet.

9 PACKAGE DIMENSIONS

Package is Halogen-free, RoHS-compliant and TSCA-compliant.

9.1 LQFP-EP 216-Pin (24x24x1.4mm, footprint 2.0mm)

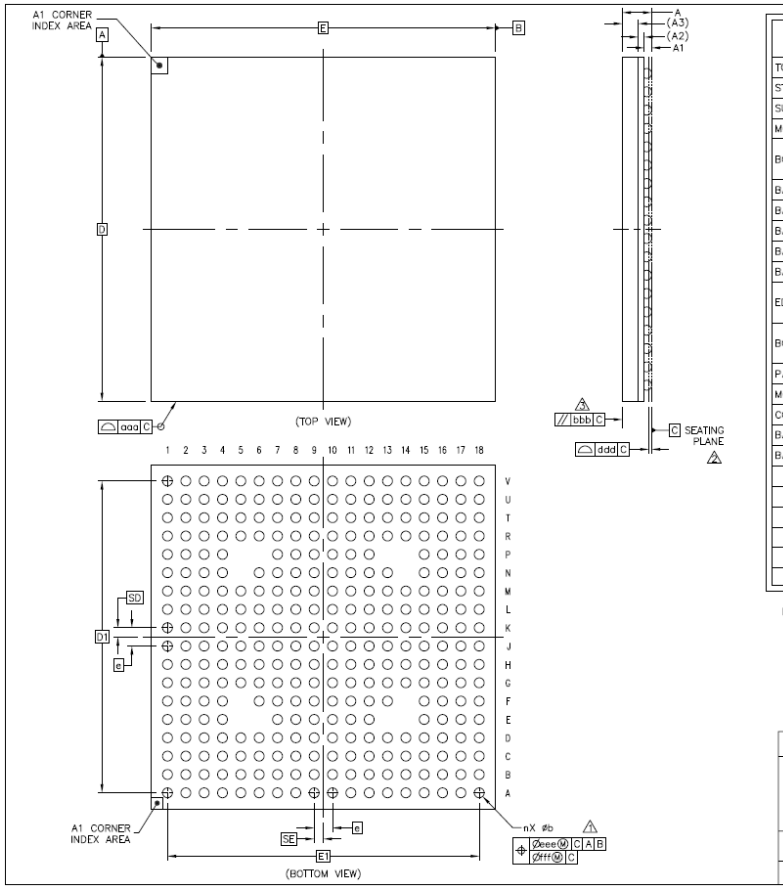


		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	---	---	1.6
STAND OFF		A1	0.05	---	0.15
MOLD THICKNESS		A2	1.35	1.4	1.45
LEAD WIDTH(PLATING)		b	0.13	0.18	0.23
LEAD WIDTH		b1	0.13	0.16	0.19
L/F THICKNESS(PLATING)		c	0.09	---	0.2
L/F THICKNESS		c1	0.09	---	0.16
BODY SIZE	X	D	26 BSC		
	Y	E	26 BSC		
	X	D1	24 BSC		
	Y	E1	24 BSC		
LEAD PITCH		e	0.4 BSC		
		L	0.45	0.6	0.75
FOOTPRINT		L1	1 REF		
		θ	0°	3.5°	7°
		θ1	0°	---	---
		θ2	11°	12°	13°
		θ3	11°	12°	13°
		R1	0.08	---	---
		R2	0.08	---	0.2
		S	0.2	---	---
EP SIZE	X	M	8.1	8.2	8.3
	Y	N	8.7	8.8	8.9
		P	3.7	3.75	3.8
		Q	3.9	3.95	4
		R	4	4.05	4.1
		Y	4.2	4.25	4.3
		T	0.05	---	0.15
		U	1.9	---	2
		V	1.75	---	1.85
PACKAGE EDGE TOLERANCE		aaa	0.1		
LEAD EDGE TOLERANCE		bbb	0.2		
COPLANARITY		ccc	0.08		
LEAD OFFSET		ddd	0.07		
MOLD FLATNESS		eee	0.05		

Figure 9.1-1 LQFP-EP 216-Pin Package Demension

Note: To make the EPAD and PCB soldered tightly, please download the [MA35 Series LQFP-EP 216-Pin Package Application Note] document for reference in PCB design and layout as well as PCBA assembly and soldering.

9.2 BGA 312-Ball (15x15x1.4mm, 0.8mm pitch)



	SYMBOL	COMMON DIMENSIONS		
		MIN.	NOR.	MAX.
TOTAL THICKNESS	A	---	---	1.4
STAND OFF	A1	0.27	---	0.37
SUBSTRATE THICKNESS	A2		0.26	REF
MOLD THICKNESS	A3		0.7	REF
BODY SIZE	D		15	BSC
	E		15	BSC
BALL DIAMETER			0.4	
BALL OPENING			0.35	
BALL WIDTH	b	0.38	---	0.48
BALL PITCH	e		0.8	BSC
BALL COUNT	n		312	
EDGE BALL CENTER TO CENTER	D1		13.6	BSC
	E1		13.6	BSC
BODY CENTER TO CONTACT BALL	SD		0.4	BSC
	SE		0.4	BSC
PACKAGE EDGE TOLERANCE	aaa		0.1	
MOLD FLATNESS	bbb		0.2	
COPLANARITY	ddd		0.12	
BALL OFFSET (PACKAGE)	eee		0.15	
BALL OFFSET (BALL)	fff		0.08	

NOTES:

1. DIMENSION b IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER, PARALLEL TO DATUM PLANE C.
2. DATUM C (SEATING PLANE) IS DEFINED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.
3. PARALLELISM MEASUREMENT SHALL EXCLUDE ANY EFFECT OF MARK ON TOP SURFACE OF PACKAGE.

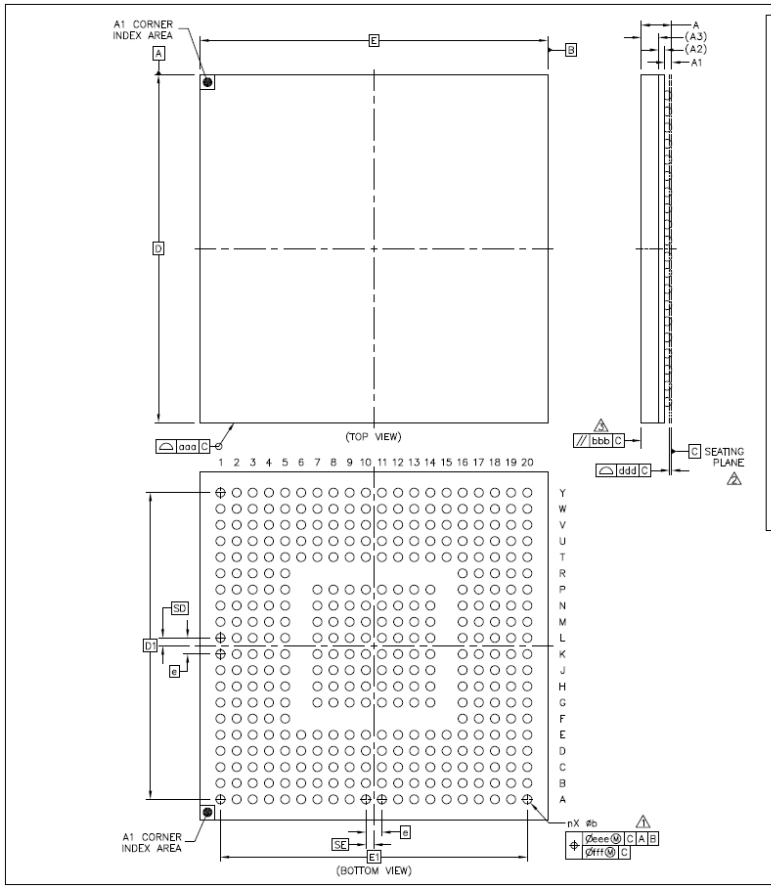
ASE Group

TITLE	DWG. NO.	REV.
PACKAGE OUTLINE 312 BALLS LFBGA 15 X 15 X 1.4 MM 0.8 PITCH	98A0312MB004	002
	SHEET 1 OF 2	SIZE A4

UNIT	DIMENSION AND TOLERANCES	REFERENCE DOCUMENT
MM	ASME Y14.5M	---

Figure 9.2-1 BGA 312-Ball Package Demension

9.3 BGA 364-Ball (14x14x1.4mm, 0.65mm pitch)



	SYMBOL	COMMON DIMENSIONS		
		MIN.	NOR.	MAX.
TOTAL THICKNESS	A	---	---	1.4
STAND OFF	A1	0.22	---	0.32
SUBSTRATE THICKNESS	A2		0.26	REF
MOLD THICKNESS	A3		0.7	REF
BODY SIZE	D		14	BSC
	E		14	BSC
BALL DIAMETER			0.35	
BALL OPENING			0.3	
BALL WIDTH	b	0.32	---	0.42
BALL PITCH	e		0.65	BSC
BALL COUNT	n		364	
EDGE BALL CENTER TO CENTER	D1		12.35	BSC
	E1		12.35	BSC
BODY CENTER TO CONTACT BALL	SD		0.325	BSC
	SE		0.325	BSC
PACKAGE EDGE TOLERANCE	aaa		0.1	
MOLD FLATNESS	bbb		0.2	
COPLANARITY	ddd		0.08	
BALL OFFSET (PACKAGE)	eee		0.15	
BALL OFFSET (BALL)	fff		0.08	

NOTES:

1. DIMENSION b IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER, PARALLEL TO DATUM PLANE C.
2. DATUM C (SEATING PLANE) IS DEFINED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.
3. PARALLELISM MEASUREMENT SHALL EXCLUDE ANY EFFECT OF MARK ON TOP SURFACE OF PACKAGE.

ASE Group

TITLE	DWG. NO.	REV.
PACKAGE OUTLINE 364 BALLS LFBGA 14 X 14 X 1.4 MM 0.65 PITCH	98A0364MB008	002
	SHEET 1 OF 2	SIZE A4
UNIT: MM	DIMENSION AND TOLERANCES: ASME Y14.5M	REFERENCE DOCUMENT: ---

Figure 9.3-1 BGA 364-Ball Package Demension

9.4 Thermal Characteristics

9.4.1 Thermal Performance of Package under Forced Convection

PKG type	PCB condition	θ ja (°C/W)			Psi JT (°C/W)	θ jc (°C/W)	θ jb (°C/W)
		0 m/s	1 m/s	2m/s			
LQFP-EP 216-Pin 24x24 - Ambient	JEDEC JESD 51-7, 2S2P	17.6	13.50	12.40	0.41	7.10	10.03
LFBGA 312-Ball 15x15 - Ambient	JEDEC JESD 51-9, 2S2P	21.70	18.40	17.40	0.29	6.10	10.22
LFBGA 364-Ball 14x14 - Ambient	JEDEC JESD 51-9, 2S2P	25.10	22.20	21.10	0.37	8.60	13.82

Table 9.4-1 Thermal Performance of Package

9.4.2 Thermal Performance Terminology

The major thermal dissipation paths can be illustrated as following

T_J: the maximum junction temperature;

T_A: the ambient or environment temperature;

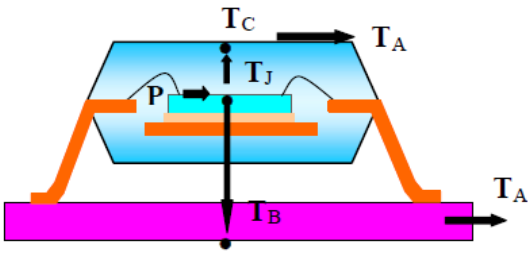
T_C: the top center of compound surface temperature;

T_B: the bottom center of PCB surface temperature;

P: total input power

1. Junction to ambient thermal resistance, θ_{JA}, **defined as:**

$$\theta_{JA} = \frac{T_J - T_A}{P} \quad ; \qquad (1)$$

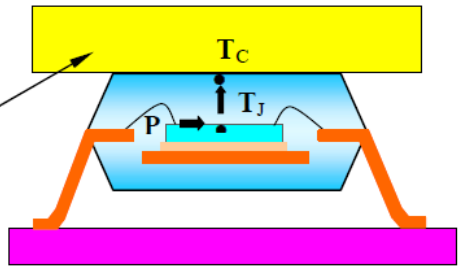


Thermal Dissipation of LQFP package

Figure 9.4-1 Junction to Ambient Thermal Resistance

2 .Junction to case thermal resistance, θ_{JC}, **defined as:**

$$\theta_{JC} = \frac{T_J - T_C}{P} \quad ; \qquad (2)$$



Attach a block with constant temperature onto package.

Figure 9.4-2 Junction to Case Thermal Resistance

9.4.3 Simulation Conditions

Input Power	**Main Die: 1.5W** **DRAM Die: 0.5 W**
Test Board (PCB)	FR4 PCB thickness is 1.6mm Copper thickness is 2-OZ for Microstrip (Top/Bottom) layer Copper thickness is 1-OZ for stripline (Inner) layer LQFP216 follows JESD 51-7, 2S2P PCB is size 3”x 4.5” BGA follows JESD 51-9, 2S2P PCB size is 4”x 4.5”
Control Condition	Air Flow = 0, 1, 2, 3 m/s

Table 9.4-2 Thermal Characteristics Simulation Conditions

9.5 PCB Reflow Profile Suggestion

9.5.1 Profile Setting Consideration

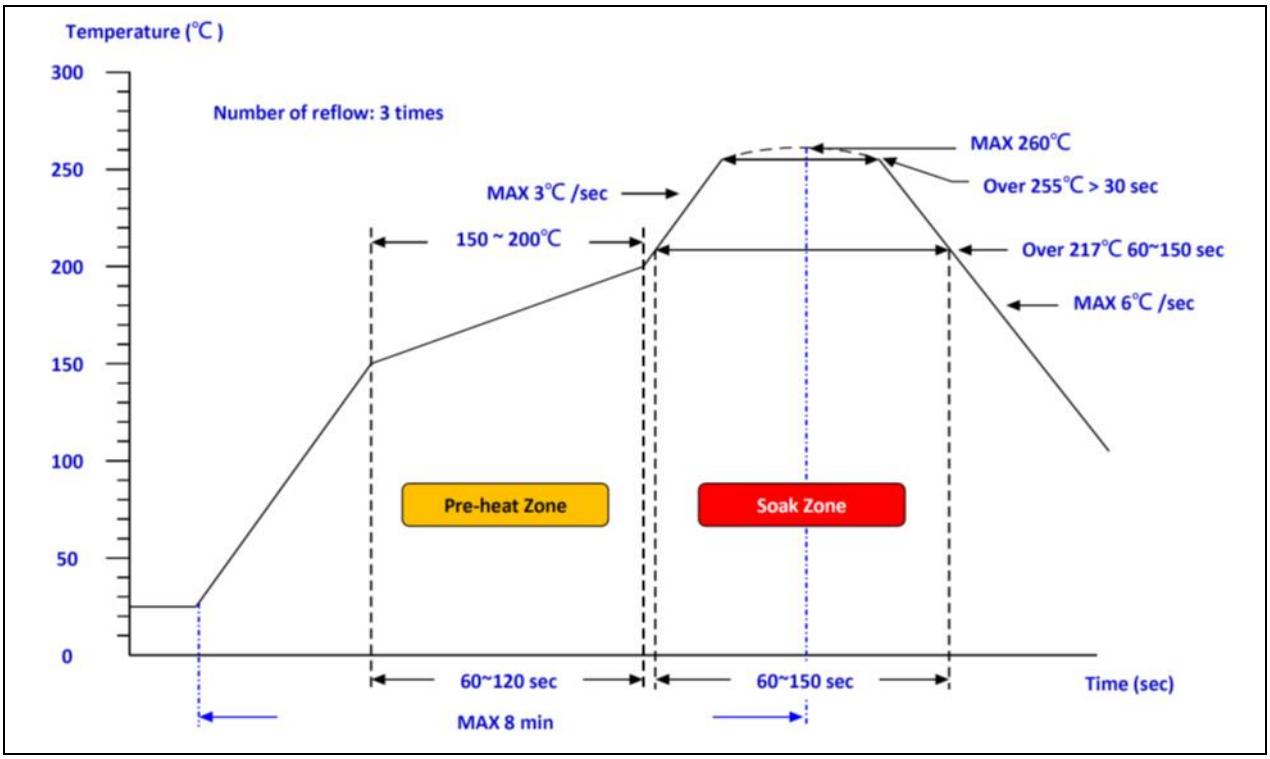


Figure 9.5-1 PCB Reflow Profile Diagram

Profile Feature	Sn-Pb Eutestic Assembly		Pb-Free Eutestic Assembly	
	Large Body	Small Body	Large Body	Small Body
Average ramp-up rate (T_L to T_P)	< 3°C/second		< 3°C/second	
Preheat • Temperature Min (Ts_{min}) • Temperature Max (Ts_{max}) • Time (min to max) (ts)	100°C 150°C 60-120 seconds		150°C 200°C 60-120 seconds	
Time maintained above: • Temperature (T_L) • Time (t_L)	183°C 60-150 seconds		217°C 60-150 seconds	
Peak Temperature (T_p)	225+0/-5°C		245+5/-5°C	
Time within 5°C of actual Peak Temperature (t_p)	10-20 seconds		10-30 seconds	
Ramp-down Rate	6°C/second max.		6°C/second max.	
Time 25°C to Peak Temperature	6 minutes max.		8 minutes max.	
Notes: 1. All temperatures refer to topside of the package, measured on the package body surface.				

2. Depends on other parts on board density and follower solder paste manufacturer's guideline.

Table 9.5-1 PCB Reflow Profile Parameters

9.5.2 Profile Suggestion

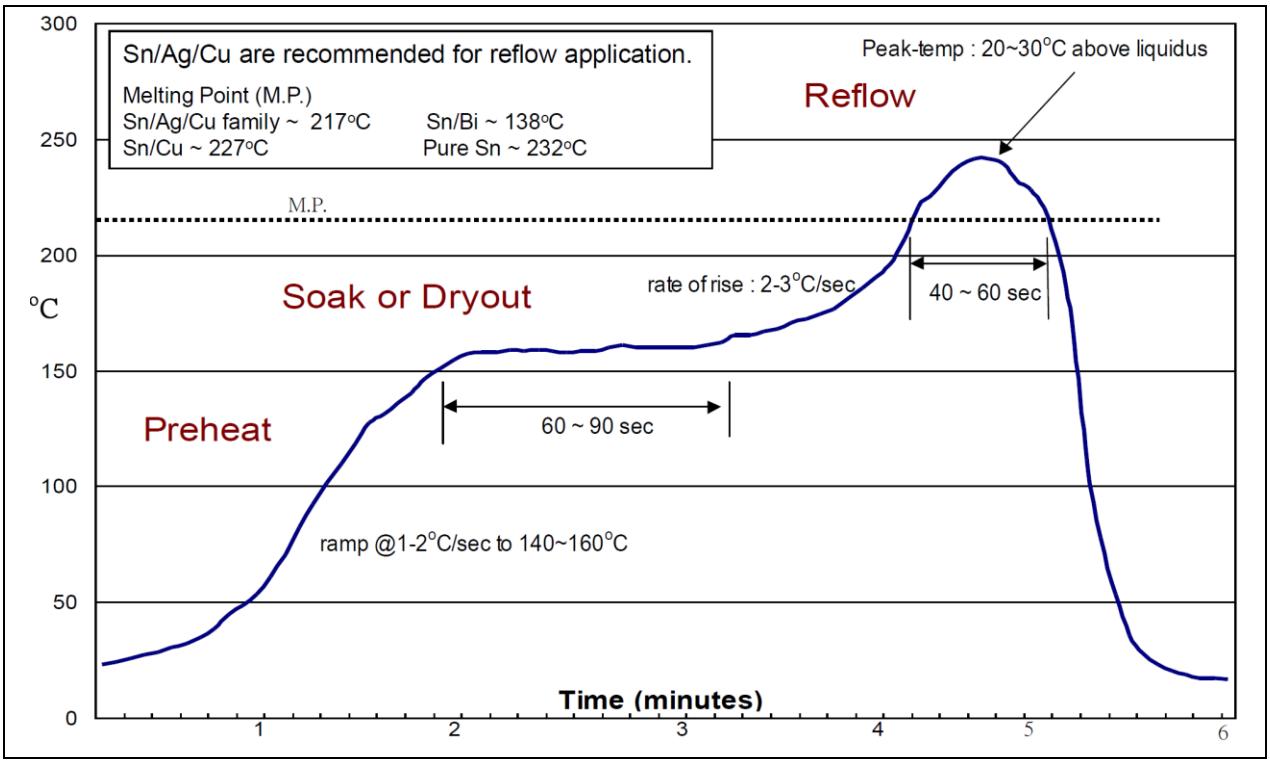


Figure 9.5-2 Profile Suggestion for MA35D1 Series

9.5.3 PCB Assembly Considerations

This profile is designed for use with 96.5Sn/3Ag/0.5Cu and can serve as a general guideline in establishing a reflow profile.

Reflow Profile:

- Heating-up@1~3°C/sec to 140°C
- Preheat@140-150°C for 120~160 sec
- Ramp@2~3 °C/sec to peak temperature (220 ~ 225°C), temperature over 183°C for 45~ 75 sec
- Cooling down to room temperature@4~2°C/sec to avoid undesired intermetallic compound layer.

9.6 PKG Baking and Vacuumed

The moisture-sensitivity caution label (Figure 9.6-1) is applied to the outside of the sealed moisture-barrier bag. This label contains detailed information specific to the device (moisture-sensitivity level, shelf life, etc.).

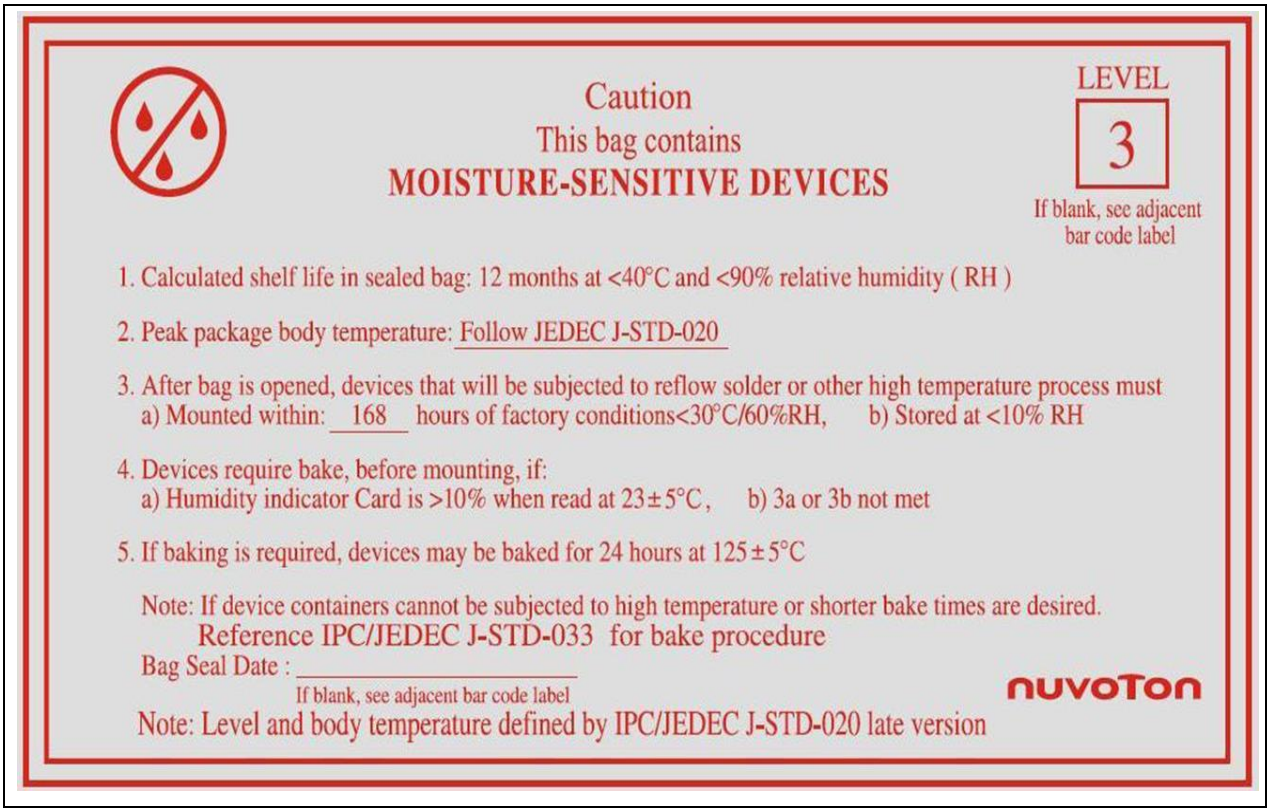


Figure 9.6-1 Cautions for PKG Baking

10 ABBREVIATIONS

10.1 Abbreviations

Acronym	Description
ADC	Analog-to-Digital Converter
AES	Advanced Encryption Standard
AIC	Advanced Interrupt Controller
APB	Advanced Peripheral Bus
AHB	Advanced High-Performance Bus
AMBA	Advanced Microcontroller Bus Architecture
BCH	Bose–Chaudhuri–Hocquenghem
BPS	Bit Per Second
CAN	Controller Area Network
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
DDR	Double Data Rate
DDR2	Double Data Rate 2
DMA	Direct Memory Access
EBI	External Bus Interface
ECC	Elliptic Curve Cryptography
ECC	Error Correcting code
EHCI	Enhance Host Controller Interface
EINT	External Interrupt pin
EMAC	Ethernet MAC Controller
eMMC	Embedded Multimedia Card
ETU	Elementary time unit
FIFO	First In, First Out
FIQ	Fast Interrupt
FMI	Flash Memory Interface
GPIO	General-Purpose Input/Output
HCLK	The Clock of Advanced High-Performance Bus
HMAC	keyed-Hash Message Authentication Code
HSUSBD	High Speed USB 2.0 Device Controller
HXT	12 MHz External High Speed Crystal Oscillator
I^2C	Inter-Integrated Circuit
I^2S	Inter-IC Sound
LIN	Local Interconnect Network

LPDDR	Low Power DDR
LSB	Least Significant Bit
LVD	Low Voltage Detect
LVR	Low Voltage Reset
LXT	32.748 kHz External Low Speed Crystal Oscillator
MLC	Multi-Level Cell NAND Flash
MMU	Memory Management Unit
MSB	Most Significant Bit
OHCI	Open Host Controller Interface
PCLK	The Clock of Advanced Peripheral Bus
PCM	Pulse Code Modulation
PDMA	Peripheral Direct Memory Access
PLL	Phase-Locked Loop
PMBus	Power Management Bus
PRNG	Pseudo Random Number Generator
PWM	Pulse Width Modulation
RMII	Reduced Media Independent Interface
RSA	Rivest、Shamir and Adleman Cryptography
RTC	Real Time Clock
SC	Smart Card
SD	Secure Digital
SDHC	Secure Digital High Capacity
SDIC	SDRAM Interface Controller
SDIO	Secure Digital Input Output
SDR	Single Data Rate
SHA	Secure Hash Algorithm
SLC	Single Level Cell NAND Flash
SMBus	System Management Bus
SPI	Serial Peripheral Interface
SPS	Samples per Second
TMR	Timer Controller
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
WDT	Watchdog Timer
WWDT	Window Watchdog Timer

11 REVISION HISTORY

Date	Revision	Description
2022.10.5	1.00	• Initial version.
2023.2.24	1.01	• Added note in VDD_CPU pin description for LQFP216 package in section 4.2.4. • Added new part number MA35D16F887C in sections 3.1, 3.3 and 4.2.1.
2023.3.21	1.02	• Added notes in section 4.2.1, 4.2.2, 4.2.3 and 4.2.4. • Updated the contents in section 6.33.6.
2023.4.27	1.03	• Updated the contents in section 4.2.1.
2023.6.12	1.04	• Updated the core speed in chapter 1 and sections 1.1 and 2.1. • Updated Figure 3.2-1, Figure 5.1-1, Figure 6.5-1 and Figure 6.5-5. • Updated the sections 3.3, 6.4.8 and 6.33.7.
2023.9.15	1.05	• Updated the sections 3.3, 4.2.1 and 6.5.10. • Updated Figure 5.1-1.
2023.12.25	1.06	• Updated the operating temperature description from “Junction Temperature T_J (Absolute)” to “Operating Junction Temperature (T_J)" in section 3.3. • Updated some voltage values from 1.1V to 1.2V of System Power Distribution in section 6.4.3. • Updated the Basic Configuration of Serial Peripheral Interface (SPI) and Controller Area Network with Feasibility Data Rate (CAN FD) in sections 6.27.4 and 6.33.4. • Added the Basic Configuration of Gigabit Media Access Controller (GMAC) and LCD Display Controller (DISP) in sections 6.34.4 and 6.40.4. • Updated the sections 2.1, 4.2.4, 6.1.6, 6.4.4, 6.4.8, 6.5.2, 6.5.10, 6.6.6, 6.9.4, 6.15.5, 6.17.2, 6.17.6, 6.17.8, 6.18.2, 6.18.5, 6.23.7, 6.34.8 and 6.47.3. • Updated Figure 5.1-1, Figure 6.5-1, Figure 6.5-3, Figure 6.17-3, Figure 6.17-6, Figure 6.17-7, Figure 6.17-30, Figure 6.17-34, Figure 6.18-1, Figure 6.18-34 and Figure 7-1. • Updated Table 6.24-1.
2024.2.5	1.07	• Added a note for EPAD in Figure 4.1-1 and sections 4.2.1 and 4.2.4. • Added a note for PB.8 (ADC0_CH0) in sections 4.2.1, 4.2.2, 4.2.3 and 4.2.4. • Updated the sections 6.33.4 and 6.44.4. • Updated Figure 6.26-1 and Figure 6.44-1.
2024.5.31	1.08	• Added a note for GPIO PE.15 (UART0_RXD) pin in sections 4.2.1, 4.2.2, 4.2.3, 4.2.4 and 6.24.4.1. • Updated the section 6.24.5.8. • Updated Figure 6.14-1, Figure 6.35-1 and Figure 6.35-5.

2024.8.23	1.09	• Added new part number MA35D16AJ87C in sections 3.1, 3.3 and 4.2.2. • Updated the sections 3.2, 6.15.7, 6.23.7, 6.33.4.3 and 6.33.4.4.
2024.9.18	1.10	• Added new part number MA35D14F884 and MA35D14F984 in sections 3.1, 3.3, 4.2.1.1 and 4.2.4. • Updated the section 3.2. • Added notes in sections 4.2.4, 6.4.3 and in Figure 7-1.
2025.3.12	1.11	• Added a note for LQFP-EP 216-Pin package in sections 4.1.1 and 9.1. • Added notes for GPIO PG.11 ~ PG.15 pins in sectoins 4.2.1, 4.2.2 and 4.2.3. • Updated the pinout functions of GPIO PB.8 and PB.9 pins in section 4.2.1. • Updated the sections 6.24.2, 6.24.3 and Figure 5.1-1.
2025.12.15	1.12	• Added new part number MA35D16AI87C, MA35D16FH87C, MA35D16FI87C and MA35D16FJ87C in sections 3.1, 3.3, 4.2.1.2, 4.2.2 and 4.2.4. • Updated the sections 6.4.7, 6.4.8 and 6.26.7.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.