CV180ZB/CV1800B/CV1801B

Preliminary Datasheet

Version: 0.3.0.0

Release date: 2022-11-07

Revision History

Revision	Date	Description
0.1.0.0	2022/06/29	Preliminary release
0.2.0.0	2022/09/04	Remove Random Number Generator
0.3.0.0	2022/11/07	Part number update

Terms and Conditions

The document and all information contained herein remain the CVITEK Co., Ltd' s ("CVITEK") confidential information, and should not disclose to any third party or use it in any way without CVITEK' s prior written consent. User shall be liable for any damage and loss caused by unauthority use and disclosure.

CVITEK reserves the right to make changes to information contained in this document at any time and without notice.

All information contained herein is provided in "AS IS" basis, without warranties of any kind, expressed or implied, including without limitation mercantability, non-infringement and fitness for a particular purpose. In no event shall CVITEK be liable for any third party' s software provided herein, User shall only seek remedy against such third party. CVITEK especially claims that CVITEK shall have no liable for CVITEK' s work result based on Customer' s specification or published shandard.

Table of contents

Revision History 2
Terms and Conditions 3
Table of contents 4
Table of contens for figures 16
Table of contens for tables 19
1 Product Overview 21
1.1 Overview (CV180ZB/CV1800B/CV1801B) 21
1.2 Application Scenarios 21
1.2.1 CV180ZB/CV1800B/CV1801B Intelligent IP Camera Solutions 21
1.3 Architecture 22
1.3.1 Overview 22
1.3.2 Processor Core 23
1.3.3 TPU 23
1.3.4 Video Coding (CV180ZB/CV1800B/CV1801B) 24
1.3.5 Video Interface 24
1.3.6 ISP and Image Processing 25
1.3.7 CV Hardware Acceleration EngineCV 25
1.3.8 Audio Encoding and Decoding 25
1.3.9 Network Interface 26
1.3.10 Security System Module 26
1.3.11 Intelligent Secure Operating Environment 26
1.3.12 Peripheral Interfaces (CV180ZB/CV1800B/CV1801B) 26
1.3.13 External Memory Interface 27
1.3.14 SDK 27
1.3.15 Chip Physical Specifications 28
1.4 Boot and Upgrade Modes 28
1.4.1 Overview 28
1.4.2 Boot Mode and Corresponding Signal Latching Value Relationship 28
1.4.3 Image Burning Mode 29

1.4.4 Secure Boot 29
1.5 Address Space Mapping 30
2 Hardware Characteristics 35
2.1 Package and Pin Distribution 35
2.1.1 Package CV180ZB/CV1800B/CV1801B 35
2.1.2 Pin Distribution CV180ZB/CV1800B/CV1801B 36
2.2 Pin information Description 36
2.3 Welding Process Suggestions 38
2.4 Moisture Sensitivity Parameters 39
2.4.1 Moisture Barrier Packaging for CVITEK Products 39
2.5 Electrical Performance Parameters 42
2.5.1 Power Consumption Parameters 42
2.5.2 Temperature and Thermal Resistance Parameters (CV180ZB/CV1800B/CV1801B) 42
2.5.3 Destructive Voltage 43
2.5.4 Power Sequencing(CV180ZB/CV1800B/CV1801B) 43
2.5.5 The DC/AC Electrical Parameters of the Power Supply 46
2.5.6 1.8V I/O Electrical Parameters 47
2.5.7 18OD33 IO (VDDIO=1.8V) Electrical Parameters 47
2.5.8 18OD33 IO (VDDIO=3.0V) Electrical Parameters 49
2.5.9 Audio GPIO Electrical Parameters 50
2.5.10 ETH GPIO Electrical Parameters 50
2.5.11 MIPI Rx Electrical Parameters 50
2.5.12 Sub-LVDS Electrical Parameters 52
2.5.13 HiSPi Electrical Parameters 52
2.5.14 SDIO Electrical Parameters 53
2.5.15 2.5.15 VI RAW/BT.656/BT.1120Electrical Parameters(CV180ZB/CV1800B/CV1801B) 53
2.5.16 AUDIO CODEC Electrical Parameters 53
2.6 Timing 55
2.6.1 SPI NOR Timing 55
2.6.2 SPI NAND Timing 56
2.6.3 VI Timing 58

2.6.4 AIAO (I2S/PCM) Timing 58
2.6.5 I2C Timing 60
2.6.6 SPI Timing 62
2.6.7 MIPI Rx Timing 63
2.6.8 Sub-LVDS Timing 65
2.6.9 HiSPi Timing 66
2.6.10 SDIO/MMC Timing 67
3 System 72
3.1 Reset 72
3.1.1 Overview 72
3.1.2 Reset Control 72
3.1.3 Reset Configuration Register 74
3.2 Clock 78
3.2.1 Overview 78
3.2.2 Function Block Diagram 79
3.2.3 Clock Resource and Frequency Division Structure 79
3.2.4 PLL Configuration 80
3.2.5 CLK_DIV Clock Frequency Division Configuration 82
3.2.6 PLL CRG Register Overview 84
3.2.7 PLL CRG Register Overview 86
3.2.8 CLK_DIV CRG Register Overview 95
3.2.9 CLK_DIV CRG Register Overview 96
3.3 Processor Subsystem 114
3.4 Interrupt System 115
3.5 System Controller 116
3.5.1 Overview 116
3.5.2 Function Description 116
3.5.3 System Control Register 117
3.6 DMA Controller 123
3.6.1 Overview 123
3.6.2 Characteristics 124
3.6.3 Function Description 124
3.6.4 Working Mode 129

3.6.5 DMAC Register 131
3.7 Timer 183
3.7.1 Overview 183
3.7.2 Characteristics 183
3.7.3 Function description 183
3.7.4 Operation mode 184
3.7.5 Timer register overview 184
3.7.6 Timer register description 185
3.8 Watchdog 187
3.8.1 Overview 187
3.8.2 Charateristics 187
3.8.3 Function description 187
3.8.4 Working mode 189
3.8.5 WDT register overview 189
3.8.6 WDT register description 190
3.9 Real time clock 192
3.9.1 Overview 192
3.9.2 Features 193
3.9.3 Function description 193
3.9.4 Operation 195
3.9.5 RTC register overview 199
3.9.6 RTC register description 202
3.10 Power management and low power consumption mode 228
3.10.1 Overview 228
3.10.2 Clock control 229
3.10.3 DDR low power consumption control 229
3.10.4 Voltage regulation 230
3.11 Chip internal temperature detection 232
3.12 8051 subsystem 232
3.12.1 Overview 232
3.12.2 Characteristics 232
3.12.3 Function Description 234
3.12.4 Working Mode 234
3.12.5 8051 Subsystem Registers Overview 236

3.12.6 8051 Subsystem Registers 237
4 Memory Interface 250
4.1 DDR Controller 250
4.1.1 Overview 250
4.1.2 Characteristics 250
4.1.3 Function Description 251
4.1.4 Working Method 255
4.1.5 AXI Register 256
4.1.6 DDRC Register 292
4.2 SPI NOR Flash Controller 294
4.2.1 Overview 294
4.2.2 Characteristic 294
4.2.3 Function Description 294
4.2.4 Workflow 299
4.2.5 Register overview 301
4.2.6 Register Discription 301
4.3 SPI NAND Flash Controller 305
4.3.1 Overview 305
4.3.2 Characteristics 305
4.3.3 Function Description 306
4.3.4 Operation Flow 309
4.3.5 Data Structure (NAND Flash/SPI NAND Flash) 311
4.3.6 Register Overview 312
4.3.7 Register description 312
5 Network interface 315
5.1 Ethernet MAC 315
5.1.1 Overview 315
5.1.2 Function description 315
5.1.3 Data flow overview 315
5.1.4 Single port function configuration description 316
5.1.5 Ethernet transceiver frame management function 316
5.1.6 Ethernet packet receiving interrupt management function 316
5.1.7 Configure the working state of PHY chip 317

5.1.8 Working mode switching 317
5.1.9 Typical applications 318
5.1.10 Register offset address description 318
5.1.11 GMAC register overview 318
5.1.12 GMAC register description 319
5.2 Ethernet PHY 323
5.2.1 Overview 323
5.2.2 Function description 323
5.2.3 Functional block diagram 323
6 Video and image codec 324
6.1 Overalll overview 324
6.2 VCU (Video Codec Unit) 324
6.2.1 Overview 324
6.2.2 Features 324
6.2.3 Function description 325
6.3 JCU (JPEG Codec Unit) 328
6.3.1 Overview 328
6.3.2 Features 328
6.3.3 Function description 329
7 Video and graphics processing 332
7.1 VPSS (sc_top) 332
7.1.1 Overview 332
7.1.2 Function description 332
7.2 LDC (Lens Distortion Correction) 333
7.2.1 Overview 333
7.2.2 Function description 333
8 AI engine 335
8.1 TPU (Tensor Processing Unit) 335
8.1.1 Overview 335
8.1.2 Characteristics 336
9 Video interface 338
9.1 VI 338

9.1.1 Overview 338
9.1.2 Characteristics 338
9.1.3 Mode function description 339
9.1.4 Image storage mode 344
9.1.5 VI register overview 344
9.1.6 VI register overview 346
9.2 MIPI Rx 359
9.2.1 Overview 359
9.2.2 Charateristics 360
9.2.3 Function Description 360
9.2.4 MIPI Rx Register Overview 375
9.2.5 MIPI RxRegister Overview 378
10 ISP 399
10.1 Function Overview 399
10.2 Overview 400
10.2.1 Function Block Diagram 400
10.2.2 Working Mode 403
10.3 ISP Interruption System 403
10.3.1 Function Description 403
10.3.2 Interrupt Timing 405
10.4 Module Function 406
10.4.1 Color_bar (patgen) 406
10.4.2 Crop 406
10.4.3 AE (Auto Exposure) 407
10.4.4 AF (Auto Focus) 407
10.4.5 DIS (Digital image stabilization) 408
10.4.6 BLC (Black level correction) 408
10.4.7 DG (Digital Gain) 408
10.4.8 DPC 408
10.4.9 GE 409
10.4.10 LSC (Lens shading correction) 409
10.4.11 DRC (LTM) 409
10.4.12 WBG (White Balance Gain) 409

10.4.13 BNR (Bayer Noise Reduction) 409
10.4.14 DEMOSAIC (CFA) 410
10.4.15 CCM 410
10.4.16 Gamma 410
10.4.17 Dehaze 410
10.4.18 CSC 411
10.4.19 3DNR (3-Dimensional Noise Reduction) 411
10.4.20 YNR 411
10.4.21 LDCI (DCI) 411
10.4.22 Sharpen 411
10.4.23 CNR 412
10.4.24 CAC (PFC inside CNR) 412
10.4.25 CLUT (HSV_3D_LUT) 412
10.4.26 RGBCAC 412
10.4.27 PREYEE 412
10.4.28 Hist_V 412
10.4.29 CACP 412
10.4.30 CA2 413
10.4.31 LCAC 413
10.4.32 User Gamma 413
11 Audio interface 414
11.1 AIAO 414
11.1.1 Overview 414
11.1.2 Features 415
11.1.3 Function description 416
11.1.4 Operation control 420
11.1.5 AIAO register overview 420
11.1.6 AIAO register description 421
11.2 Audio Codec 435
11.2.1 Overview 435
11.2.2 Characteristics 435
11.2.3 Audio Codec register overview 436
12 Peripherals equipment 443

12.1 I2C 443
12.1.1 Overview 443
12.1.2 Function description 443
12.1.3 Function block diagram 443
12.1.4 I2C Agreement timing 444
12.1.5 Working mode 445
12.1.6 I2C register overview 447
12.1.7 I2C register description 449
12.2 UART 457
12.2.1 Overview 457
12.2.2 Characteristics 457
12.2.3 Function description 458
12.2.4 Working mode 460
12.2.5 UART register overview 465
12.2.6 UART register description 466
12.3 SPI 472
12.3.1 Overview 472
12.3.2 Characteristics 472
12.3.3 Function description 473
12.3.4 Working mode 473
12.3.5 Three kinds of serial peripheral bus sequence diagram 478
12.3.6 Register Overview 482
12.3.7 Register Description 483
12.4 SD/SDIO Controller 495
12.4.1 Function Description 495
12.4.2 Application Explanation 503
12.4.3 Register Overview 521
12.4.4 Register Description 522
12.5 GPIO 536
12.5.1 Overview 536
12.5.2 Characteristics 536
12.5.3 Working Mode 536
12.5.4 GPIORegister Overview 537
12.5.5 GPIO Register Description 538

12.6 USB DRD (Dual Role Device) 541
12.6.1 Overview 541
12.6.2 Function Description 542
12.6.3 USBC Function and Register Description 543
12.6.4 The Illustration of Host Initialization Program 582
12.6.5 Host Register Description 583
12.6.6 Device Initialization Program 592
12.6.7 Device Register Description 593
12.7 SARADC 603
12.7.1 Overview 603
12.7.2 Features 603
12.7.3 Working method 604
12.7.4 SARADC register overview 604
12.7.5 SARADC register description 604
12.8 Temperature sensor 607
12.8.1 Overview 607
12.8.2 Working method 607
12.8.3 Temperature sensor register overview 610
12.8.4 Temperature sensor register description 610
12.9 PWM 614
12.9.1 Overview 614
12.9.2 Features 614
12.9.3 Operation 615
12.9.4 PWM Register Overview 617
12.9.5 PWM Register Description 618
12.10 Key scan 622
12.10.1 Overview 622
12.10.2 Working Method 622
12.10.3 Basic Setting 624
12.10.4 Using FIFO mode 624
12.10.5 Using snapshot array mode 624
12.10.6 Key scan Register Overview 625
12.10.7 Key scan Register Description 625
12.11 Wiegand 629

12.11.1 Overview 629
12.11.2 Working method 631
12.11.3 Wiegand Register Overview 632
12.11.4 Wiegand Register Description 633
12.12 IRRX Infrared Interface 638
12.12.1 Overview 638
12.12.2 Characteristics 638
12.12.3 Working Mode 638
12.12.4 IRRX Register Overview 638
12.12.5 IRRX Register Description 640
13 Security Subsystem Module 649
13.1 CryptoDMA 649
13.1.1 Overview 649
13.1.2 Function Characteristics 650
13.1.3 DMA Function Description 650
13.1.4 Function description of symmetric key algorithm block encryption mode 651
13.1.5 CryptoDMA Register Overview 653
13.1.6 CryptoDMA Register Overview 655
13.2 Secure Debug Firewall 666
13.2.1 Overview 667
13.2.2 Status Inquiry and Password Input Interface (I2C) 667
13.2.3 Status Inquiry and Password Input Process 669
13.3 Efuse Controller 669
13.3.1 Overview 670
13.3.2 Efuse entity address translation and virtual register address 671
13.3.3 Efuse Ctrl Register Overview 676
13.3.4 Efuse CTRL Register Overview 678
13.3.5 eFuse CTRL Operation Process 687
14 Intelligent Secure Operation Environment 689
14.1 Establishment of Trust Chain 689
14.2 Data Encryption Security 690
14.3 Software and Hardware verification 691

14.4 Secure Storage and Transmission 691
14.5 Security Update 691

Table of contens for figures

Figure 1-1 intelligent IP camera solutions. 22
Figure 1-2 CV180ZB/CV1800B/CV1801B architeture 23
Figure 2-1 CV180ZB/CV1800B/CV1801B package dimensions, top view 35
Figure 2-2. CV180ZB/CV1800B/CV1801B package dimensions, bottom view 35
Table 2-3 CV180ZB/CV1800B/CV1801B pin distribution 36
Figure 2-4 Lead-free Reflow Soldering process curve 38
Figure 2-5 Vacuum drying packaging information 39
Figure 2-6 Desiccant packs, humidity card, chip, and tray 40
Figure 2- 7 I2C Timing Diagram 60
Figure 2- 8 SPI Timing Diagram 62
Figure 3- 1 Reset Management Module Block Diagram 72
Figure 3- 2 Clock Management Block Diagram 79
Figure 3- 3 Clock Source Frequency Division Diagram 80
Figure 3- 4 DMAC Hardware Control Fow Diagram 123
Figure 3- 5 DMA Transmission Structure 125
Figure 3- 6 Linked List Relative Address and Data Format 127
Figure 3- 7 Interrupt Status and Source Diagram 128
Figure 3- 8 WatchDog Application block diagram 188
Chart 3- 9 Example of using PWM to control DCDC voltage. 231
Figure 4- 1 SoC/DRAM Interconnection Diagram 251
Figure 4- 2 Standard SPI Interface Mode Write Operation Sequence 295
Figure 4- 3 Standard SPI Interface Mode Read Operation Sequence 295
Figure 4- 4 Dual-Input SPI Interface Sequence 296
Figure 4- 5 Dual-IO SPI Interface Sequence 296
Figure 4- 6 Quad-Input SPI Interface Sequence 297
Figure 4- 7 Quad-IO SPI Interface Sequence 297
Figure 4- 8 Standard SPI Write Operation Timing 306
Figure 4- 9 Standard SPI Read Operation Timing 307
Figure 4- 10 SPI Nand x2 Interface Mode Operation Sequence 307
Figure 4- 11 SPI Nand x4 Interface Mode Operation Sequence 308

Chart 5- 1 GMAC conceptual data stream 316
Chart 5- 2 built-in 10/100 Ethernet PHY chart 323
Figure 6- 1 VENC function block diagram 326
Figure 8- 1 TPU working mode diagram 335
Figure 9- 1 VI function block diagram 338
Figure 9- 2 DC Synchronous signal mode -vhs mode 342
Figure 9- 3 VI register overview 344
Figure 9- 4 MIPI Rx Functional Block Diagrams and Position 360
Figure 9- 5 Interface Types Supported by MIPI Rx 361
Figure 9- 6 Transmission Mechanism of Data Packet 362
Figure 9- 7 CSI-2 Long Packet Format 363
Figure 9- 8 CSI-2 Short Packet Format 364
Figure 9- 9 YUV422 8-bit Frame Format 366
Figure 9- 10 YUV422-10bit Data Transmission Sequence 366
Figure 9- 11 YUV422-10bit Data Packet Transmission Correspondence 367
Figure 9- 12 YUV422-10bit Frame Format 367
Figure 9- 13 RAW8 Data Transmission Sequence 368
Figure 9- 14 RAW8 Frame Format 368
Figure 9- 15 RAW10 Data Transmission Sequence 368
Figure 9- 16 RAW10 Frame Format 369
Figure 9- 17 RAW12Data Transmission Sequenc 369
图表 9-18 RAW16Data Transmission Sequenc 370
图表 9-19 RAW16 Frame Format 370
Figure 9- 20 MIPI Interface Image Format 372
Figure 9- 21 MIPI Interface Wide Dynamic Data Transfer (using DT) 373
Figure 9- 22 MIPI Interface Wide Dynamic Data Transfer (using ID) 374
Figure 9- 23 MIPI Interface Wide Dynamic Data Transfer (register setting) 375
Figure 9- 24 MIPI Rx PHY Register Overview 375
Table 9-25 MIPI Rx CSI control registers overview 377
Table 9-26 MIPI Rx Sub-LVDS control registers overview 378
Figure 10- 1 ISP Overall Structure Diagram 400
Figure 10- 2 Pre_raw_fe and Pre_raw_be Module Diagram 401
Figure 10- 3 raw_top Module Diagram 402
Figure 10- 4 rgb_top Module Diagram 402

Figure 10- 5 yuv_top Module Diagram 403
Figure 10- 6 ISP Timing Diagram when Interrupt occurs 406
Figure 10- 7 Image Cutting Diagram 407
Figure 12- 1 TI SSP Single Frame Transmission Format 480
Figure 12- 2 TI SSP Continuous Frame Transmission Format 481
Figure 12- 3 NS Microwire Single Frame Transmission Format 481
Figure 12- 4 NS Microwire Continuous Frame Transmission Format 482
Figure 12- 5 Single Block and Multi Block Read Operation 501
Figure 12- 6 Clock Configuration Flow Chart 505
Figure 12- 347 Abort Command Sequence 511
Figure 12- 358 Asynchronous Abort Command Procedure 512
Figure 12- 9 Wiegand 26 Format 629
Figure 13- 1 Security Subsystem Module 649
Figure 13- 2 ECB Mode 651
Figure 13- 3 CBC Mode 652
Figure 13- 4 CTR Mode 653
Figure 13- 5 eFuse CTRL Module architecture 671
Figure 14- 1 Establishment of trust chain 690

Table of contens for tables

Table2-1 Lead-free reflow soldering process parameters 38
Table 2-2 Humidity classification and floor life 40
Table 2-3 Baking temperature and time table 41
Table 2-4 Thermal resistance parameters for CV180ZB/CV1800B/CV1801B 42
Table 2-5 Temperature-related parameters 42
Table 2-6 Destructive voltage parameters (CV180ZB/CV1800B/CV1801B) 43
Table 2- 7 The power supply electrical parameters of CV180ZB/CV1800B/CV1801B (Recommended Operating Conditions) 46
Table 2- 8 1.8V I/O Electrical Parameters 47
Table 2- 9 18OD33 IO (VDDIO=1.8V) Electrical Parameters 48
Table 2- 10 18OD33 IO (VDDIO=3.0V) Electrical Parameters 49
Table 2- 11 Audio GPIO Electrical Parameters 50
Table 2- 12. MIPI D-PHY High Speed(MISH) DifferentialDC Electrical Parameters 51
Table 2- 13. MIPI D-PHY High Speed(MIHS) Differential AC Electrical Parameters 51
Table 2- 14 MIPI D-PHY Low Power(MILP) Differential DC Electrical Parameters 51
Table 2- 15 MIPI D-PHY Low Power(MILP) Differential AC Electrical Parameters 52
Table 2- 16 Sub-LVDS(SL) Differential DC Electrical Parameters 52
Table 2- 17 Sub-LVDS(SL) Differential AC Electrical Parameters 52
Table 2- 18 HiSPi Differential DC Electrical Parameters 53
Table 2- 19 Audio CODEC Overall Index 53
Table 2- 20 Audio DAC Electrical Parameters 53
Table 2- 21 Audio ADC Electrical Parameters 54
Table 2- 22 SPI NAND Input Timing Requirements 56
Table 2- 23 SPI NAND Output Timing Requirements 57
Table 2- 24 SPI Timing Requirements 62
Table 3- 1 PLL Configuration Parameters 80
Table 3- 2 Interger PLL Configuration Parameters 81
Table 3- 3 Fractional PLL Configuration Parameters 82
Table 3- 4 Clock Source and Preset Frequency Division Parameters 83
Table 3- 5 Interrupt Number and Interrupt Source Mapping Table 115

Table 3- 6 DMAC Access Space Type 124
Table 3- 7 Timer register overview (address 0x030A0000) 184
Table 3- 8 WDT register overview (address 0x03010000) 190
Table 3- 9 RTC_REG register overview (base address: 0x05026000) 199
Table 4- 1 DDR2 Command Truth Table 252
Table 4- 2 DDR3 Command Truth Table 253
Table 5- 1 GMAC register overview 318
Table 10- 1 Interrupt Indication Register 403
Table 11- 1 AIAO subsystem register overview (base address: 0x0410_8000) 420
Table 11- 2 I2S_TDM_0/1/2/3 register overview (address 0x0410_0000 + n*0x10000) 421
Table 12- 1 SD3.0 Supported Speed and Voltage 502
Table 12- 2 SD Register Overview 521
Table 12- 3 Four GPIO Module Base Addresses of the Chip 537
Table 13- 1 Status Query I2C Interface Register Address 668
Figure 13- 2 eFuse entity (row) address and logical (row) address corresponding value 672

1 Product Overview

1.1 Overview (CV180ZB/CV1800B/CV1801B)

CV180ZB/CV1800B/CV1801B is a high-performance, low-power consumption chip designed for various consumer monitoring IP cameras, home intelligence, and other products. It integrates an H.264/H.265 video compression encoder and ISP and supports various image enhancement and correction algorithms such as digital wide dynamic range, 3D noise reduction, defogging, and lens distortion correction to provide professional-grade video image quality to customers.

The chip also integrates a self-developed intelligent reference design (human form detection, area detection, motion detection), built-in DDR, and complete peripherals and external devices, providing a high-integration and concise solution to facilitate customer product development and mass production.

In addition, it also provides secure boot, secure update, and secure encryption and so on to provide a series of security solutions for users from development, mass production to product application.

The chip integrates an 8-bit MCU subsystem, which can replace general external MCUs to achieve the goal of saving BOM cost and power consumption.

1.2 Application Scenarios

1.2.1 CV180ZB/CV1800B/CV1801B Intelligent IP Camera Solutions

The typical application scenarios for intelligent IP camera solutions are shown in Figure 1-1.

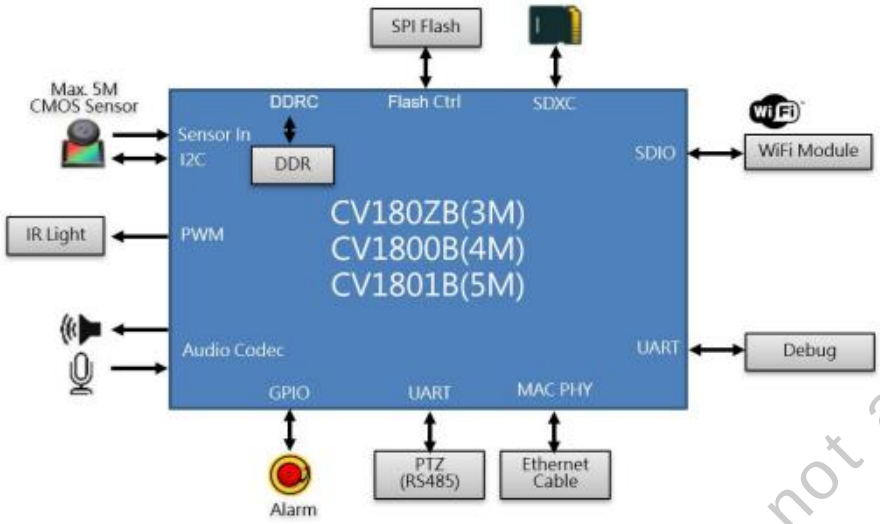


Figure 1-1 intelligent IP camera solutions.

1.3 Architecture

1.3.1 Overview

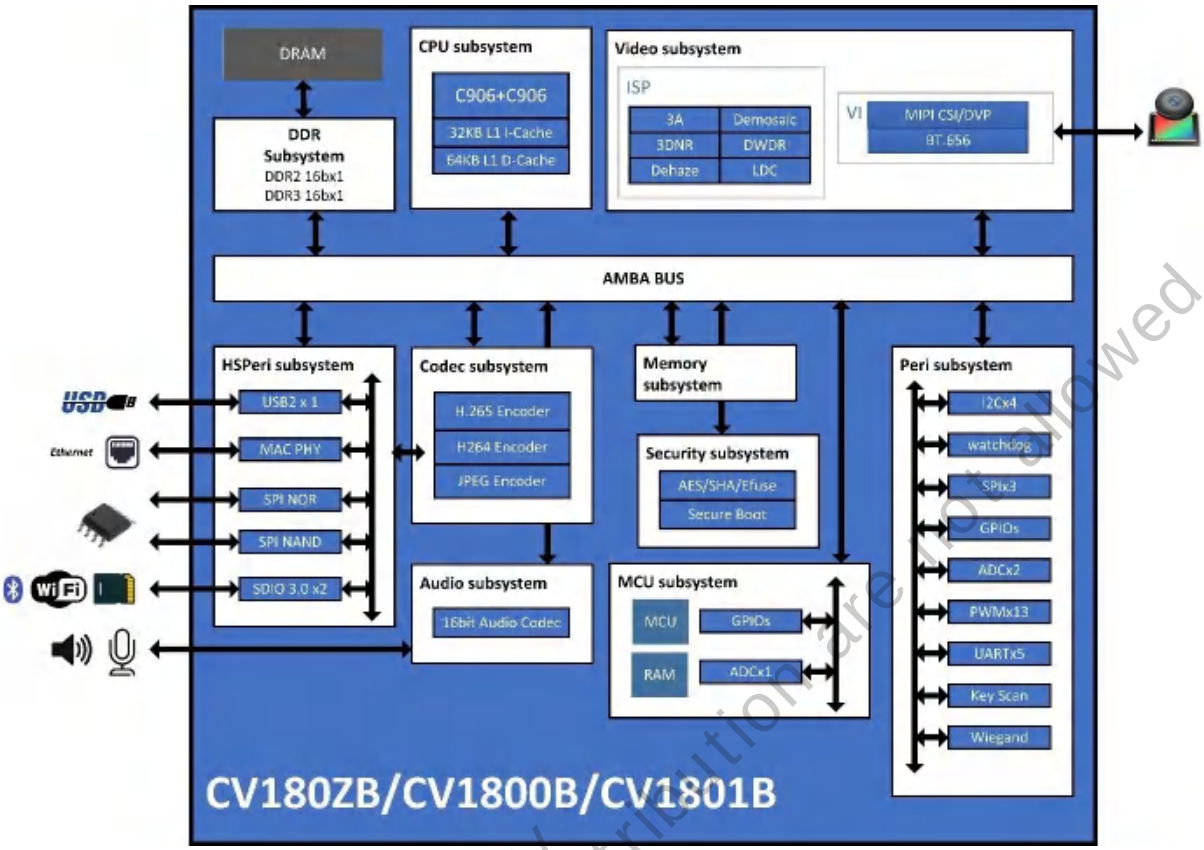


Figure 1-2 CV180ZB/CV1800B/CV1801B architeture

1.3.2 Processor Core

Main processor RISCV C906 @ 1.0Ghz .

- 32KB I-cache, 64KB D-Cache
- Integrated vector and floating-point processing units (FPU).

Coprocessor RISCV C906 @ 700Mhz

- Integrated floating-point processing units (FPU).

1.3.3 TPU

Built-in CVITEK TPU, integrated intelligent reference design (human form detection, area detection, motion detection)

1.3.4 Video Coding (CV180ZB/CV1800B/CV1801B)

H.264 Baseline/Main/High profile

H.265 Main profile

H.264/H.265 both support I-frames and P-frames.

MJPEG/JPEG baseline

H.264/H.265 has maximum encoding resolution : 2880x1620 (5M) (CV1801B)

H.264/H.265 has maximum encoding resolution : 2688x1520 (4M) (CV1800B)

H.264/H.265 has maximum encoding resolution : 2304x1296 (3M) (CV180ZB)

H.264/H.265 's encoding performance:

- 2880x1620@20fps+720x576@20fps (CV1801B)
- 2688x1520@25fps+720x576@25fps (CV1800B)
- 2304x1296@25fps+720x576@30fps (CV180ZB)

JPEG 's maximum encoding and decoding performance:

- 2880x1620@20fps (CV1801B)
- 2688x1520@25fps (CV1800B)
- 2304x1296@30fps (CV180ZB)

Support multiple bitrate control modes, including CBR/VBR/FIXQP.

Support Region of Interest (ROI) encoding.

1.3.5 Video Interface

The Input

- Support simultaneous one-channel video input (MIPI 4L).
- Support MIPI serial interface.
- Support 8/10/12 bit RGB Bayer video input.
- Support BT.656,BT.601
- Support high-definition CMOS sensors such as SONY, OnSemi, OmniVision.
- Provide programmable frequency output for the sensor to use as a reference clock.
- Support maximum width: 2304 , maximum resolution: 2304x1296 (CV180ZB)
- Support maximum width: 2688 , maximum resolution: 2688x1520 (CV1800B)
- Support maximum width: 2880 , maximum resolution: 2880x1620 (CV1801B)

1.3.6 ISP and Image Processing

Support image or video rotation by 90, 180, or 270 degrees;

Support horizontal(Flip) or vertical(Mirror) flipping of image or video;

Support overlaying two layers of OSD (On-Screen Display) on the video;

Support video scaling down to 1/32 or up to 32 times;

Support 3A algorithm: automatic exposure(AE), automatic white balance(AWB), and automatic autofocus(AF) ;

Support fixed-mode noise reduction and bad pixel correction;

Support correction of lens shading, distortion, and purple fringing;

Support direction-adaptive demosaic algorithm that selects the best demosaic algorithm based on the image orientation;

Support Gamma correction, dynamic contrast enhancement, and color management algorithms;

Support regional adaptive defogging;

Support bayer denoising, 3D denoising, detail enhancement, and sharpening enhancement;

Support local Tone mapping

Support sensor with wide dynamic range and 2-frame wide dynamic range ;

Support two-axis digital image stabilization;

Support lens distortion correction;

Provide PC-side ISP tuning tools.

.

1.3.7 CV Hardware Acceleration EngineCV

Mixed mode of software and hardware supports partial OpenCV library.

1.3.8 Audio Encoding and Decoding

Integrated Audio CODEC, supporting 16-bit audio/speech input and output;

Integrated mono microphone input;

Integrated mono output (external amplifier required to drive the speaker);

Internally integrated another microphone direct output channel for easy implementation of AEC;

Software audio codec protocols (G.711, G.726, ADPCM);
Software supports audio 3A (AEC, ANR, AGC) functions;

1.3.9 Network Interface

The Ethernet module provides one Ethernet MAC for receiving and transmitting network data. The Ethernet MAC, combined with the built-in 10/100Mbps Fast Ethernet Transceiver, can operate in 10/100Mbps full-duplex or half-duplex mode.

1.3.10 Security System Module

Hardware implementation of multiple encryption and decryption algorithms such as AES/DES/SM4;
Hardware implementation of hashing algorithms such as HASH (SHA1/SHA256);
Internal integration of 2Kbit eFuse logical space.

1.3.11 Intelligent Secure Operating Environment

Supports secure boot, provides secure hardware and software protection functions;
Supports data encryption security: data encryption program, operation core encryption;
Supports software and firmware signature verification process: confirms software trustworthiness and integrity, including boot and loading signature verification program;
Supports secure storage and transmission: protects external data storage and exchange;
Supports secure updates.

1.3.12 Peripheral Interfaces (CV180ZB/CV1800B/CV1801B)

Integrated POR, Power sequence;

3 single-ended ADCs (3 in MCU domain);

4 I2C interfaces (1 in mcu domain);

3 SPI interfaces;

5 groups of UART (1 in mcu domain) ;

4 groups of (16 channels) PWM;

2 SDIO interfaces:
One supports 3V connection for SD 3.0 Card (supporting a maximum capacity of SDXC 2TB and speed up to UHS-I);
One supports 3.0V connection for other SDIO 3.0 devices (supporting speed up to UHS-I).

51 GPIO interfaces (9 in MCU domain);

Integrated keyscan and Wiegand;

Integrated MAC PHY supporting 10/100Mbps full-duplex or half-duplex mode;

One USB Host/device interface.

1.3.13 External Memory Interface

Built-in DRAM

CV180ZB/CV1800B DDR2 16bitx1, with a maximum speed of 1333Mbps , capacity of 512Mbit (64MB)

CV1801B DDR3 16bitx1, with a maximum speed of 1866Mbps, capacity of 1Gbit (128MB)

SPI NOR flash interface (1.8V / 3.0V)

Supports 1, 2, 4-wire mode.

Maximum support for 256MByte.

SPI Nand flash interface (1.8V / 3.0V)

Support 1KB/2KB/4KB page (corresponding maximum capacity 16GB/32GB/64GB)

Uses the built-in ECC module of the device.

1.3.14 SDK

Linux-5.10-based SDK

1.3.15 Chip Physical Specifications

Power Consumption

1080P + Video encode + AI : ~ 500mW

Other scenarios : TBD

Operating Voltage

Core voltage: 0.9V

IO voltage: 1.8V and 3.0V

DDR voltage as shown in the table below:

CV180ZB/CV1800B = 1.8V

CV1801B = 1.35V

Package

QFN package is used, with a package size of 7mmx7mmx0.9mm. The pin pitch is 0.35mm, and the total number of pins is 68.

1.4 Boot and Upgrade Modes

1.4.1 Overview

The chip is booted by the built-in ROM (BOOTROM). When the chip is reset, it detects whether there is a weak pull-up or weak pull-down on two pins (SPINOR_MOSI, SPINOR_WP_X) to determine the type of memory device currently in use.

For chips with secure boot, software execution or upgrades will be verified during startup and chip upgrades to ensure that the software being executed or upgraded is secure.

1.4.2 Boot Mode and Corresponding Signal Latching Value Relationship

Supports booting from SPI Nor Flash (SPINOR_WP_X pull down, SPINOR_MOSI pull up)

Supports booting from SPI Nand Flash (SPINOR_WP_X pull down, SPINOR_MOSI pull down)

1.4.3 Image Burning Mode

Supports burning image through SD card;
Supports burning image through USB device mode;
If the image already exists in flash, software supports upgrading through the network.

1.4.4 Secure Boot

Supports secure boot and upgrade;
AES hardware encryption and decryption ;
SHA/Secure Efuse security hardware.

1.5 Address Space Mapping

Start Address [31:0]	End Address [31:0]	Space Function	Space Size(Byte)
0x01000000	0x018FFFFF	reserve	
0x01900000	0x01900FFF	ap_mailbox	4K
0x01901000	0x01901FFF	ap_system_ctrl	4K
0x01902000	0x019EFFFF	reserve	
0x01A00000	0x01FFFFFF	reserve	
0x02000000	0x02FFFFFF	reserve	64K
0x03000000	0x03000FFF	TOP_MISC control register	4K
0x03001000	0x03001FFF	PINMUX control register	4K
0x03002000	0x03002FFF	CLKGEN/PLL control register	4K
0x03003000	0x03003FFF	RSTGEN control register	4K
0x03004000	0x03005FFF	reserve	
0x03006000	0x03006FFF	reserve	4K
0x03007000	0x03008FFF	reserve	
0x03009000	0x03009FFF	reserve	4K
0x0300A000	0x0300AFFF	reserve	4K
0x0300B000	0x0300FFFF	reserve	
0x03010000	0x03010FFF	WATCH DOG0 control register	4K
0x03011000	0x03011FFF	WATCH DOG1 control register	4K
0x03012000	0x03012FFF	WATCH DOG2 control register	4K
0x03020000	0x03020FFF	GPIO0 control register	4K
0x03021000	0x03021FFF	GPIO1 control register	4K
0x03022000	0x03022FFF	GPIO2 control register	4K
0x03023000	0x03023FFF	GPIO3 control register	4K
0x03024000	0x0302FFFF	reserve	
0x03030000	0x03030FFF	WGN0 control register	4K
0x03031000	0x03031FFF	WGN1 control register	4K
0x03032000	0x03032FFF	WGN2 control register	4K
0x03033000	0x0303FFFF	reserve	
0x03040000	0x0304FFFF	KEYSCAN control register	64K
0x03050000	0x0305FFFF	EFUSE control register	64K
0x03060000	0x03060FFF	PWM0 control register	4K
0x03061000	0x03061FFF	PWM1 control register	4K
0x03062000	0x03062FFF	PWM2 control register	4K
0x03063000	0x03063FFF	PWM3 control register	4K
0x03064000	0x0309FFFF	reserve	
0x030A0000	0x030AFFFF	TIMER control register	64K

0x030C0000	0x030CFFFF	reserve	
0x030D0000	0x030D0FFF	reserve	4K
0x030D1000	0x030D1FFF	reserve	4K
0x030D2000	0x030D2FFF	reserve	4K
0x030D3000	0x030DFFFF	reserve	
0x030E0000	0x030EFFFF	TEMPSEN control register	64K
0x030F0000	0x030FFFFF	SARADC control register	64K
0x04000000	0x0400FFFF	I2C0 control register	64K
0x04010000	0x0401FFFF	I2C1 control register	64K
0x04020000	0x0402FFFF	I2C2 control register	64K
0x04030000	0x0403FFFF	I2C3 control register	64K
0x04040000	0x0404FFFF	I2C4 control register	64K
0x04050000	0x0405FFFF	reserve	
0x04060000	0x0406FFFF	SPI_NAND control register	64K
0x04070000	0x0407FFFF	ETH0 control register	
0x04080000	0x040FFFFF	reserve	
0x04100000	0x04107FFF	I2S0 control register	64K
0x04108000	0x0410FFFF	I2S Global control register	64K
0x04110000	0x0411FFFF	I2S1 control register	64K
0x04120000	0x0412FFFF	I2S2 control register	64K
0x04130000	0x0413FFFF	I2S3 control register	64K
0x04140000	0x0414FFFF	UART0 control register	64K
0x04150000	0x0415FFFF	UART1 control register	64K
0x04160000	0x0416FFFF	UART2 control register	64K
0x04170000	0x0417FFFF	UART3 control register	64K
0x04180000	0x0418FFFF	SPI0 control register	64K
0x04190000	0x0419FFFF	SPI1 control register	64K
0x041A0000	0x041AFFFF	SPI2 control register	64K
0x041B0000	0x041BFFFF	SPI3 control register	64K
0x041C0000	0x041CFFFF	UART4 control register	64K
0x041D0000	0x041DFFFF	AUDSRC control register	64K
0x041E0000	0x042FFFFF	reserve	
0x04300000	0x0430FFFF	reserve	
0x04310000	0x0431FFFF	SD0 control register	64K
0x04320000	0x0432FFFF	SD1 control register	
0x04330000	0x0433FFFF	DMA control register	64K
0x04340000	0x0434FFFF	USB control register	64K
0x04350000	0x043FFFFF	reserve	
0x04400000	0x0440FFFF	ROM 内存空间	64K
0x04410000	0x04FFFFFF	reserve	

0x05000000	0x05000FFF	reserve	4KB
0x05020000	0x05020FFF	RTCSYS_Timer control register	4KB
0x05021000	0x05021FFF	RTCSYS_GPIO control register	4KB
0x05022000	0x05022FFF	RTCSYS_UART control register	4KB
0x05023000	0x05023FFF	RTCSYS_INTR control register	4KB
0x05024000	0x05024FFF	RTCSYS_MBOX control register	4KB
0x05025000	0x05025FFF	RTCSYS_CTRL control register	4KB
0x05026000	0x05026FFF	RTCSYS_CORE	4KB
0x05027000	0x05027FFF	RTCSYS_IO control register	4KB
0x05028000	0x05028FFF	RTCSYS_OSC control register	4KB
0x05029000	0x05029FFF	reserve	4KB
0x0502A000	0x0502AFFF	RTCSYS_32kless control register	4KB
0x0502B000	0x0502BFFF	RTCSYS_I2C control register	4KB
0x0502C000	0x0502CFFF	RTCSYS_SAR control register	4KB
0x0502D000	0x0502DFFF	RTCSYS_WDT control register	4KB
0x0502E000	0x0502EFFF	RTCSYS_IRRXcontrol register	4KB
0x05200000	0x053FFFFF	RTCSYS_SRAM	8KB
0x05400000	0x057FFFFF	RTCSYS_SPINOR	4MB
0x08000000	0x08001FFF	reserve	8K
0x08004000	0x08005FFF	DDR Controler control register	8K
0x08006000	0x08007FFF	reserve	8K
0x08008000	0x08009FFF	DDR AXI Monitor control register	8K
0x0800A000	0x0800BFFF	DDR Global control register	8K
0x08010000	0x08011FFF	reserve	8K
0x08012000	0x08013FFF	reserve	8K
0x08014000	0x09FFFFFF	reserve	
0x0A000000	0x0A07FFFF	ISP control register	512K
0x0A080000	0x0A0803FF	sc_top control register	1K
0x0A080400	0x0A080BFF	reserve	2K
0x0A080C00	0x0A080CFF	osd enc control register	256B
0x0A080D00	0x0A080FFF	reserve	768B
0x0A081000	0x0A081FFF	reserve	4K
0x0A082000	0x0A082FFF	img_v control register	4K
0x0A083000	0x0A083FFF	img_d control register	4K
0x0A084000	0x0A084FFF	sc_d control register	4K
0x0A085000	0x0A085FFF	sc_v1 control register	4K
0x0A086000	0x0A086FFF	sc_v2 control register	4K
0x0A087000	0x0A087FFF	reserve	4K
0x0A088000	0x0A088FFF	reserve	4K
0x0A089000	0x0A089FFF	reserve	4K

0x0A08A000	0x0A08AFFF	reserve	4K
0x0A08B000	0x0A08BFFF	cmdq control register	4K
0x0A08C000	0x0A08CFFF	reserve	4K
0x0A08D000	0x0A08DFFF	reserve	4K
0x0A08E000	0x0A09FFFF	reserve	72K
0x0A0A0000	0x0A0AFFFF	reserve	64K
0x0A0A0000	0x0A0BFFFF	reserve	64K
0x0A0C0000	0x0A0C1FFF	ldc control register	8K
0x0A0C2000	0x0A0C3FFF	VIO/MIPI_RX0 control register	8K
0x0A0C4000	0x0A0C5FFF	reserve	8K
0x0A0C6000	0x0A0C7FFF	reserve	8K
0x0A0C8000	0x0A0C9FFF	VIPSYS control register	8K
0x0A0CA000	0x0A0CFFFF	reserve	24K
0x0A0D0000	0x0A0D0FFF	CSI_PHY control register	4K
0x0A0D1000	0x0A0D1FFF	reserve	4K
0x0A0D2000	0x0AFFFFFF	reserve	
0x0B000000	0x0B00FFFF	JPEG codec control register	64K
0x0B010000	0x0B01FFFF	H.264 codec control register	64K
0x0B020000	0x0B02FFFF	H.265 codec control register	64K
0x0B030000	0x0BFFFFFF	reserve	
0x0C000000	0x0FFFFFFF	reserve	
0x10000000	0x1FFFFFFF	SPI_NOR 内存空间	256M
0x20000000	0x7FFFFFFF	reserve	
0x80000000	0xFFFFFFFF	DDR 内存空间	2G

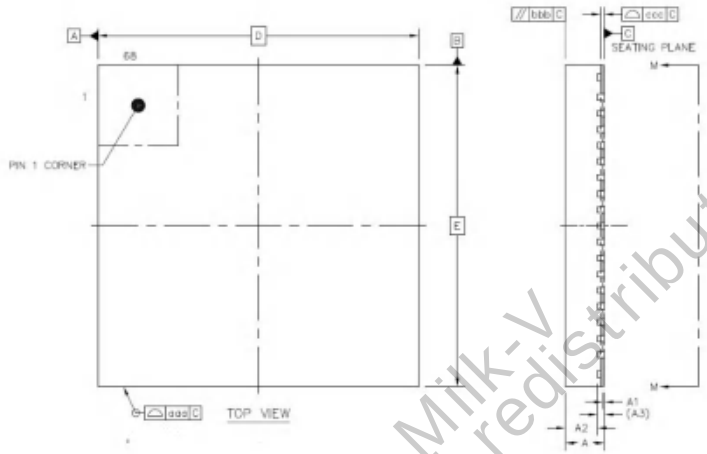
* Performing read/write operations on the reserved address space may result in unpredictable results.

2 Hardware Characteristics

2.1 Package and Pin Distribution

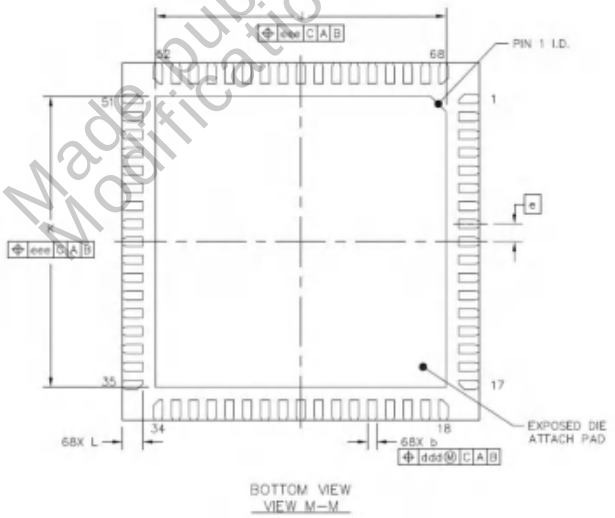
2.1.1 Package CV180ZB/CV1800B/CV1801B

CV180ZB/CV1800B/CV1801B uses QFN package with a package size of 7mmx7mmx0.9mm. The pin pitch is 0.35mm and there are a total of 68 pins. For detailed package dimensions, please refer to the figure below.



		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	0.8	0.85	0.9
STAND OFF		A1	0	0.035	0.05
MOLD THICKNESS		A2	---	0.7	---
L/F THICKNESS		A3		0.152 REF	
LEAD WIDTH		b	0.13	0.18	0.23
BODY SIZE	X	D		7 BSC	
	Y	E		7 BSC	
LEAD PITCH		e		0.35 BSC	
EP SIZE	X	J	5.6	5.7	5.8
	Y	K	5.6	5.7	5.8
LEAD LENGTH		L	0.35	0.4	0.45
PACKAGE EDGE TOLERANCE		aaa		0.1	
MOLD FLATNESS		bbb		0.1	
COPLANARITY		ccc		0.08	
LEAD OFFSET		ddd		0.1	
EXPOSED PAD OFFSET		eee		0.1	

Figure 2-1 CV180ZB/CV1800B/CV1801B package dimensions, top view



		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	0.8	0.85	0.9
STAND OFF		A1	0	0.035	0.05
MOLD THICKNESS		A2	---	0.7	---
L/F THICKNESS		A3		0.152 REF	
LEAD WIDTH		b	0.13	0.18	0.23
BODY SIZE	X	D		7 BSC	
	Y	E		7 BSC	
LEAD PITCH		e		0.35 BSC	
EP SIZE	X	J	5.6	5.7	5.8
	Y	K	5.6	5.7	5.8
LEAD LENGTH		L	0.35	0.4	0.45
PACKAGE EDGE TOLERANCE		aaa		0.1	
MOLD FLATNESS		bbb		0.1	
COPLANARITY		ccc		0.08	
LEAD OFFSET		ddd		0.1	
EXPOSED PAD OFFSET		eee		0.1	

Figure 2-2. CV180ZB/CV1800B/CV1801B package dimensions, bottom view

2.1.2 Pin Distribution CV180ZB/CV1800B/CV1801B

		68	67	66	65	64	63	62	61	60	59	58	57	56	55	54	53	52
		PAD_AUD_AVREF	PAD_AUD_AINL_MIC	VSS18A_AUD	PAD_MIPIRX0P	PAD_MIPIRX0N	PAD_MIPIRX1P	PAD_MIPIRX1N	PAD_MIPIRX2P	PAD_MIPIRX2N	PAD_MIPIRX3P	PAD_MIPIRX3N	PAD_MIPIRX4P	PAD_MIPIRX4N	VDDC	USB_DM	USB_DP	VDD18A_USB_PLL_ETH_CSI
PAD_AUD_AOUTR	1																	51 VDD33A_ETH_USB_SD1
VDD18A_AUD	2																	50 PAD_ETH_RXM___EPHY_TXP
SD0_CLK	3																	49 PAD_ETH_RXP___EPHY_TXN
SD0_CMD	4																	48 PAD_ETH_TXM___EPHY_RXP
SD0_D0	5																	47 PAD_ETH_TXP___EPHY_RXN
VDDC	6																	46 VDDC
SD0_D1	7																	45 USB_VBUS_DET
SD0_D2	8																	44 ADC1
SD0_D3	9																	43 SD1_CLK
VDDIO_SD0_SPI	10																	42 SD1_CMD
SD0_CD	11																	41 SD1_D0
SD0_PWR_EN	12																	40 SD1_D1
VDDC	13																	39 SD1_D2
SPK_EN	14																	38 SD1_D3
UART0_TX	15																	37 VDDC
UART0_RX	16																	36 SD1_GPIO1
SPINOR_HOLD_X	17																	35 SD1_GPIO0
		18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
		SPINOR_SCK	SPINOR_MOSI	SPINOR_WP_X	SPINOR_MISO	SPINOR_CS_X	IIC0_SCL	IIC0_SDA	AUX0	VDDQ	VDDQ	VDDQ	VDDIO_RTC	PWR_VBAT_DET	PWR_SEQ2	PTEST	XTAL_XIN	XTAL_XOUT

Table 2-3 CV180ZB/CV1800B/CV1801B pin distribution

2.2 Pin information Description

Please refer to CV180xB_QFN68_PINOUT_CN.xlsx

2.3 Welding Process Suggestions

Please refer to Figure 2-4 for the lead-free reflow soldering process curve.
Please refer to Pure Sn

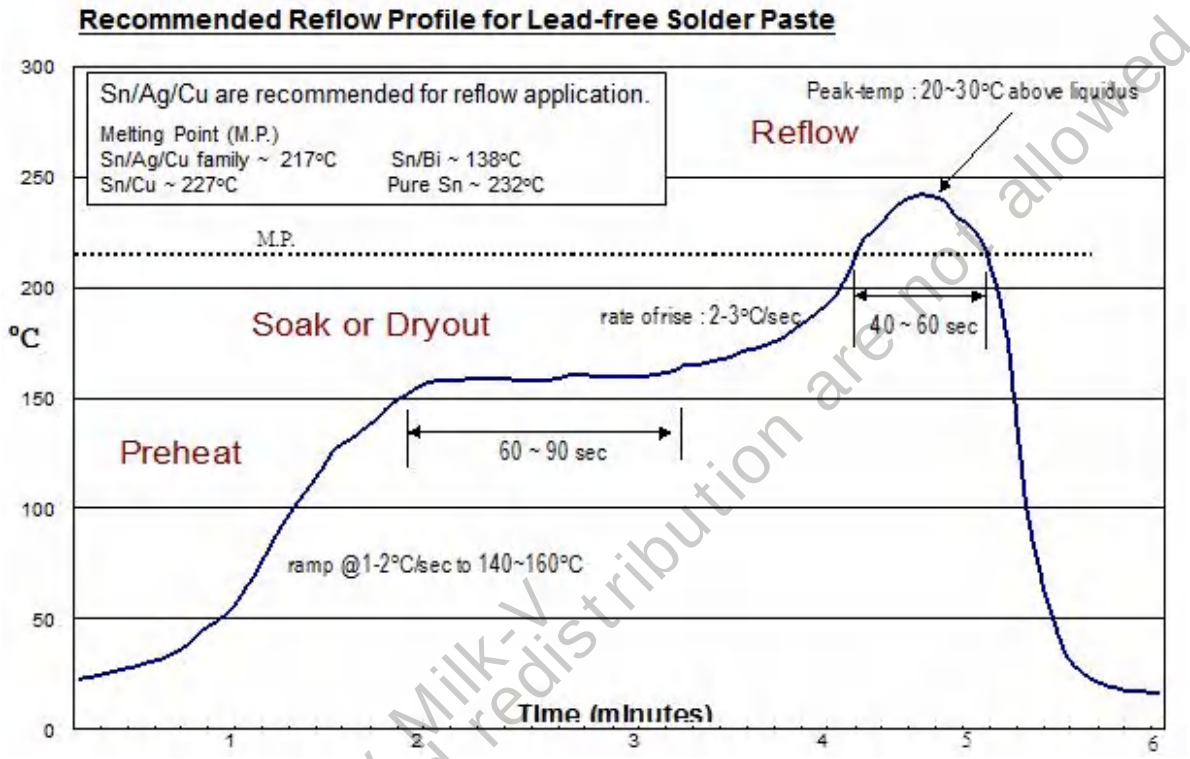


Figure 2-4 Lead-free Reflow Soldering process curve

Please refer to Table 2-1 for the parameters of lead-free reflow soldering process.

- The following parameters are only recommended values for reference. Clients need to make relative adjustments according to actual production conditions.

Table2-1 Lead-free reflow soldering process parameters

Area	Time	Heating Rate	Peak temperature	Cooling Rate
preheat zone (40~150°C)	60~120sc	1~2°C/sec		
soak zone(150~200°C)	60~90sec	< 1°C/sec		
reflow zone (>熔点 20~30°C)	40~60sec	2~3°C/sec	Sn/Ag/Cu 237~247°C Sn/Cu 247~257°C Pure Sn 252~262°C	

Area	Time	Heating Rate	Peak temperature	Cooling Rate
cooling zone (Tmax ~ Tamb)				1~4°C/sec

Due to environmental protection factors, the parameters for leaded reflow soldering are not currently provided.

2.4 Moisture Sensitivity Parameters

2.4.1 Moisture Barrier Packaging for CVITEK Products

This section establishes the principles for the storage and use of chips (moisture sensitive products) during welding. The relevant terms are as follows:

- Floor life: refers to the maximum allowable time between opening the moisture barrier packaging and reflow, in an environment with temperature < 30°C/60% RH.
- Shelf life: refers to the normal storage time after the moisture barrier packaging has been sealed.

2.4.1.1 Package Information

The moisture-proof vacuum package includes (1) chips and tray, (2) desiccant packs, and (3) humidity indicator cards (HIC).

Figure 2-5 Vacuum drying packaging information

Figure 2-6 Desiccant packs, humidity card, chip, and tray

2.4.1.2 Moisture-sensitive Product Incoming Inspection

After opening the vacuum moisture-proof bag before SMT, inspect the humidity card. There are many different styles of humidity cards, but if it shows that it has been exposed to moisture, it must be baked before SMT use. The relevant time and temperature parameters for baking are shown in Table 2-3.

If re-packaging after opening, and it has not been exposed for more than 2 hours in an environment of <30°C/60% RH, it can be vacuum dried and packaged by only replacing the drying bag. If it exceeds 2 hours, it is recommended to re-bake, replace the drying bag, and then reseal the package.

2.4.1.3 Storage and Use. (refer to JEDEC J-STD-033)

Shelf life:

The sealed vacuum moisture-proof bag can be stored for at least 12 months in an environment of 40°C/90% RH.

Floor life:

Before SMT, if the humidity card indicates that the components have not been exposed to moisture after opening in an environment of 30°C/60% RH, it can be used directly without baking. The time for Level 3 (the floor life classification of this chip is Level 3) is shown in Table 2-2.

Table 2-2 Humidity classification and floor life

Moisture classification level and floor life

Level	Floor Life (out of bag) at factory ambient ≤ 30 °C/60% RH or as stated
1	Unlimited at ≤ 30 °C/85%RH
2	1 year
2a	4 weeks
3	168 hours
4	72 hours
5	48 hours
5a	24 hours
6	Mandatory bake before use. After bake, must be reflowed within the time limit specified on the label

2.4.1.4 Rebaking

If found to have been exposed to moisture after opening, before SMT or before being resealed in vacuum packaging, they should be baked first. Baking temperature and time can be referred to in Table 2-3.

After baking, shelf life can be recalculated after being sealed in moisture-proof packaging.

If not sealed in moisture-proof packaging after baking, the storage time should refer to the floor life.

Table 2-3 Baking temperature and time table

Package Thickness	Level	Bake @ 125 °C	Bake @ 40 °C ≤ 5% RH
≤ 1.4 mm	2a	4 h.	5 days
	3	7 h.	11 days
	4	9 h.	13 days
	5	10 h.	14 days
	5a	14 h.	19 days
≤ 2.0 mm	2a	18 h.	21 days
	3	24 h.	33 days
	4	31 h.	43 days
	5	37 h.	52 days
	5a	48 h.	68 days
≤ 4.0 mm	2a	48 h.	67 days
	3	48 h.	67 days
	4	48 h.	68 days
	5	48 h.	68 days
	5a	48 h.	68 days

2.5 Electrical Performance Parameters

2.5.1 Power Consumption Parameters

Typical scenario: 1080P + Video Encode + AI ~500mW
Other scenarios: TBD

2.5.2 Temperature and Thermal Resistance Parameters (CV180ZB/CV1800B/CV1801B)

Thermal resistance values ThetaJA, JB, JC of the chip. The results of the actual test conducted on the JEDEC 2s2p PCB are shown in Table 2 and 4.

Table 2-4 Thermal resistance parameters for CV180ZB/CV1800B/CV1801B

PCB Condition	Package Size(mm)	Theta JA (C/W)			Psi Jt (C/W)	Theta JC (C/W)	Theta JB (C/W)
		0 m/s	1 m/s	2 m/s			
JEDEC 2s2p PCB	7x7	24.9	19.5	18.3	0.2	9.60	6.61

Temperature-related parameters of the chip are shown in Table 2-5.

Table 2-5 Temperature-related parameters

	The minimum	The maximum	Note
working environment temperature Tamb	-30°C	70°C	1
the recommended value for junction temperature (Tjunc) of the chip	-30°C	85°C ~ 105°C	2
the destructive junction temperature	-40°C	+125°C	3, 4

1. The maximum operating temperature in the working environment depends on the

power consumption and heat dissipation conditions of the scenario, without violating the premise of junction temperature.

2. The recommended range of junction temperature is mainly considered to avoid thermal runaway caused by poor heat dissipation conditions at high temperatures, which may lead to uncontrolled temperature entering the destructive junction temperature range and damaging the chip. In addition, long-term operation at high temperatures may slightly accelerate chip aging and reduce its service life.
3. The DRAM used guarantees a junction temperature of only -40°C to 115°C. Content inside the DRAM cannot be guaranteed to remain intact beyond this range.
4. When the chip operates at the destructive junction temperature, it may cause irreversible physical damage to the chip.

2.5.3 Destructive Voltage

Destructive voltage parameters are shown in Table 2-6. Working above the destructive voltage may cause irreversible physical damage.

Table 2-6 Destructive voltage parameters (CV180ZB/CV1800B/CV1801B)

Parameter		Max	Unit
VDDC	Core power	1.05V	V
VDD18A_AUD	Analog power for Audio ADC/DAC	1.98	V
VDD18A_USB_PLL_ETH_CSI	Analog power for USB, PLL, ETH, efuse , MIPI	1.98	V
VDD33A_ETH_USB_SD1	Analog power for Ethernet PHY, USB PHY, IO power for SD1 domain	3.465	V
VDDIO_SD0_SPI	IO power for SPI & SD0 domain	3.465	V
VDDIO_RTC	IO power for RTC domain (backup power)	1.98	V
VDDQ	IO & DRAM Power for DDR2/DDR3L/DDR3	1.98	V

2.5.4 Power Sequencing(CV180ZB/CV1800B/CV1801B)

In principle, the chip can be divided into the following groups. The power domains within the same group are powered on/off simultaneously. For different groups, the power on/off time is separated according to the following conditions.

Core power domain

VDDC

1.8V IO domain

VDD18A_AUD (analog)

VDD18A_USB_PLL_ETH_CSI (analog)

18OD33 IO domain (1.8V domain or 3V domain depending on the voltage)

VDDIO_SD0_SPI (if SD0 need to be connected with SD Card, this must be 3V)

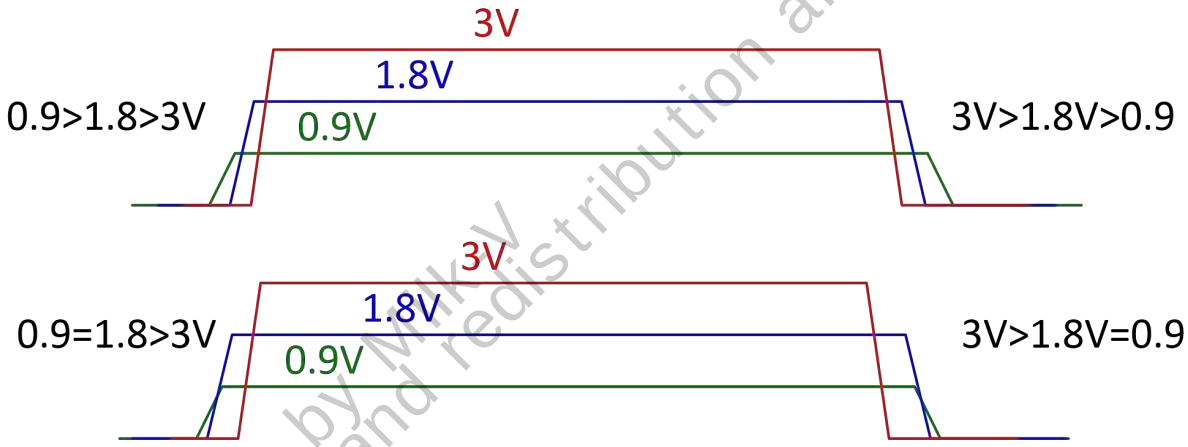
3V domain

VDD33A_ETH_USB_SD1

DR IO & DRAM domain

VDDQ

In principle, 0.9V power domain and 1.8V power domain can be powered on at the same time, or power-on the 0.9V power domain prior to 1.8V power domain. However, the 3V power domain must be powered on after the establishment of 1.8V power domain. Violations may cause irreversible damage to the chip. The power-off sequence is reverse of the power-on sequence.



Possible risky power-up and power-down behaviors include:

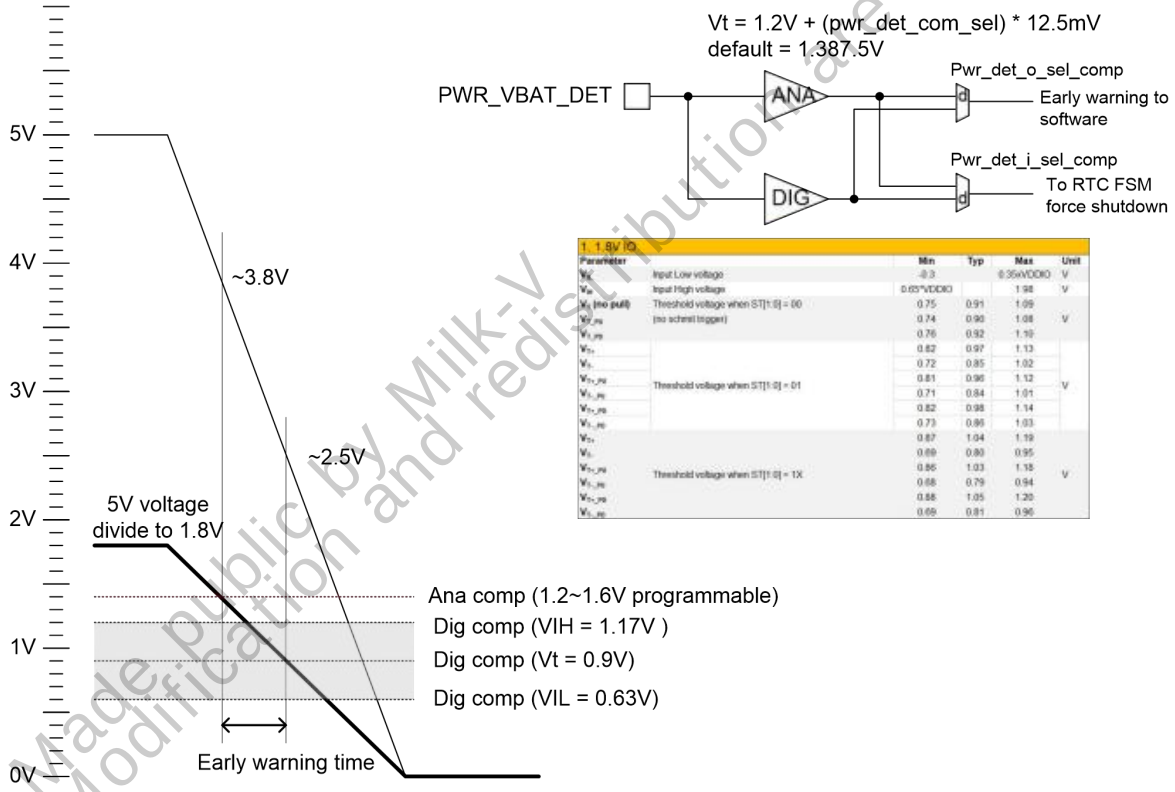
1. During power-up, if VDD3 > 2V while VDD18 has not reached 1.8V-10%, it may damage the 3V circuit.
2. During power-down, if VDD3 is lower than 2V while VDD18 has already dropped below 1.8V-10%.
3. During power-up, if VDD18 > 0.7V while VDD09 is still below 0.5V, it may cause efuse misoperation.
4. During power-down, if VDD09 is below 0.5V while VDD18 is still higher than 0.7V, it may also cause efuse misoperation.

The chip provides two pins, PWR_SEQ1 and PWR_SEQ2, to co-control the power supply (VDDIO_RTC domain) switch. SEQ1 is preset to 0.9V and 1.8V, while SEQ2 controls 3V.

Some plug-in systems may use RC to determine the switch for both 0.9V and 1.8V, but it is important that the chip's 3V still needs to be controlled by SEQ2 to prevent damage.

During power-up, SEQ1 —>SEQ2
During power-down, SEQ2 —>SEQ1.

PWR_VBAT_DET is used to detect the status of the main power supply. If the voltage is low, the software will receive an interrupt first (such as stopping writing flash to prevent damage to the file system). If the voltage continues to drop, the RTC module will actively start the power-down program. PWR_VBAT_DET also needs to be logic high to start up.



2.5.5 The DC/AC Electrical Parameters of the Power Supply

Table 2- 7 The power supply electrical parameters of CV180ZB/CV1800B/CV1801B (Recommended Operating Conditions)

Parameter		Min	Typ	Max	Unit
VDDC	Core power	0.81	0.9	0.99	V
VDD18A_AUD	Analog power for Audio ADC/DAC	1.62	1.8	1.98	V
VDD18A_USB_PLL_ETH_CSI	Analog power for Ethernet PHY, USB PHY, PLL, Efuse, MIPI	1.62	1.8	1.98	V
VDD33A_ETH_USB_SD1	Analog power for Ethernet PHY, USB PHY, IO power for SD1 domain	2.97	3.3	3.465	V
					V
VDDIO_SD0_SPI	IO power for SD0 & SPI domain	1.71 2.85	1.8 3.0/3.3	1.89 3.15/3.465	V
VDDIO_RTC	IO power for RTC domain IO & LDO	1.3V	1.8	+10%	V
VDDQ	IO & DRAM Power for DDR3L IO & DRAM Power for DDR3 IO & DRAM Power for DDR2	1.283 1.425 1.710	1.35 1.50 1.80	1.417 1.575 1.890	V
Tjunc	Junction Temperature (Max reduce from 125C due to DRAM)	-40	25	115 (note)	℃

Note: The operating junction temperature of the DRAM used is guaranteed to be only between -40°C to 115°C. Contents inside the DRAM cannot be guaranteed to be intact beyond this temperature range.

2.5.6 1.8V I/O Electrical Parameters

For domain (VDD18A_USB_PLL_ETH_CSI, VDDIO_RTC)

Table 2- 8 1.8V I/O Electrical Parameters

Parameter		Min	Typ	Max	Unit
V_{IL}	Input Low voltage	-0.3		0.35xVDDIO	V
V_{IH}	Input High voltage	0.65*VDDIO		1.98	V
V_T (no pull) V_{T_PU} V_{T_PD}	Threshold voltage when ST[1:0] = 00 (no schmit trigger)	0.75 0.74 0.76	0.91 0.90 0.92	1.09 1.08 1.10	V
V_{T+} V_{T-} V_{T+_PU} V_{T-_PU} V_{T+_PD} V_{T-_PD}	Threshold voltage when ST[1:0] = 01	0.82 0.72 0.81 0.71 0.82 0.73	0.97 0.85 0.96 0.84 0.98 0.86	1.13 1.02 1.12 1.01 1.14 1.03	V
V_{T+} V_{T-} V_{T+_PU} V_{T-_PU} V_{T+_PD} V_{T-_PD}	Threshold voltage when ST[1:0] = 1X	0.87 0.69 0.86 0.68 0.88 0.69	1.04 0.80 1.03 0.79 1.05 0.81	1.19 0.95 1.18 0.94 1.20 0.96	V
I_I	Input leakage (V_I = 1.8V or 0V)			+/-10u	A
I_{OZ}	Tri-state output leakage current (V_O=1.8V or 0V)			+/-10u	A
R_{PU}	Pull up resistor	55k	79k	121k	Ω
R_{PD}	Pull down resistor	51k	87k	169k	Ω
V_{OL}	Output low voltage			0.45	V
V_{OH}	Output high voltage	1.35			V
I_{OL}	Low level output current @ V_{OL} (max) DS[1:0] = 00 DS[1:0] = 01 DS[1:0] = 10 DS[1:0] = 11	 7.6 15.2 22.6 29.7	 12.8 25.3 37.4 49	 18.0 35.5 52.2 67.9	 mA mA mA mA
I_{OH}	High level output current @ V_{OH} (max) DS[1:0] = 00 DS[1:0] = 01 DS[1:0] = 10 DS[1:0] = 11	 4.8 9.5 14.3 18.9	 10.8 21.5 32.1 42.4	 18.9 37.4 55.9 73.9	 mA mA mA mA

2.5.7 18OD33 IO (VDDIO=1.8V) Electrical Parameters

For domain (VDDIO_SD0_SPI, VDDIO_SD1)

Table 2- 9 18OD33 IO (VDDIO=1.8V) Electrical Parameters

Parameter		Min	Typ	Max	Unit
V_{IL}	Input Low voltage	-0.3		0.58	V
V_{IH}	Input High voltage	1.27		2.00	V
V_T (no pull)	Threshold voltage when ST = 0	0.91	0.97	1.03	V
V_{T_PU}	(no schmit trigger)	0.90	0.96	1.02	
V_{T_PD}		0.91	0.97	1.06	
V_{T+} (no pull)	Threshold voltage when ST = 1	1.03	1.07	1.12	V
V_{T-} (no pull)		0.75	0.83	0.91	
V_{T+_PU}		1.02	1.06	1.11	
V_{T-_PU}		0.74	0.82	0.90	
V_{T+_PD}		1.03	1.08	1.13	
V_{T-_PD}		0.75	0.83	0.92	
I_I	Input leakage (V_I = 1.8V or 0V)			+/-10u	A
I_{OZ}	Tri-state output leakage current (V_O=1.8V or 0V)			+/-10u	A
R_{PU}	Pull up resistor	33k	60k	92k	Ω
R_{PD}	Pull down resistor	34k	61k	158k	Ω
V_{OL}	Output low voltage			0.45	V
V_{OH}	Output high voltage	1.40			V
I_{OL}	Low level output current @ V_{OL} (max)				
	DS[2:0] = 000	4.9	7.8	11.1	mA
	DS[2:0] = 001	7.4	11.7	16.4	mA
	DS[2:0] = 010	9.8	15.5	21.7	mA
	DS[2:0] = 011	12.2	19.2	26.7	mA
	DS[2:0] = 100	14.6	23.0	31.9	mA
	DS[2:0] = 101	17.0	26.6	36.8	mA
	DS[2:0] = 110	19.4	30.2	41.6	mA
	DS[2:0] = 111	21.7	33.7	46.2	mA
I_{OH}	High level output current @ V_{OH} (max)				
	DS[2:0] = 000	3.6	6.2	9.5	mA
	DS[2:0] = 001	5.4	9.3	14.3	mA
	DS[2:0] = 010	7.2	12.4	19.1	mA
	DS[2:0] = 011	9.0	15.4	23.8	mA
	DS[2:0] = 100	10.8	18.5	28.5	mA
	DS[2:0] = 101	12.6	21.6	33.1	mA
	DS[2:0] = 110	14.4	24.6	37.8	mA
	DS[2:0] = 111	16.2	27.7	42.5	mA

2.5.8 18OD33 IO (VDDIO=3.0V) Electrical Parameters

For domain (VDDIO_SD0_SPI, VDDIO_SD1)

Table 2- 10 18OD33 IO (VDDIO=3.0V) Electrical Parameters

Parameter		Min	Typ	Max	Unit
V_{IL}	Input Low voltage	-0.3		0.25*VDDIO	V
V_{IH}	Input High voltage	0.625*VDDIO		3.3	V
V_T (no pull)	Threshold voltage when ST = 0	0.82	0.95	1.11	V
V_{T_PU}	(no schmit trigger)	0.81	0.93	1.09	
V_{T_PD}		0.83	0.96	1.13	
V_{T+} (no pull)	Threshold voltage when ST = 1	1.00	1.10	1.23	V
V_{T-} (no pull)		0.75	0.90	1.08	
V_{T+_PU}		1.00	1.09	1.21	
V_{T-_PU}		0.73	0.88	1.05	
V_{T+_PD}		1.01	1.11	1.25	
V_{T-_PD}		0.75	0.91	1.09	
I_I	Input leakage (V_I = 3.0V or 0V)			+/-10u	A
I_{OZ}	Tri-state output leakage current (V_O=3.0V or 0V)			+/-10u	A
R_{PU}	Pull up resistor	33k	60k	93k	Ω
R_{PD}	Pull down resistor	34k	62k	285k	Ω
V_{OL}	Output low voltage			0.125*VDDIO	V
V_{OH}	Output high voltage	0.75*VDDIO			V
I_{OL}	Low level output current @ V_{OL} (max)				
	DS[2:0] = 000	3.1	5.5	8.6	mA
	DS[2:0] = 001	4.7	8.2	12.7	mA
	DS[2:0] = 010	6.2	10.8	16.9	mA
	DS[2:0] = 011	7.7	13.4	20.8	mA
	DS[2:0] = 100	9.3	16.1	24.9	mA
	DS[2:0] = 101	10.8	18.7	28.8	mA
	DS[2:0] = 110	12.3	21.2	32.6	mA
	DS[2:0] = 111	13.8	23.7	36.3	mA
I_{OH}	High level output current @ V_{OH} (max)				
	DS[2:0] = 000	5.0	7.5	10.5	mA
	DS[2:0] = 001	7.5	11.2	15.7	mA
	DS[2:0] = 010	10.1	14.9	21.0	mA
	DS[2:0] = 011	12.6	18.6	26.2	mA
	DS[2:0] = 100	15.1	22.3	31.4	mA
	DS[2:0] = 101	17.6	26.0	36.5	mA
	DS[2:0] = 110	20.1	29.8	41.8	mA
	DS[2:0] = 111	22.6	33.4	46.9	mA

2.5.9 Audio GPIO Electrical Parameters

Table 2- 11 Audio GPIO Electrical Parameters

Parameter		Min	Typ	Max	Unit
V_{IL}	Input Low voltage	-0.3		0.55	V
V_{IH}	Input High voltage	1.2		1.98	V
V_{T+} V_{T-}	Threshold voltage with schmitt trigger	0.8 0.65	0.95 0.82	1.1 0.99	V
I_I	Input leakage (V_I = 1.8V or 0V)			+/-4u	A
I_{OZ}	Tri-state output leakage current (V_O=1.8V or 0V)			+/-4u	A
V_{OL}	Output low voltage			0.4	V
V_{OH}	Output high voltage	1.4			V
I_{OL}	Low level output current @ V_{OL} (max)	4.9	9.9	18.4	mA
I_{OH}	High level output current @ V_{OH} (max)	11.3	17.1	26.1	mA

2.5.10 ETH GPIO Electrical Parameters

Table 2- 12 ETH GPIO Electrical Parameters

Parameter		Min	Typ	Max	Unit
V_{IL}	Input Low voltage	-0.3		0.3*VDD18A	V
V_{IH}	Input High voltage	0.7*VDD18A		1.98	V
V_{T+} V_{T-}	Threshold voltage with schmitt trigger	0.84 0.66	0.99 0.83	1.14 1.01	V
I_I	Input leakage (V_I = 1.8V or 0V)			+/-1.3u	A
I_{OZ}	Tri-state output leakage current (V_O=1.8V or 0V)			+/-1.3u	A
V_{OL}	Output low voltage			0.4	V
V_{OH}	Output high voltage	VDD18A-0.4			V
I_{OL}	Low level output current @ V_{OL} (max) DS=0 Low level output current @ V_{OL} (max) DS=1	8.8 10.2	15.7 17.8	27.3 30.5	mA
I_{OH}	High level output current @ V_{OH} (max) DS=0 High level output current @ V_{OH} (max) DS=1	4.0 4.7	5.3 6.2	7.4 8.5	mA

2.5.11 MIPI Rx Electrical Parameters

The MIPI D-PHY High Speed(MIHS) Electrical Parameters are listed in Table 2- 12 and Table 2- 13.

The MIPI D-PHY Low Power(MILP) Electrical Parameters are listed in Table 2- 14 and Table 2- 15.

Table 2- 12. MIPI D-PHY High Speed(MISH) DifferentialDC Electrical Parameters

Parameter	Symbol	Data Rate	Min	Typ	Max	Unit
Common Mode Voltage Range (VP+VM)/2	VCM(MIHS)	≤1.5Gbps	70	200	330	mV
		>1.5Gbps				
Differential input impedance	ZID(MIHS)	≤1.5Gbps	80	100	125	ohm
		>1.5Gbps				
Single-ended threshold for HS termination enable	VTERM-EN(MIHS)	≤1.5Gbps	--	--	450	mV
		>1.5Gbps				

Table 2- 13. MIPI D-PHY High Speed(MIHS) Differential AC Electrical Parameters

Parameter	Symbol	Data Rate	Min	Typ	Max	Unit
Differential Input Threshold Voltage (VP—VM)	VIDTH(MIHS)	≤1.5Gbps	-70	--	70	mV
		>1.5Gbps	-40	--	40	
Single-ended Input Voltage VP,VM	VIS(MIHS)	≤1.5Gbps	-40	--	460	mV
		>1.5Gbps				
Common-mode interface beyond 450MHz	ΔVCMRX	≤1.5Gbps	--	--	100	mV
		>1.5Gbps				
Common-mode interface 50MHz-450MHz	ΔVCMRX(LF)	≤1.5Gbps	-50	--	50	mV
		>1.5Gbps	-25	--	25	
Single-ended threshold for HS termination enable	VTERM-EN(MIHS)	≤1.5Gbps	--	--	450	mV
		>1.5Gbps				
Common-mode termination	CCM	≤1.5Gbps	--	--	60	pF
		>1.5Gbps				

Table 2- 14 MIPI D-PHY Low Power(MILP) Differential DC Electrical Parameters

Parameter	Symbol	Min	Typ	Max	Unit
Logic 1 input voltage	VIHLP	740	--	--	mV
Logic 0 input voltage	VILLP	--	--	550	mV
Input hysteresis	VHYST	25	--	--	mV

Table 2- 15 MIPI D-PHY Low Power(MILP) Differential AC Electrical Parameters

Parameter	Symbol	Min	Typ	Max	Unit
Input pulse rejection	eSPIKE	--	--	300	V·ps
Pulse width response	TMIN-RX	20	--	--	ns
Peak interference amplitude	VINT	--	--	200	mV
Interference frequency	fINT	450	--	--	MHz

2.5.12 Sub-LVDS Electrical Parameters

The electrical parameters are listed Table 2- 16 and Table 2- 17.

Table 2- 16 Sub-LVDS(SL) Differential DC Electrical Parameters

Parameter	Symbol	Min	Typ	Max	Unit
Common Mode Voltage Range (VP+VM)/2	VCM(SL)	600	900	1200	mV
Differential input impedance	ZID(SL)	80	100	120	mV

Table 2- 17 Sub-LVDS(SL) Differential AC Electrical Parameters

Parameter	Symbol	Min	Typ	Max	Unit
Differential Input Threshold Voltage (VP—VM)	VIDTH(SL)	-70	--	70	mV
Single-ended Input Voltage VP,VM	VIS(SL)	400	--	1400	mV

2.5.13 HiSPi Electrical Parameters

HiSPi is divided into SLVS (HSSL) and HiVCM(HSHI).The electrical parameters are listed in Table 2- 18 and Table 2- .

Table 2- 18 HiSPi Differential DC Electrical Parameters

Parameter	Symbol	Min	Typ	Max	Unit
Common Mode Voltage Range (VP+VM)/2	VCM(HSSL)	50	200	350	mV
	VCM(HSHI)	660	900	1170	
Differential input impedance	ZID(HSSL)	80	100	125	mV
	ZID(HSHI)	80	100	125	

Table 2- 20 HiSPi Differential AC Electrical Parameters

Parameter	Symbol	Min	Typ	Max	Unit
Differential Input Threshold Voltage (VP—VM)	VIDTH(HSSL)	-70	--	70	mV
	VIDTH(HSHI)	-100	--	100	
Single-ended Input Voltage VP,VM	VIS(HSSL)	-40	--	490	mV
	VIS(HSHI)	550	--	1350	

2.5.14 SDIO Electrical Parameters

Please refer to 2.5.7 and 2.5.8 for SD0/SD1.

2.5.15 2.5.15 VI RAW/BT.656/BT.1120Electrical Parameters(CV180ZB/CV1800B/CV1801B)

Please refer to 2.5.7 and 2.5.8 according to the domain of IO.

2.5.16 AUDIO CODEC Electrical Parameters

Table 2- 19 Audio CODEC Overall Index

Parameter	Min	Typ	Max	Unit	Description
Analog Power AVDD	1.62	1.8	1.98	V	
VREF		1.4/1.8 *VDD		V	

Table 2- 20 Audio DAC Electrical Parameters

Parameter	Min	Typ	Max	Unit	Description
Full Output Amplitude		1.55		Vpp	Maximum output signal swing

Table 2- 21 Audio ADC Electrical Parameters

Parameter	Min	Typ	Max	Unit	Description
Maximum Input Amplitude		1.75		Vpp	Maximum output signal swing

2.6 Timing

2.6.1 SPI NOR Timing

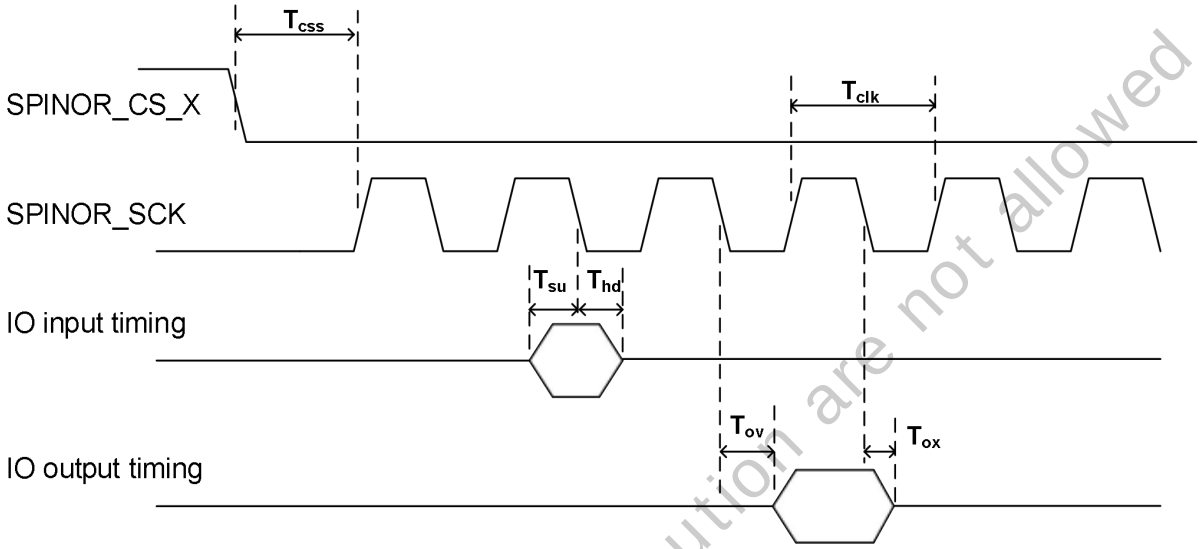


Figure 2- 7 SPI NOR Timing Diagram

* IO input timing / IO output timing is IO timing of SPINOR_SDI、SPINOR_SDO、SPINOR_HOLD_X、SPINOR_WP_X in 1 x I/O, 2 x I/O and 4x I/O modes.

Table 2- 24 SPI_NOR Timing Requirements

Symbol	Description	Min	Typ	Max	Unit
T_{css}	Time of CS negative edge to the first clock edge	13.4	-	-	ns
T_{clk}	Clock Cycle	13.4	-	-	ns
T_{su}	Input Signal Setup Time	3.5	-	-	ns
T_{hd}	Input Signal Hold Time	0	-	-	ns
T_{ov}	Output Signal Delay	-	-	2.6	ns
T_{OX}	Output Signal Hold Time	-1.5	-	-	ns

2.6.2 SPI NAND Timing

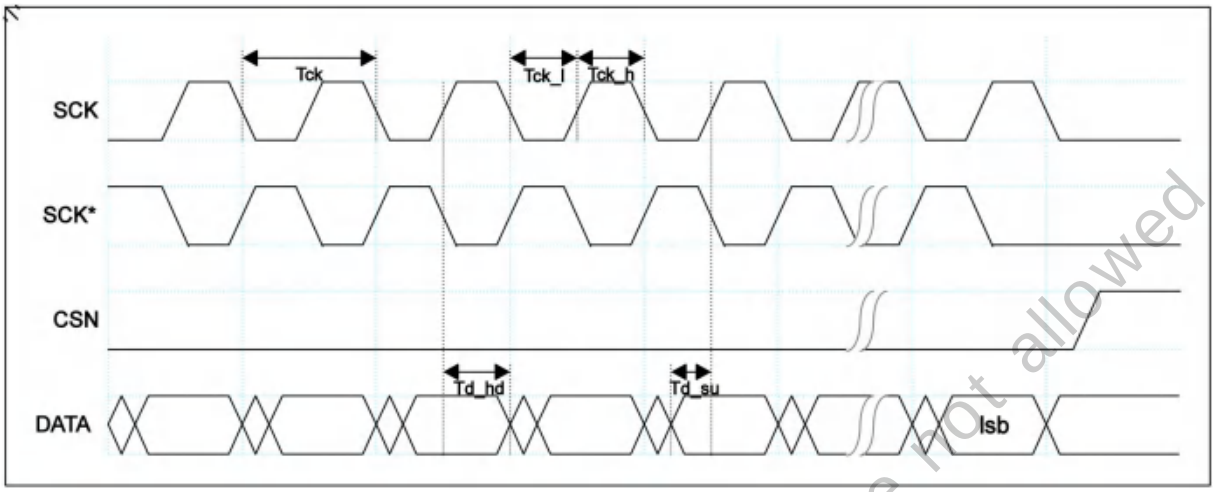


Figure 2- 8 SPI NAND Input Timing Diagram

Table 2- 22 SPI NAND Input Timing Requirements

Parameter	Symbol	Min	Typ	Max	Unit
Clock Cycle	Tck	10.66		170.56	ns
Data Input Setup Time	Td_su	2.00			ns
Data Input Hold Time	Td_hd	1.20			ns

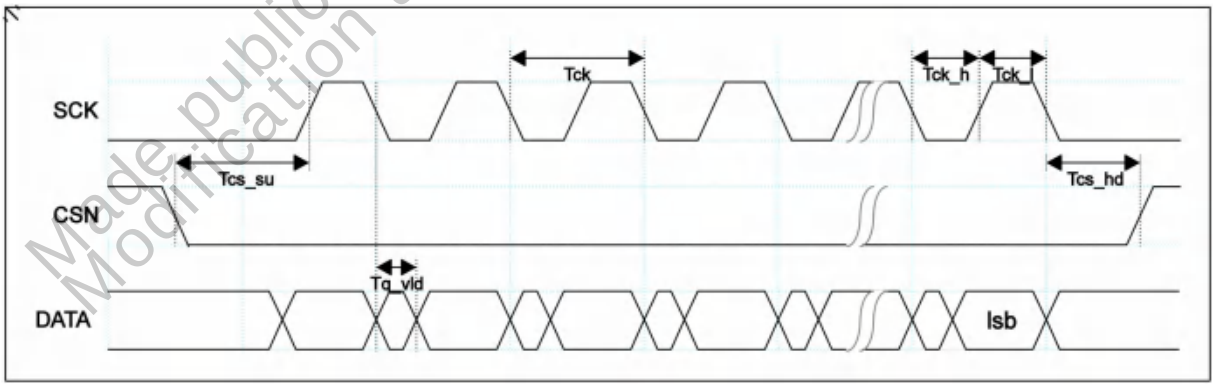


Figure 2- 9 SPI NAND Output Timing Diagram

Table 2- 23 SPI NAND Output Timing Requirements

Parameter	Symbol	Min	Typ	Max	Unit
Clock Cycle	Tck	10.66		170.56	ns
Clock High Period	Tck_h	5.33		85.28	ns
Clock Low Period	Tck_l	5.33		85.28	ns
Output CS Setup Time	Tcs_su	10.66			ns
Output CS Hold Time	Tcs_hd	10.66			ns
Clock LOW to Output Valid	Tq_vld	-1.00		2.00	ns

2.6.3 VI Timing

The timing of VI is shown as the figure below.

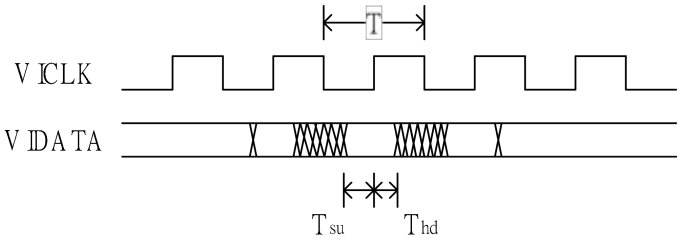


Figure 2-10 VI Timing Diagram

Wherein, the VI timing parameters are listed in the table below.

Table 2-27 VI Timing Requirements

	Symbol	Min	Typ	Max	Unit
VICLK clock cycle	T	6.73			ns
VIDATA setup time	Tsu	1.9			ns
VIDATA hold time	Thd	0.8			ns

2.6.4 AIAO (I2S/PCM) Timing

The RX timing diagram of I2S and PCM modes for connecting with external Audio Codec is shown as below figure.

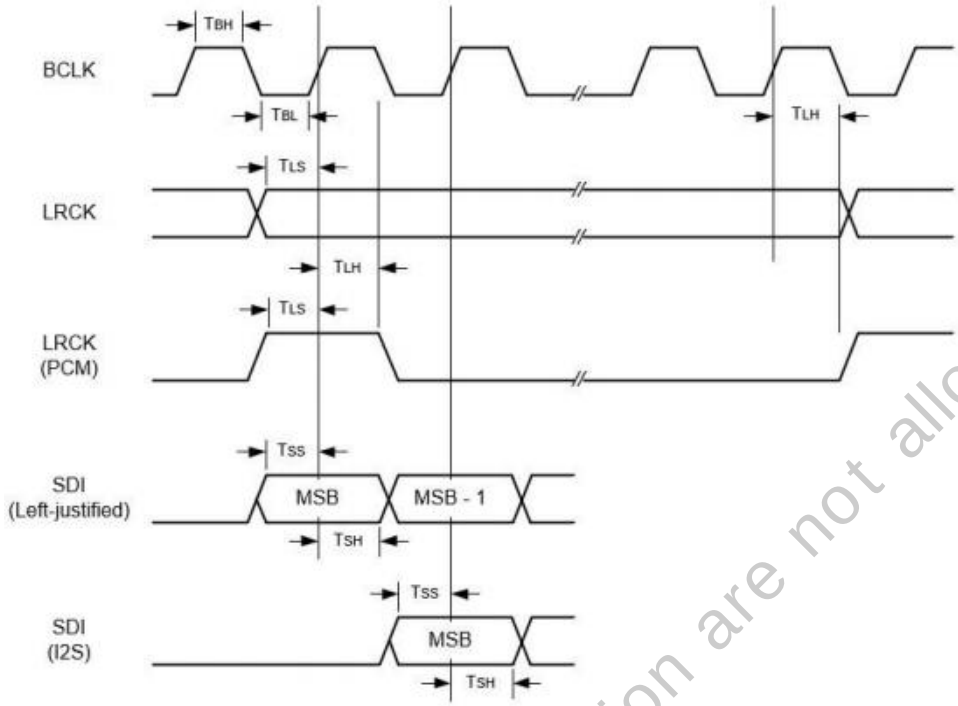


Figure 2-11 I2S & PCM Rx Timing Diagram

The TX timing diagram of I2S and PCM modes is shown as below figure.

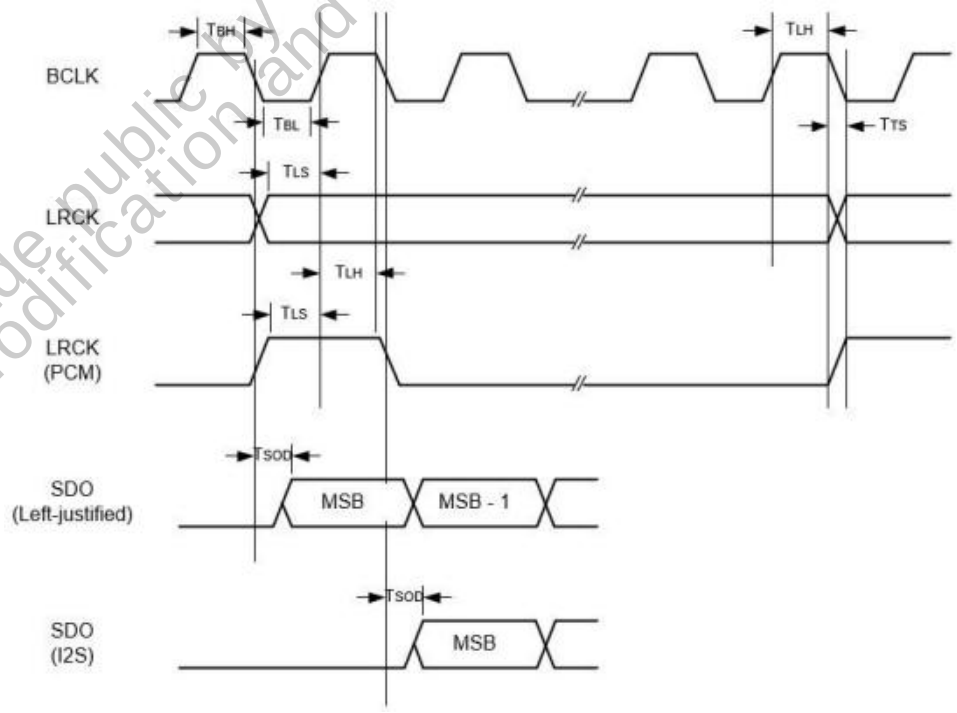


Figure 2-12 I2S & PCM Tx Timing Diagram

The digital timing specifications are listed as below table.

Table 2-28 I2S / PCM Timing Requirements

Symbol	Parameter	Min	Typ	Max	Unit
T_{BL}	BCLK low pulse width (master and slave modes)	40	-	-	ns
T_{BH}	BCLK high pulse width (master and slave modes)	40	-	-	ns
T_{LS}	LRCK setup time to BCLK rising (slave mode)	10	-	-	ns
T_{LH}	LRCK hold time from BCLK rising (slave mode)	10	-	-	ns
T_{SS}	SDI setup time to BCLK rising (master and slave modes)	10	-	-	ns
T_{SH}	SDI hold time from BCLK rising (master and slave modes)	10	-	-	ns
T_{TS}	BCLK falling to LRCK timing skew (master mode)	0	-	10	ns
T_{SOD}	SDO delay time from BCLK falling (master and slave modes)	0	-	10	ns

2.6.5 I2C Timing

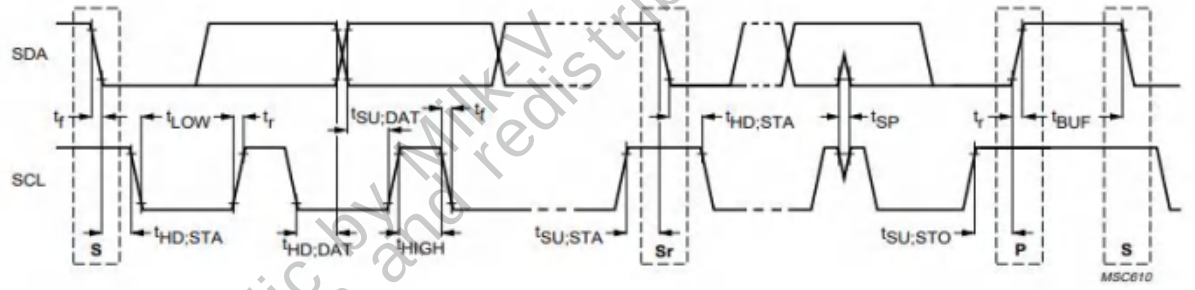


Figure 2- 7 I2C Timing Diagram

Table 2- 29 I2C Timing Requirements

PARAMETER	SYMBOL	STANDARD-MODE		FAST-MODE		UNIT
		MIN.	MAX.	MIN.	MAX.	
SCL clock frequency	f_{SCL}	0	100	0	400	kHz
Hold time (repeated) START condition. After this period, the first clock pulse is generated	$t_{HD;STA}$	4.0	–	0.6	–	µs
LOW period of the SCL clock	t_{LOW}	4.7	–	1.3	–	µs
HIGH period of the SCL clock	t_{HIGH}	4.0	–	0.6	–	µs
Set-up time for a repeated START condition	$t_{SU;STA}$	4.7	–	0.6	–	µs
Data hold time: for CBUS compatible masters (see NOTE, Section 10.1.3) for I^2C-bus devices	$t_{HD;DAT}$	 5.0 0(2)	 – 3.45(3)	 – 0(2)	 – 0.9(3)	 µs µs
Data set-up time	$t_{SU;DAT}$	250	–	100(4)	–	ns
Rise time of both SDA and SCL signals	t_r	–	1000	$20 + 0.1C_b$(6)	300	ns
Fall time of both SDA and SCL signals	t_f	–	300	$20 + 0.1C_b$(5)	300	ns
Set-up time for STOP condition	$t_{SU;STO}$	4.0	–	0.6	–	µs
Bus free time between a STOP and START condition	t_{BUF}	4.7	–	1.3	–	µs
Capacitive load for each bus line	C_b	–	400	–	400	pF
Noise margin at the LOW level for each connected device (including hysteresis)	V_{nL}	$0.1V_{DD}$	–	$0.1V_{DD}$	–	V
Noise margin at the HIGH level for each connected device (including hysteresis)	V_{nH}	$0.2V_{DD}$	–	$0.2V_{DD}$	–	V

2.6.6 SPI Timing

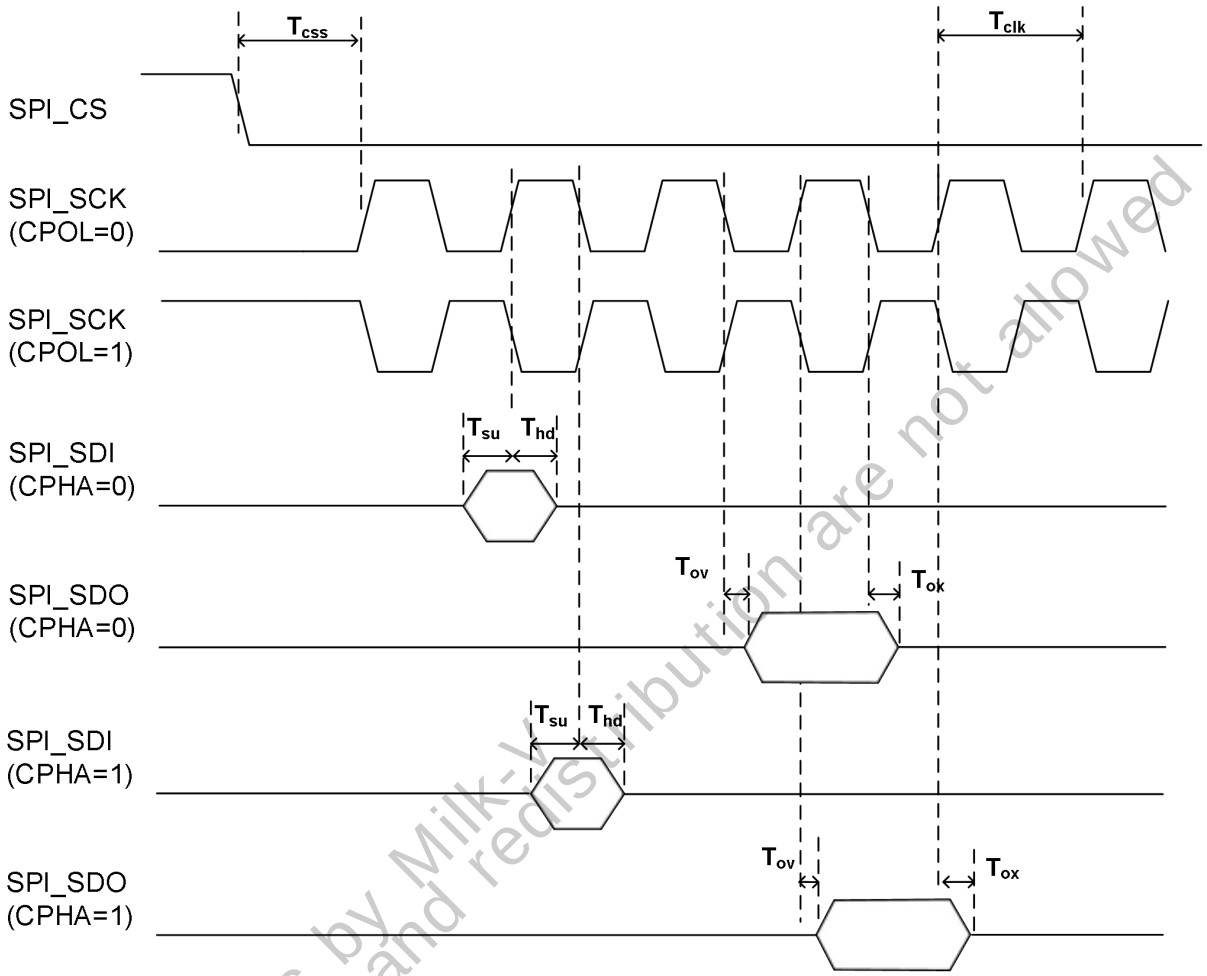


Figure 2- 8 SPI Timing Diagram

Table 2- 24 SPI Timing Requirements

Symbol	Description	Min	Typ	Max	Unit
F_{clk}	SCK frequency		-	46.8	MHz
T_{css}	Time of CS negative edge to the first clock edge	21.4	-	-	ns
T_{clk}	Clock cycle	21.4	-	-	ns
T_{su}	Input setup time	9.5	-	-	ns
T_{hd}	Input hold time	0	-	-	ns
T_{ov}	Input delay	-	-	3	ns
T_{OX}	Output hold time	-3	-	-	ns

2.6.7 MIPI Rx Timing

The speed range of MIPI Rx is : 0.08Gbps≤Data Rate≤1.5Gbps

A. 0.08Gbps≤Data Rate≤1.5Gbps

The timing is shown as Figure 2-15, and the timing requirements are listed in Table 2-31.

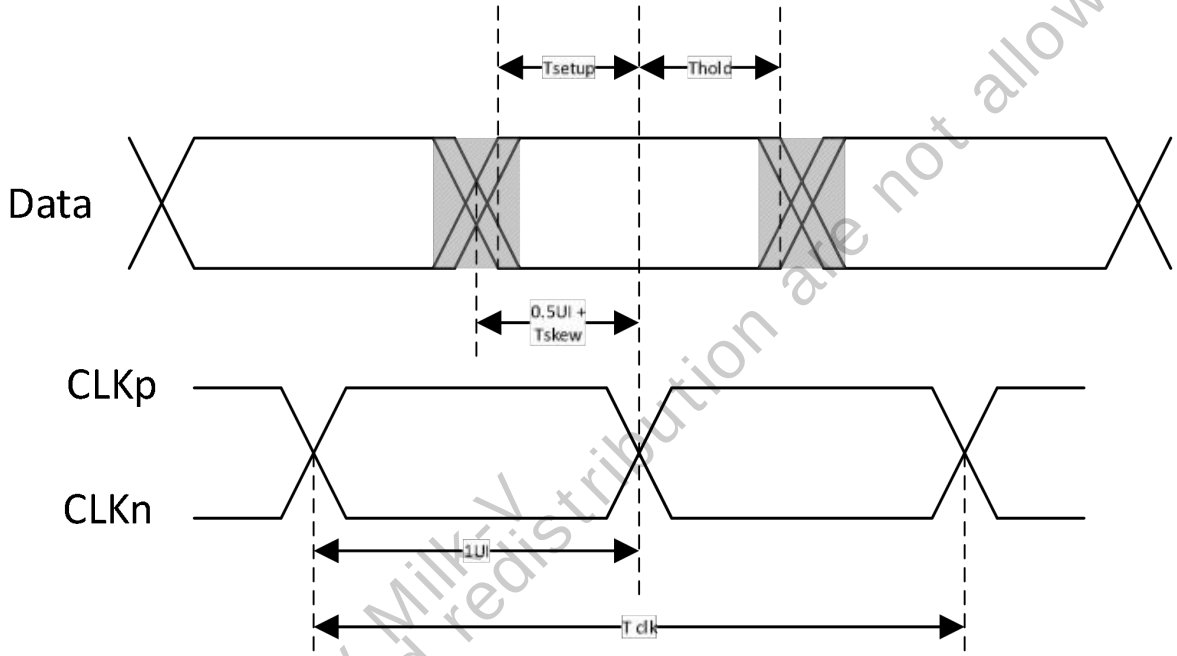


Figure 2- 15 Timing Diagram of MIPI Rx Clock when 0.08gbps ≤ Data Rate ≤ 1.5gbps

Table 2- 31 Timing parameters of MIPI Rx at 0.08Gbps ≤ Data Rate ≤ 1.5Gbps

Parameter	Symbol	Data Rate	Min	Typ	Max	Unit
Data Rate	Data Rate	0.08G≤Data Rate≤1G	0.08		1	Gbps
		1G<Data Rate≤1.5G	1		1.5	
Differential Clock Period	Tclk	0.08G≤Data Rate≤1G	1		12.5	ns
		1G<Data Rate≤1.5G	-0.2		0.2	
TX Data to Clock Skew	T_{SKEW}	0.08G≤Data Rate≤1G	-0.15		0.15	UIHS *
		1G<Data Rate≤1.5G	-0.2		0.2	
RX Data to Clock Setup Time Tolerance	T_{SETUP}	0.08G≤Data Rate≤1G	0.15			UIHS
		1G<Data Rate≤1.5G	0.2			
RX Data to Clock Hold Time Tolerance	T_{HOLD}	0.08G≤Data Rate≤1G	0.15			UIHS
		1G<Data Rate≤1.5G	0.2			

* UIHS= 1/(Data Rate) = Tclk/2

2.6.8 Sub-LVDS Timing

The Sub-LVDS clock data timing diagram is shown in the figure below, and the timing requirements are listed in Table 2-32.

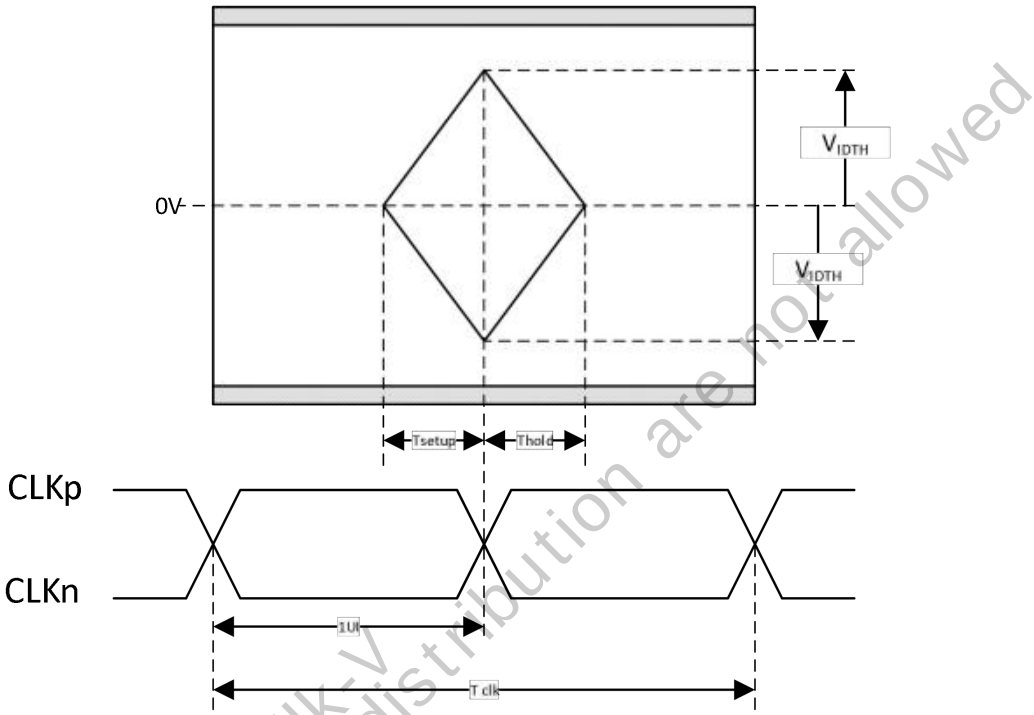


Figure 2- 16. Sub-LVDS Data Rate Timing Diagram

Table 2- 32. Sub-LVDS Timing Requirements

Parameter	Symbol	Min	Typ	Max	Unit
Data Rate	Data Rate	--	--	1.5	Gbps
Unit Interval	UI	666.6	--	--	ns
Differential Clock Period	Tclk	1333.3	--	--	ns
RX Data to Clock Setup Time Tolerance	T_{SETUP}	0.15	--	--	UI
RX Data to Clock Hold Time Tolerance	T_{HOLD}	0.15	--	--	UI
Differential Input Threshold Voltage (VP—VM)	VIDTH(SL)	-70	--	70	mV

* UI= 1/(Data Rate) = Tclk/2

2.6.9 HiSPi Timing

The HiSPi clock data timing diagram is shown in the figure below, and the timing requirements are listed in Table 2-33.

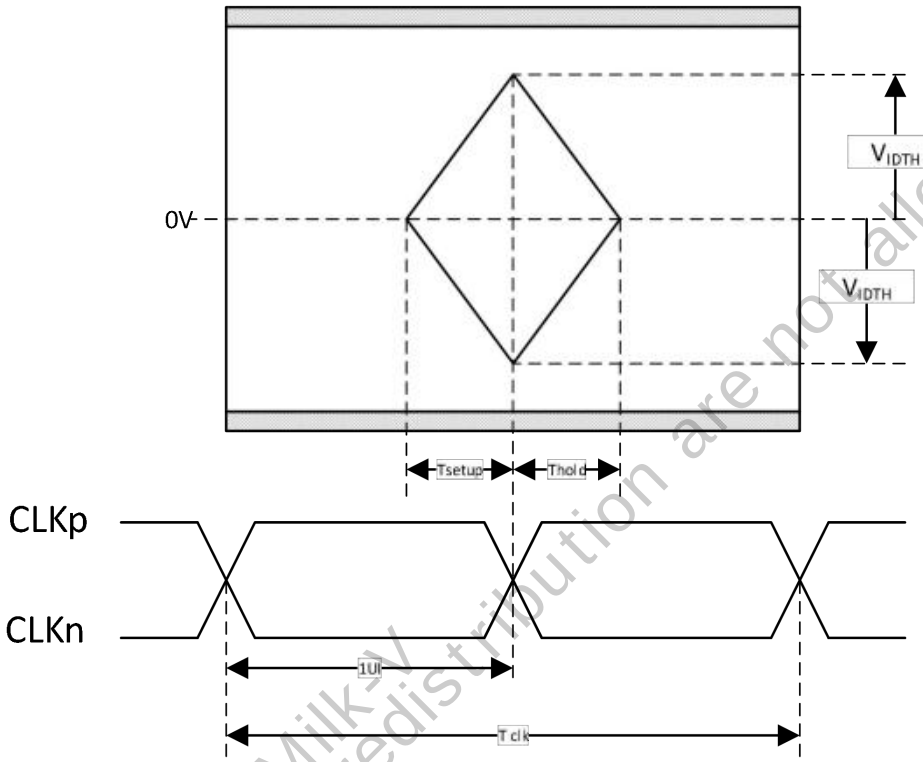


Figure 2- 17. HiSPi Clock Data Timing Diagram

Table 2- 33. HiSPi Timing Requirements

Parameter	Symbol	Min	Typ	Max	Unit
Data Rate	Data Rate	--	--	1.5	Gbps
Unit Interval	UI	666.6	--	--	ns
Differential Clock Period	Tclk	1333.3	--	--	ns
RX Data to Clock Setup Time Tolerance	T_{SETUP}	0.15	--	--	UI
RX Data to Clock Hold Time Tolerance	T_{HOLD}	0.15	--	--	UI
Differential Input Threshold Voltage (VP—VM)	VIDTH(HSSL)	-70	--	70	mV
	VIDTH(HSHI)	-100	--	100	

* UI= 1/(Data Rate) = Tclk/2

2.6.10 SDIO/MMC Timing

The data input and output timing of single edge is shown in Figure 2-18.

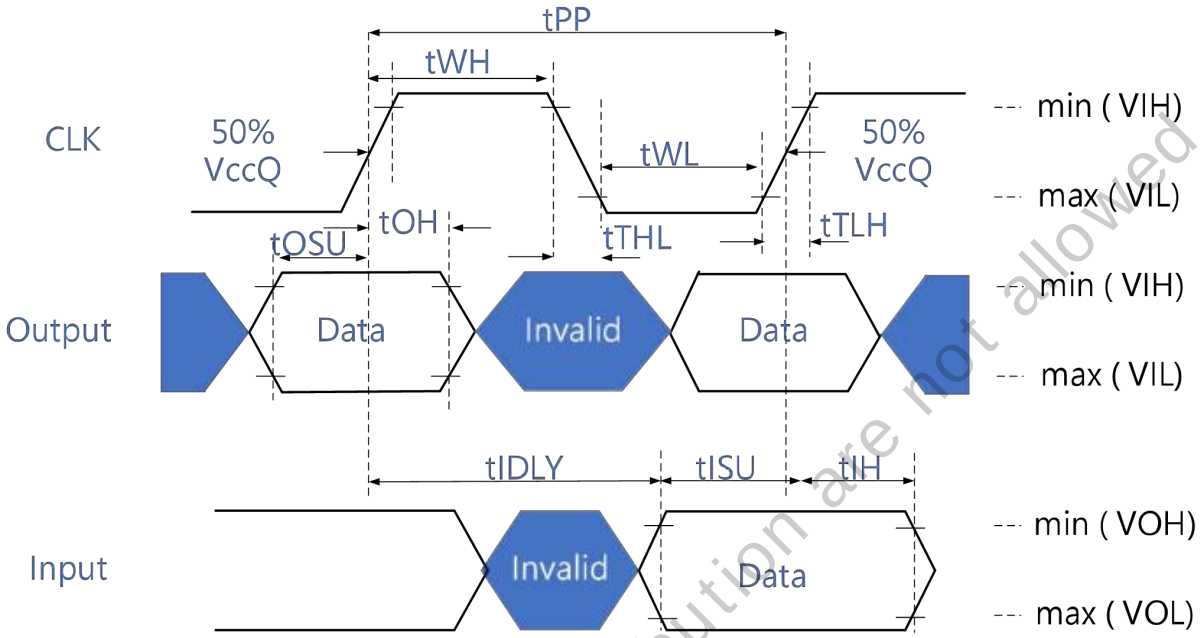


Figure 2- 18 SDIO / MMC Single Edge (SDR) Data Input/ Output Timing Diagram

Table 2- 34 SDIO / MMC Single Edge Defalut Speed (DS) Mode Timing Requirements

Parameter	Symbol	Min	Typ	Max	Unit	Note
Clock CLK						
Clock frequency Data transfer Mode	fPP	0	-	26	MHz	fpp=1/tpp CL≤30pF
Clock frequenyc Idenfification Mode	fOD	0	-	400	KHz	CL≤30pF
Clock high time	tWH	10	-	-	ns	CL≤30pF
Clock low time	tWL	10	-	-	ns	CL≤30pF
Clock rise time	tTLH	-	-	10	ns	CL≤30pF
Clock fall time	tTHL	-	-	10	ns	CL≤30pF
Inputs CMD, DAT (referenced to CLK)						
Input set-up time	tISU	6	-	-	ns	CL≤30pF
Input hold time	tIH	8.3	-	-	ns	CL≤30pF
Outputs CMD, DAT (referenced to CLK)						

Parameter	Symbol	Min	Typ	Max	Unit	Note
Output set-up time	tOSU	5	-	-	ns	CL≤30pF
Output hold time	tOH	5	-	-	ns	CL≤30pF

Table 2- 35 SDIO/MMC Single Edge High speed (HS) Mode Timing Requirements

Parameter	Symbol	Min	Typ	Max	Unit	Note
Clock CLK						
Clock frequency Data transfer Mode	fpp	0	-	52	MHz	fpp=1/tpp CL≤30pF
Clock high time	tWH	6.5	-	-	ns	CL≤30pF
Clock low time	tWL	6.5	-	-	ns	CL≤30pF
Clock rise time	tTLH	-	-	3	ns	CL≤30pF
Clock fall time	tTHL	-	-	3	ns	CL≤30pF
Inputs CMD, DAT (referenced to CLK)						
Input set-up time	tISU	6	-	-	ns	CL≤30pF
Input hold time	tIH	2.5	-	-	ns	CL≤30pF
Outputs CMD, DAT (referenced to CLK)						
Output set-up time	tOSU	6	-	-	ns	CL≤30pF
Output hold time	tOH	3	-	-	ns	CL≤30pF

The timing of double edge data input/output is shown as the figure below.

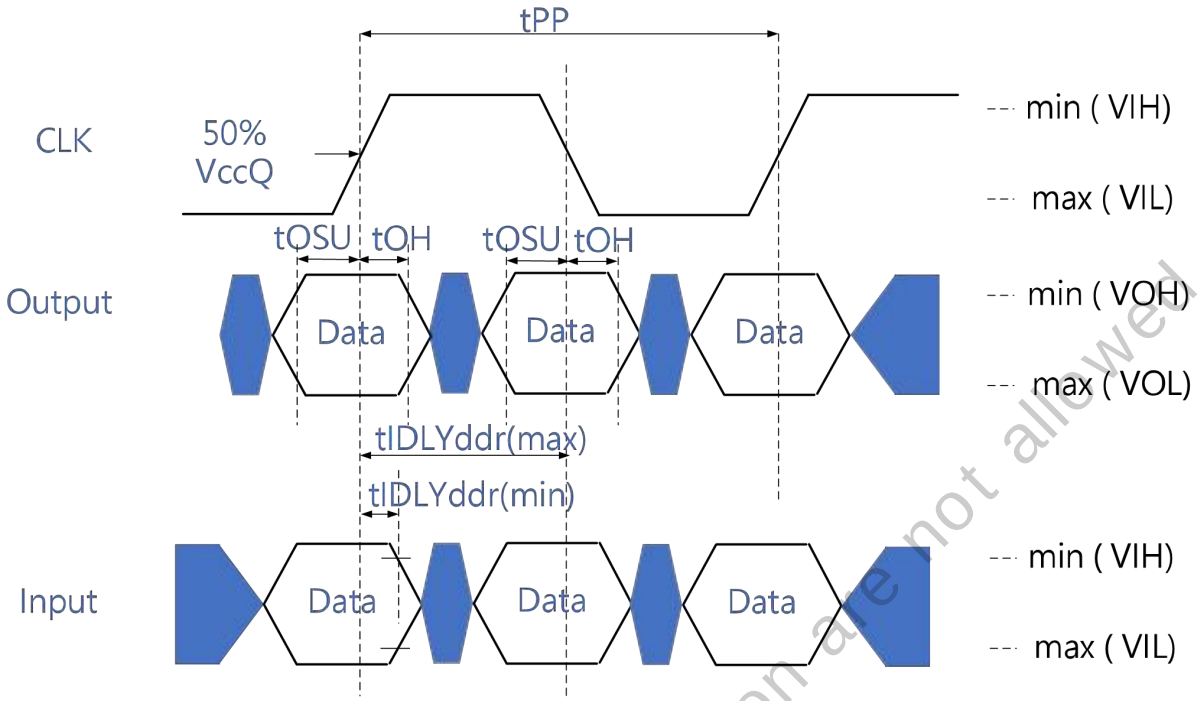


Figure 2- 19 SDIO / MMC Double Edge DDR50 Mode Data Input/ Output Timing Diagram

Table 2- 36 SDIO/MMC Double Edge e DDR50 Mode Timing Requirements

Parameter	Symbol	Min	Typ	Max	Unit	Note
Clock CLK						
Clock frequency Data transfer Mode	fP	0	-	52	MHz	fpp=1/tpp CL≤30pF
Inputs DAT (referenced to CLK)						
Input delay time during data transfer	tIDLYddr	1.5	-	7	ns	CL≤20pF
Outputs DAT (referenced to CLK)						
Output set-up time	tOSU	3	-	-	ns	CL≤20pF
Output hold time	tOH	2.5	-	-	ns	CL≤20pF

The timing diagram of HS200 and SDR104 data input/output is shown as the figure below.

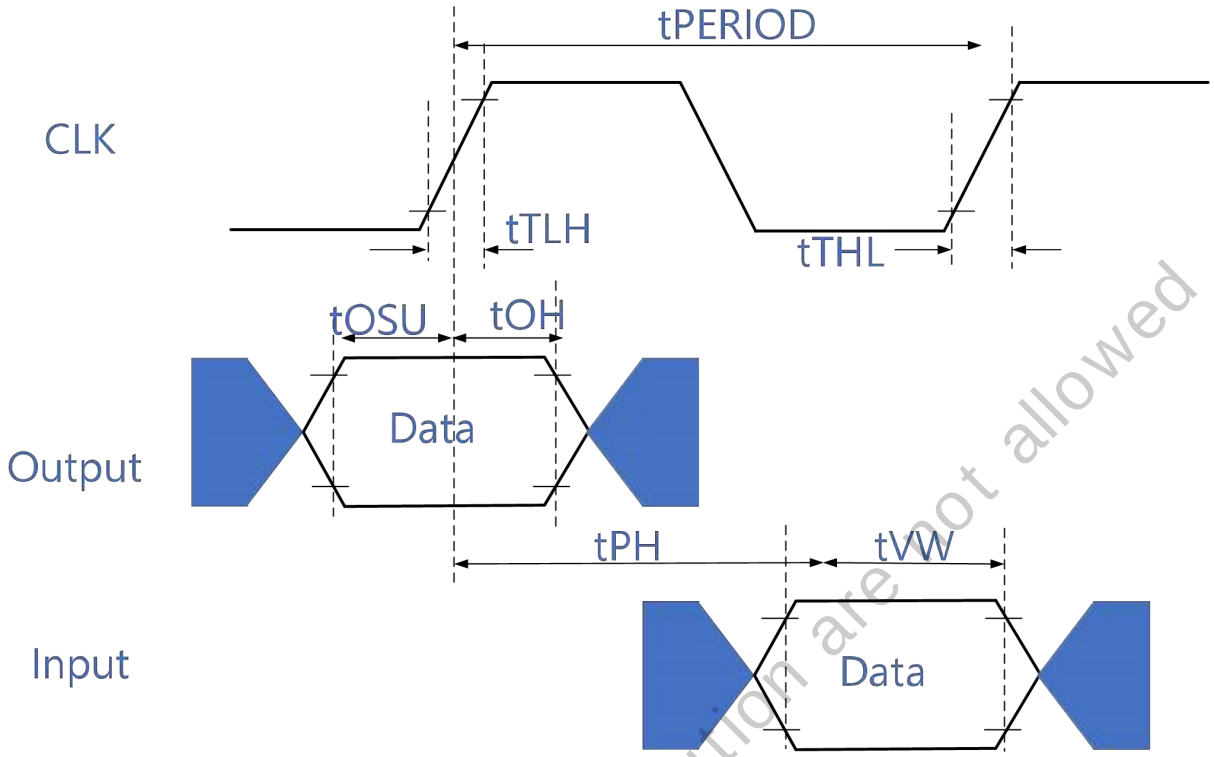


Figure 2- 20 SDIO/MMC HS200 and SDR104 Mode Data Command Input/Output Timing Diagram

Table 2- 37 SDIO/MMC HS200 and SDR104 Mode Output Parameters

Parameter	Symbol	Min	Typ	Max	Unit	Note
Output set-up time	tOSU	1.4	-	-	ns	$C_{DEVICE} \leq 6pF$
Output hold time	tOH	0.8	-	-	ns	

Table 2- 38 SDIO/MMC HS200 and SDR104 Mode Input Parameters

Parameter	Symbol	Min	Typ	Max	Unit	Note
Phase difference between device TX CMD/DAT and RX CLK	tPH	0	-	2	UI	Unit Interval (UI) is one bit nominal time. For 200Mhz, UI=5ns
Input valid data window	tVW	0.575	-	-	UI	TVW=2.88ns at 200MHz

3 System

3.1 Reset

3.1.1 Overview

The reset management module manages the reset sequence of the whole chip、subsystem and functional modules.

3.1.2 Reset Control

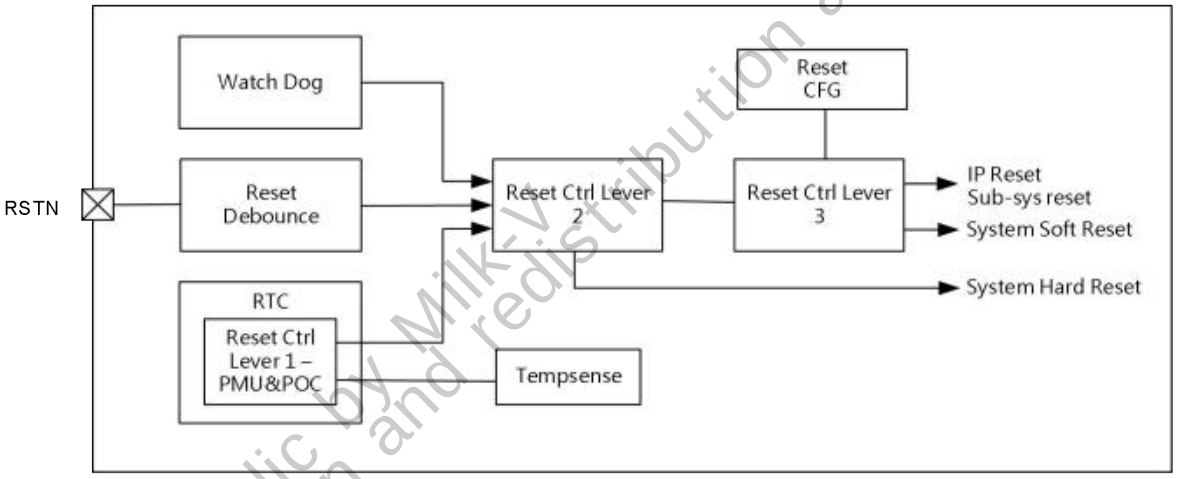


Figure 3- 1 Reset Management Module Block Diagram

3.1.2.1 Power on Reset

Power on reset (POR) is generated by the real-time clock(RTC) module. Refer to section 3.9 real time clock for details.

3.1.2.2 System Hard Reset

The system hard reset is generated by Reset Ctrl Level 2, which is used to reset all subsystems and functional modules of the chip. The reset sources are from:

- Power on reset
- Watchdog reset

- Overheating protection reset
- External reset pin (RSTN)
  - Built in debounce circuit, the effective signal of high and low level of RSTN must reach 6.56ms.

3.1.2.3 Soft Reset

Soft reset control is triggered by configuring the corresponding reset configuration register (Reset CRG). It is realized in Reset Ctrl Level 3. It includes -

- System soft reset: reset the whole chip, except for a few circuits and RTC internal circuits
- Reset of processor subsystem: reset the processor and processor subsystem
- Function subsystem reset: reset each function subsystem and function module
- Function module reset: reset each function module

3.1.2.4 Soft Reset of Processor Subsystem

SOFT_AC_RSTN_0 is used to generate reset to the processor and subsystem. After the configuration register is written to 0, the reset controller will wait for 24us delay before triggering the corresponding processor reset. During this period, the processor should stop accessing the bus to avoid the bus hanging after reset. After triggering reset, the corresponding reset signal will be automatically released after 8us, and the processor and processor subsystem will complete the reset and start up.

3.1.2.5 Soft Reset of Function Subsystem and Function Module

SOFT_RSTN_0~3 are used to generate reset to each function module. The reset signal will not be cleared automatically. Therefore, after the software configures the corresponding register as 0 to trigger the reset, it also needs to be configured as 1 to release the reset. Before reset, make sure that the built-in DMA access to the bus and processor access to the module are idle. Otherwise, the reset will fail and cause system hang up.

3.1.3 Reset Configuration Register

3.1.3.1 Overview of Reset Configuration Register

Base address 0x03003000

Name	Address Offset	Description
SOFT_RSTN_0	0x000	soft-reset ctrl register 0
SOFT_RSTN_1	0x004	soft-reset ctrl register 1
SOFT_RSTN_2	0x008	soft-reset ctrl register 2
SOFT_RSTN_3	0x00c	soft-reset ctrl register 3
SOFT_CPUAC_RSTN	0x020	CPU auto clear soft-reset ctrl register
SOFT_CPU_RSTN	0x024	CPU soft-reset ctrl register

3.1.3.2 Description of Reset Configuration Register

SOFT_RSTN_0

Offset Address: 0x000

Bits	Name	Access	Description	Reset
1:0	Reserved			
2	reg_soft_reset_x_ddr	R/W	DDR system software reset (active low)	0x1
3	reg_soft_reset_x_h264c	R/W	H264 IP software reset (active low)	0x1
4	reg_soft_reset_x_jpeg	R/W	JPEG IP software reset (active low)	0x1
5	reg_soft_reset_x_h265c	R/W	H265 IP software reset (active low)	0x1
6	reg_soft_reset_x_vipsys	R/W	VIP system software reset (active low)	0x1
7	reg_soft_reset_x_tdma	R/W	TPU_DMA IP software reset (active low)	0x1
8	reg_soft_reset_x_tpu	R/W	TPU IP software reset (active low)	0x1
9	reg_soft_reset_x_tpusys	R/W	TPU system software reset (active low)	0x1
10	Reserved			
11	reg_soft_reset_x_usb	R/W	USB IP software reset (active low)	0x1
12	reg_soft_reset_x_eth0	R/W	ETH0 IP software reset (active low)	0x1
13	Reserved	R/W		0x1
14	reg_soft_reset_x_nand	R/W	NAND IP software reset (active low)	0x1
15	Reserved	R/W		0x1

Bits	Name	Access	Description	Reset
16	reg_soft_reset_x_sd0	R/W	SD0 IP software reset (active low)	0x1
17	Reserved			
18	reg_soft_reset_x_sdma	R/W	SDMA IP software reset (active low)	0x1
19	reg_soft_reset_x_i2s0	R/W	I2S0 IP software reset (active low)	0x1
20	reg_soft_reset_x_i2s1	R/W	I2S1 IP software reset (active low)	0x1
21	reg_soft_reset_x_i2s2	R/W	I2S2 IP software reset (active low)	0x1
22	reg_soft_reset_x_i2s3	R/W	I2S3 IP software reset (active low)	0x1
23	reg_soft_reset_x_uart0	R/W	UART0 IP software reset (active low)	0x1
24	reg_soft_reset_x_uart1	R/W	UART1 IP software reset (active low)	0x1
25	reg_soft_reset_x_uart2	R/W	UART2 IP software reset (active low)	0x1
26	reg_soft_reset_x_uart3	R/W	UART3 IP software reset (active low)	0x1
27	reg_soft_reset_x_i2c0	R/W	I2C0 IP software reset (active low)	0x1
28	reg_soft_reset_x_i2c1	R/W	I2C1 IP software reset (active low)	0x1
29	reg_soft_reset_x_i2c2	R/W	I2C2 IP software reset (active low)	0x1
30	reg_soft_reset_x_i2c3	R/W	I2C3 IP software reset (active low)	0x1
31	reg_soft_reset_x_i2c4	R/W	I2C4 IP software reset (active low)	0x1

SOFT_RSTN_1

Offset Address: 0x004

Bits	Name	Access	Description	Reset
0	reg_soft_reset_x_pwm0	R/W	PWM0 IP software reset (active low)	0x1
1	reg_soft_reset_x_pwm1	R/W	PWM1 IP software reset (active low)	0x1
2	reg_soft_reset_x_pwm2	R/W	PWM2 IP software reset (active low)	0x1
3	reg_soft_reset_x_pwm3	R/W	PWM3 IP software reset (active low)	0x1
7:4	Reserved			
8	reg_soft_reset_x_spi0	R/W	SPI0 IP software reset (active low)	0x1
9	reg_soft_reset_x_spi1	R/W	SPI1 IP software reset (active low)	0x1
10	reg_soft_reset_x_spi2	R/W	SPI2 IP software reset (active low)	0x1
11	reg_soft_reset_x_spi3	R/W	SPI3 IP software reset (active low)	0x1

Bits	Name	Access	Description	Reset
12	reg_soft_reset_x_gpio0	R/W	GPIO0 IP software reset (active low)	0x1
13	reg_soft_reset_x_gpio1	R/W	GPIO1 IP software reset (active low)	0x1
14	reg_soft_reset_x_gpio2	R/W	GPIO2 IP software reset (active low)	0x1
15	reg_soft_reset_x_efuse	R/W	EFUSE IP software reset (active low)	0x1
16	reg_soft_reset_x_wdt	R/W	WDT0 IP software reset (active low)	0x1
17	reg_soft_reset_x_ahb_rom	R/W	ROM IP software reset (active low)	0x1
18	reg_soft_reset_x_spic	R/W	SPIC IP software reset (active low)	0x1
19	reg_soft_reset_x_tempsen	R/W	TEMPSEN IP software reset (active low)	0x1
20	reg_soft_reset_x_saradc	R/W	SARADC IP software reset (active low)	0x1
25:21	Reserved			
26	reg_soft_reset_x_combo_phy0	R/W	USB_PHY IP software reset (active low)	0x1
28:27	Reserved			
29	reg_soft_reset_x_spi_nand	R/W	NAND IP software reset (active low)	0x1
30	reg_soft_reset_x_se	R/W	SE IP software reset (active low)	0x1
31	Reserved			

SOFT_RSTN_2

Offset Address: 0x008

Bits	Name	Access	Description	Reset
9:0	Reserved			
10	reg_soft_reset_x_uart4	R/W	UART4 IP software reset (active low)	0x1
11	reg_soft_reset_x_gpio3	R/W	GPIO3 IP software reset (active low)	0x1
12	reg_soft_reset_x_system	R/W	SYSTEM software reset (active low)	0x1
13	reg_soft_reset_x_timer	R/W	TIMER IP software reset (active low)	0x1
14	reg_soft_reset_x_timer0	R/W	TIMER0 IP software reset (active low)	0x1
15	reg_soft_reset_x_timer1	R/W	TIMER1 IP software reset (active low)	0x1
16	reg_soft_reset_x_timer2	R/W	TIMER2 IP software reset (active low)	0x1
17	reg_soft_reset_x_timer3	R/W	TIMER3 IP software reset (active low)	0x1
18	reg_soft_reset_x_timer4	R/W	TIMER4 IP software reset (active low)	0x1
19	reg_soft_reset_x_timer5	R/W	TIMER5 IP software reset (active low)	0x1

Bits	Name	Access	Description	Reset
20	reg_soft_reset_x_timer6	R/W	TIMER6 IP software reset (active low)	0x1
21	reg_soft_reset_x_timer7	R/W	TIMER7 IP software reset (active low)	0x1
22	reg_soft_reset_x_wgn0	R/W	WGN0 IP software reset (active low)	0x1
23	reg_soft_reset_x_wgn1	R/W	WGN1 IP software reset (active low)	0x1
24	reg_soft_reset_x_wgn2	R/W	WGN2 IP software reset (active low)	0x1
25	reg_soft_reset_x_keyscan	R/W	KEYSCAN IP software reset (active low)	0x1
26	Reserved			
27	reg_soft_reset_x_auddac	R/W	AUDDAC IP software reset (active low)	0x1
28	reg_soft_reset_x_auddac_apb	R/W	AUDDAC APB software reset (active low)	0x1
29	reg_soft_reset_x_audadc	R/W	AUDADC IP software reset (active low)	0x1
30	Reserved			
31	reg_soft_reset_x_vcsys	R/W	VCSYS SYS software reset (active low)	0x1

SOFT_RSTN_3

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
0	reg_soft_reset_x_ethphy	R/W	ETHPHY IP software reset (active low)	0x1
1	reg_soft_reset_x_ethphy_apb	R/W	ETHPHY APB REG software reset (active low)	0x1
2	reg_soft_reset_x_audsrc	R/W	AUDSRC IP software reset (active low)	0x1
3	reg_soft_reset_x_vip_cam0	R/W	VIP CAM0 IP software reset (active low)	0x1
4	reg_soft_reset_x_wdt1	R/W	WDT1 IP software reset (active low)	0x1
5	reg_soft_reset_x_wdt2	R/W	WDT2 IP software reset (active low)	0x1
31:6	Reserved			

SOFT_CPUAC_RSTN

Offset Address: 0x020

Write Lock: SOFT_CPUAC_RSTN_wr_lock

Bits	Name	Access	Description	Reset
0	reg_auto_clear_reset_x_cpucore0	R/W	CPUCORE0 auto_clear_reset (active low)	0x1
1	reg_auto_clear_reset_x_cpucore1	R/W	CPUCORE1 auto_clear_reset (active low)	0x1
2	reg_auto_clear_reset_x_cpucore2	R/W	CPUCORE2 auto_clear_reset (active low)	0x1
3	reg_auto_clear_reset_x_cpucore3	R/W	CPUCORE3 auto_clear_reset (active low)	0x1

Bits	Name	Access	Description	Reset
4	reg_auto_clear_reset_x_cpusys0	R/W	CPUSYS0 auto_clear_reset (active low)	0x1
5	reg_auto_clear_reset_x_cpusys1	R/W	CPUSYS1 auto_clear_reset (active low)	0x1
6	reg_auto_clear_reset_x_cpusys2	R/W	CPUSYS2 auto_clear_reset (active low)	0x1
31:7	Reserved			

SOFT_CPU_RSTN

Offset Address: 0x024

Bits	Name	Access	Description	Reset
0	reg_soft_reset_x_cpucore0	R/W	CPUCORE0 soft reset (active low)	0x0
1	reg_soft_reset_x_cpucore1	R/W	CPUCORE1 soft reset (active low)	0x0
2	reg_soft_reset_x_cpucore2	R/W	CPUCORE2 soft reset (active low)	0x0
3	reg_soft_reset_x_cpucore3	R/W	CPUCORE3 soft reset (active low)	0x0
4	reg_soft_reset_x_cpusys0	R/W	CPUSYS0 soft reset (active low)	0x0
5	reg_soft_reset_x_cpusys1	R/W	CPUSYS1 soft reset (active low)	0x0
6	reg_soft_reset_x_cpusys2	R/W	CPUSYS2 soft reset (active low)	0x0
31:7	Reserved			

3.2 Clock

3.2.1 Overview

The clock management module manages the chip clock. It includes –

- Management and control of clock input
- PLL clock source and related frequency multiply and division configuration
- Clock frequency division and control
- Generate clock for each module
- Management and control for each clock

3.2.2 Function Block Diagram

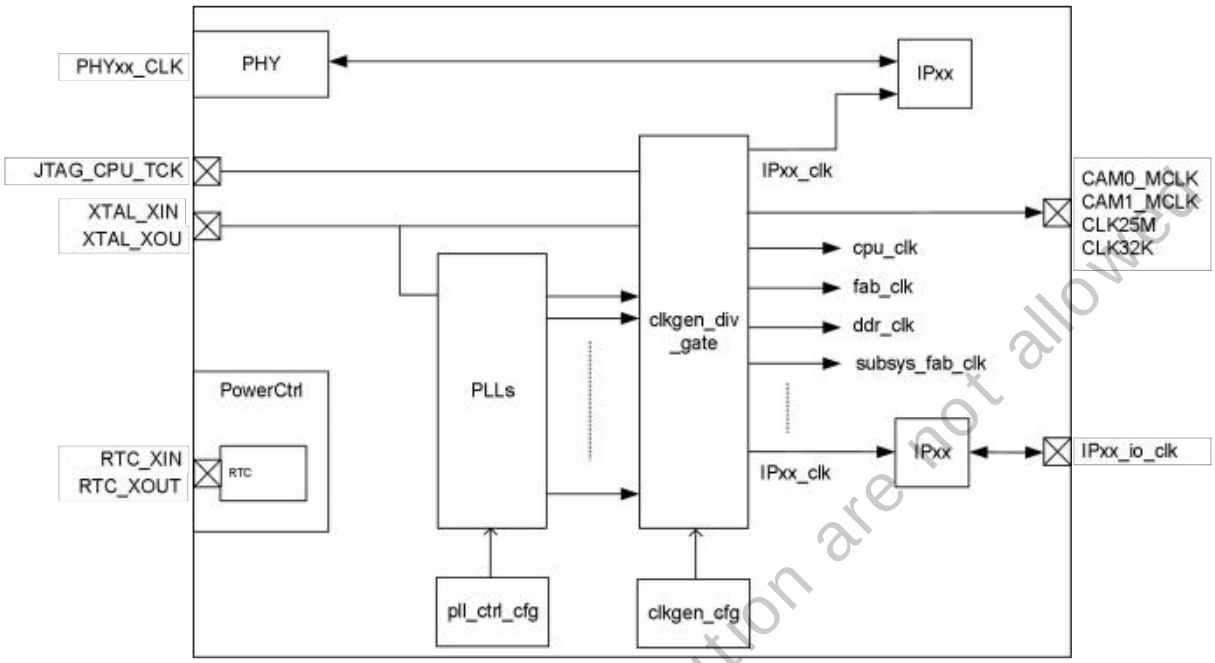


Figure 3- 2 Clock Management Block Diagram

XTAL_ XIN is reference clock of PLL and should be connected with 25MHz crystal. RTC_ XIN is the reference clock of RTC and should be connected with 32.768KHz crystal.

3.2.3 Clock Resource and Frequency Division Structure

The system clock mainly comes from external XTAL、PLLs and external input clocks. As shown in Figure 3-3, each IP generally has clock source from XTAL or PLLs. After passing through the frequency division circuit, clocks are generated and selected to be the clock of IPs or subsystems.

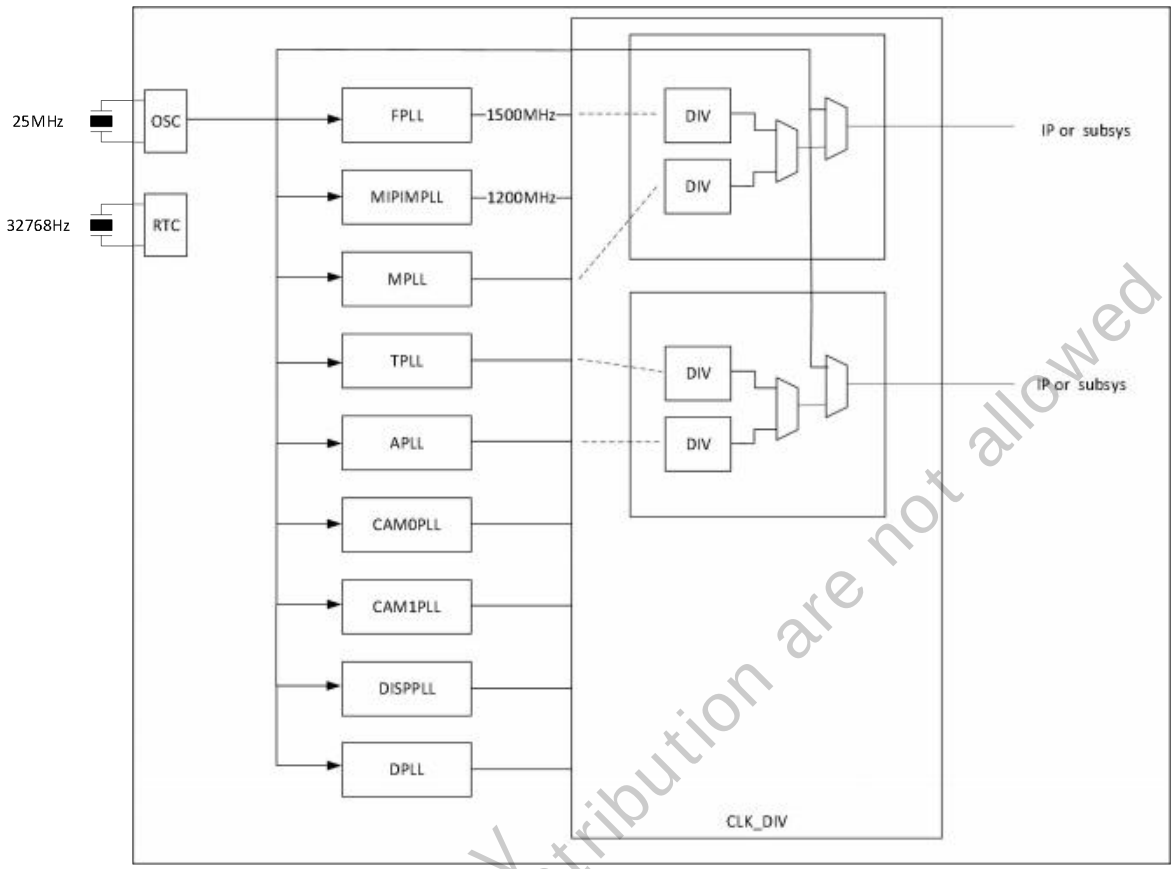


Figure 3- 3 Clock Source Frequency Division Diagram

3.2.4 PLL Configuration

As shown in Table 3-1, the chip has built-in 9 PLLs (excluding Analog IP built-in PLL), which are categorized into integer frequency multiplication and fractional frequency multiplication.

Table 3- 1 PLL Configuration Parameters

PLL	配置寄存器	下电控制寄存器	预设频率	PLL 型态
FPLL	fpll_csr	fpll_pwd (default On)	1500MHz	整数倍频
MIPIMPLL	mipimpll_csr	mipimpll_pwd (default On)	900MHz	整数倍频
MPLL	mpll_csr mpll_ssc_syn_ctrl mpll_ssc_syn_set mpll_ssc_syn_span mpll_ssc_syn_setp	mpll_pwd (default On)	1000MHz	整/分数倍频
TPLL	tpll_csr	tpll_pwd (default On)	1400MHz	整/分数倍频

PLL	配置寄存器	下电控制寄存器	预设频率	PLL 型态
	tpll_ssc_syn_ctrl tpll_ssc_syn_set tpll_ssc_syn_span tpll_ssc_syn_setp			
APLL	apll_csr apll_ssc_syn_ctrl apll_ssc_syn_set apll_ssc_syn_span apll_ssc_syn_setp	apll_pwd(default On)	1050MHz	整/分数倍频
CAM0PLL	cam0pll_csr cam0pll_ssc_syn_ctrl cam0pll_ssc_syn_set cam0pll_ssc_syn_span cam0pll_ssc_syn_setp	cam0pll_pwd(default On)	1050MHz	整/分数倍频
CAM1PLL	cam1pll_csr cam1pll_ssc_syn_ctrl cam1pll_ssc_syn_set cam1pll_ssc_syn_span cam1pll_ssc_syn_setp	cam1pll_pwd(default On)	1025MHz	整/分数倍频
DISPPLL	disppll_csr disppll_ssc_syn_ctrl disppll_ssc_syn_set disppll_ssc_syn_span disppll_ssc_syn_setp	disppll_pwd(default On)	1200MHz	整/分数倍频

3.2.4.1 Integer Frequency Multiplication PLL

The process of integer PLL adjustment is as follows.

1. Turn off clocks generated from this PLL or select others stable clock to be the clocks' source.
2. Configure*_pll_CSR register, configured according to integer PLL parameter table
3. Clear *_pll_pwd

Table 3- 2 Interger PLL Configuration Parameters

PLL 参数	范围	注意事项
PLL_REF	25MHz~2500MHz	
PLL_VCO	800MHz~2500MHz	
Pre_div_sel	1~127	PLL_VCO = PLL_REF*Div_sel/Pre_div_sel

PLL 参数	范围	注意事项
Div_sel	6~127	PLL_FOUT = PLL_VCO/Post_div_sel
Post_div_sel	1~127	
Ictrl	0~7	0.2< 1.84*(1+Mode)*(1+Ictrl)/2/Div_sel <=0.35
Mode	0~3	

3.2.4.2 Fractional Frequency Multiplication PLL

The process of fractional PLL adjustment is as follows.

1. Turn off clocks generated from this PLL or select others stable clock to be the clocks' source.
2. Configure *_ssc_syn_src_en to enable the synthesizer clock
3. Configure *_ssc_syn_set according to PLL frequency requirement
4. Toggle *_ssc_syn_up to make the configuration take effect
5. Configure *_ pll_csr register, configured according to integer PLL parameter table
6. Clear *_pll_pwd

Table 3- 3 Fractional PLL Configuration Parameters

PLL 参数	范围	注意事项
ssc_freq_in	DSIPLL: 1.2GHz DDRPLL: 1.5GHz Others 600MHz	Default clock gated , need enable
ssc_syn_set	> 4.x * 2^26	ssc_freq_in*div_sel*2^26/PLL_VCO
PLL_REF	100M~2500MHz	
PLL_VCO	800MHz~2500MHz	
Div_sel	6~127	PLL_VCO = PLL_REF*div_sel FOUT = PLL_VCO/Post_div_sel
Post_div_sel	1~127	
Ictrl	0~7	0.1< 1.84*(1+Mode)*(1+Ictrl)/2/Div_sel <=0.24
Mode	0~3	

3.2.5 CLK_DIV Clock Frequency Division Configuration

Below is the clock resource table. This table indicates the configurable clock source, preset clock frequency and frequency division of each clock. The software can switch

the clock source from XTAL to PLL after boot, and adjust the clock frequency division configuration.

Table 3- 4 Clock Source and Preset Frequency Division Parameters

CLK_NAME	XTAL	DIV	SW	PLL SRC/DIV/FREQ	DIV_IN0 SRC				DIV_IN1 SRC
clk_cpu_axi0	Y	Y		fpll/(3)/500M	fpll	dsppll			
clk_cpu_gic	Y	Y		fpll/(5)/300M	fpll				
clk_tpu	Y	Y		fpll/(3)/500M	tpll	a0pll	mipimpll	fpll	
clk_sd0	Y	Y		fpll/(15)/100M	fpll	dsppll			
clk_sd1	Y	Y		fpll/(15)/100M	fpll	dsppll			
clk_spi_nand	Y	Y		fpll/(8)/187.5M	fpll	dsppll			
clk_sdma_aud0	Y	Y		a0pll/(18)/58.3M	a0pll	a24m			
clk_sdma_aud1	Y	Y		a0pll/(18)/58.3M	a0pll	a24m			
clk_sdma_aud2	Y	Y		a0pll/(18)/58.3M	a0pll	a24m			
clk_sdma_aud3	Y	Y		a0pll/(18)/58.3M	a0pll	a24m			
clk_pwm	Y	Y		fpll/(10)/150M	fpll	dsppll			
clk_uart	Y	Y		xtal/(1)/25M	xtal	dsppll			
clk_axi4	Y	Y		fpll/(5)/300M	fpll	dsppll			
clk_axi6	Y	Y		fpll/(15)/100M	fpll				
clk_axi_vip	Y	Y		fpll/(6)/250M	mipimpll	cam0pll	dsppll	fpll	
clk_src_vip_sys_0	Y	Y		fpll/(6)/250M	mipimpll	cam0pll	dsppll	fpll	
clk_src_vip_sys_1	Y	Y		fpll/(5)/300M	mipimpll	cam0pll	dsppll	fpll	
clk_axi_video_codec	Y	Y		mipimpll/(2)/450M	a0pll	mipimpll	camlpll	fpll	
clk_vc_src0	Y	Y		mipimpll/(2)/450M	dsppll	mipimpll	camlpll	fpll	
clk_spi	Y	Y		fpll/(8)/187.5M	fpll				
clk_i2c	Y	Y		clk_axi6/(1)/100M	clk_axi6				
clk_src_vip_sys_2	Y	Y		dsppll/(2)/600M	mipimpll	cam0pll	dsppll	fpll	
clk_audsrc	Y	Y		a0pll/(18)/58.3M	a0pll	a24m			
clk_ap_debug	Y	Y		fpll/(5)/300M	fpll				
clk_src_rtc_sys_0	Y	Y		fpll/(5)/300M	fpll				
clk_c906_0	Y	Y	Y	fpll/(2)/750M	tpll	a0pll	mipimpll	mpll	fpll
clk_c906_1	Y	Y	Y	fpll/(3)/500M	tpll	a0pll	dsppll	mpll	fpll
clk_src_vip_sys_3	Y	Y		mipimpll/(2)/450M	mipimpll	cam0pll	dsppll	fpll	
clk_src_vip_sys_4	Y	Y		dsppll/(3)/400M	mipimpll	cam0pll	dsppll	fpll	

3.2.5.1 IP/SYS Source and Clock Frequency Division Configuration

1. Turn off the IP clock. If the clock cannot be turned off, it should be configured to the stable clock first.
   A、CPU frequency scaling: Configure clk_sel_0 to switch to SRC1 to avoid too low frequency.
   B、IP frequency scaling: configure clk_byp_0/1 to switch the clock to XTAL.
2. Configure the clock source and frequency divider to be adjusted.
3. Configure bit[2] of frequency divider register, and then the clock divider configuration can take effect.
4. Select the clock source to the configured clock divider.

3.2.5.2 MCLK0/MCLK1

1. MCLK0/MCLK1 provide external sensor reference clock.
2. Configure CAM0PLL, clk_cam0_src_div, and clk_cam0_src_div to provide the appropriate MCLK0/MCLK1 output frequency.

3.2.5.3 Clk_A24M

1. clk_a24m can be used as the audio clock when performance is acceptable.
2. Configure apll_frac_div_ctrl, apll_frac_div_m, and apll_frac_div_n to generate the required audio clock source.
3. The frequency of clk_a24m is 900MHz * N/M/2.

3.2.6 PLL CRG Register Overview

PLL_G2 base address : 0x03002800

Name	Address Offset	Description
pll_g2_ctrl	0x000	Group2 PLL Ctrl register
pll_g2_status	0x004	Group2 PLL Status register
mipimpll_csr	0x008	MIPIMPLL Ctrl register
apll0_csr	0x00c	APLL0 Ctrl register
disppll_csr	0x010	DISPPLL Ctrl register
cam0pll_csr	0x014	CAM0PLL Ctrl register
cam1pll_csr	0x018	CAM1PLL Ctrl register
pll_g2_ssc_syn_ctrl	0x040	Group2 PLL Synthesizer ctrl register
apll_ssc_syn_ctrl	0x050	APLL synthesizer ctrl register
apll_ssc_syn_set	0x054	APLL synthesizer set register

Name	Address Offset	Description
disppll_ssc_syn_ctrl	0x060	DISPPLL synthesizer ctrl register
disppll_ssc_syn_set	0x064	DISPPLL synthesizer set register
cam0pll_ssc_syn_ctrl	0x070	CAM0PLL synthesizer ctrl register
cam0pll_ssc_syn_set	0x074	CAM0PLL synthesizer set register
cam1pll_ssc_syn_ctrl	0x080	CAM1PLL synthesizer ctrl register
cam1pll_ssc_syn_set	0x084	CAM1PLL synthesizer set register
apll_frac_div_ctrl	0x090	APLL frac divider ctrl register
apll_frac_div_m	0x094	APLL frac divider M parameter
apll_frac_div_n	0x098	APLL frac divider N parameter
mipimpll_clk_csr	0x0a0	MIPIMPLL clock Ctrl register
a0pll_clk_csr	0x0a4	a0pll clock Ctrl register
disppll_clk_csr	0x0a8	disppll clock Ctrl register
cam0pll_clk_csr	0x0ac	cam0pll clock Ctrl register
cam1pll_clk_csr	0x0b0	cam1pll clock Ctrl register
clk_cam0_src_div	0x0c0	clk_cam0_src_div
clk_cam1_src_div	0x0c4	clk_cam1_src_div

PLL_G6 base address : 0x03002900

Name	Address Offset	Description
pll_g6_ctrl	0x000	Group6 PLL Ctrl register
pll_g6_status	0x004	Group6 PLL Status register
mpll_csr	0x008	MPLL Ctrl register
tpll_csr	0x00c	TPLL Ctrl register
fpll_csr	0x010	FPLL Ctrl register
pll_g6_ssc_syn_ctrl	0x040	Group6 PLL Synthesizer ctrl register
dpll_ssc_syn_ctrl	0x050	dpll synthesizer ctrl register
dpll_ssc_syn_set	0x054	dpll synthesizer set register
dpll_ssc_syn_span	0x058	dpll synthesizer span register
dpll_ssc_syn_step	0x05c	dpll synthesizer step register
mpll_ssc_syn_ctrl	0x060	mpll synthesizer ctrl register
mpll_ssc_syn_set	0x064	mpll synthesizer set register
mpll_ssc_syn_span	0x068	mpll synthesizer span register
mpll_ssc_syn_step	0x06c	mpll synthesizer step register
tpll_ssc_syn_ctrl	0x070	tpll synthesizer ctrl register
tpll_ssc_syn_set	0x074	tpll synthesizer set register
tpll_ssc_syn_span	0x078	tpll synthesizer span register
tpll_ssc_syn_step	0x07c	tpll synthesizer step register

3.2.7 PLL CRG Register Overview

3.2.7.1 PLL_G2 Register Overview

pll_g2_ctrl

Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	mipimpll_pwd	R/W	pll power down	0x0
3:1	Reserved			
4	apll0_pwd	R/W	pll power down	0x0
7:5	Reserved			
8	disppll_pwd	R/W	pll power down	0x0
11:9	Reserved			
12	cam0pll_pwd	R/W	pll power down	0x0
15:13	Reserved			
16	cam1pll_pwd	R/W	pll power down	0x0
31:17	Reserved			

pll_g2_status

Offset Address: 0x004

Bits	Name	Access	Description	Reset
0	updating_mipimpll_val	RO	pll setting update status	
1	updating_apll0_val	RO	pll setting update status	
2	updating_disppll_val	RO	pll setting update status	
3	updating_cam0pll_val	RO	pll setting update status	
4	updating_cam1pll_val	RO	pll setting update status	
15:5	Reserved			
16	mipimpll_lock	RO	pll lock status	
17	apll0_lock	RO	pll lock status	
18	disppll_lock	RO	pll lock status	
19	cam0pll_lock	RO	pll lock status	
20	cam1pll_lock	RO	pll lock status	
31:21	Reserved			

mipimpll_csr

Offset Address: 0x008

Bits	Name	Access	Description	Reset
6:0	mipimpll_pre_div_sel	R/W	pll pre_div_sel setting	0x0
7	Reserved			
14:8	mipimpll_post_div_sel	R/W	pll post_div_sel setting	0x0
16:15	mipimpll_sel_mode	R/W	pll mode setting	0x0
23:17	mipimpll_div_sel	R/W	pll div_sel setting	0x0
26:24	mipimpll_ictrl	R/W	pll ictrl setting	0x0
31:27	Reserved			

apll0_csr

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
6:0	apll0_pre_div_sel	R/W	pll pre_div_sel setting	0x0
7	Reserved			
14:8	apll0_post_div_sel	R/W	pll post_div_sel setting	0x0
16:15	apll0_sel_mode	R/W	pll mode setting	0x0
23:17	apll0_div_sel	R/W	pll div_sel setting	0x0
26:24	apll0_ictrl	R/W	pll ictrl setting	0x0
31:27	Reserved			

disppll_csr

Offset Address: 0x010

Bits	Name	Access	Description	Reset
6:0	disppll_pre_div_sel	R/W	pll pre_div_sel setting	0x0
7	Reserved			
14:8	disppll_post_div_sel	R/W	pll post_div_sel setting	0x0
16:15	disppll_sel_mode	R/W	pll mode setting	0x0
23:17	disppll_div_sel	R/W	pll div_sel setting	0x0
26:24	disppll_ictrl	R/W	pll ictrl setting	0x0
31:27	Reserved			

cam0pll_csr

Offset Address: 0x014

Bits	Name	Access	Description	Reset
6:0	cam0pll_pre_div_sel	R/W	pll pre_div_sel setting	0x0
7	Reserved			
14:8	cam0pll_post_div_sel	R/W	pll post_div_sel setting	0x0
16:15	cam0pll_sel_mode	R/W	pll mode setting	0x0
23:17	cam0pll_div_sel	R/W	pll div_sel setting	0x0
26:24	cam0pll_ictrl	R/W	pll ictrl setting	0x0
31:27	Reserved			

cam1pll_csr

Offset Address: 0x018

Bits	Name	Access	Description	Reset
6:0	cam1pll_pre_div_sel	R/W	pll pre_div_sel setting	0x0
7	Reserved			
14:8	cam1pll_post_div_sel	R/W	pll post_div_sel setting	0x0
16:15	cam1pll_sel_mode	R/W	pll mode setting	0x0
23:17	cam1pll_div_sel	R/W	pll div_sel setting	0x0
26:24	cam1pll_ictrl	R/W	pll ictrl setting	0x0
31:27	Reserved			

pll_g2_ssc_syn_ctrl

Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	reg_mipimpll_sel_syn_clk	R/W	mipimpll gen synthesizer clock source 0:450M	0x1

Bits	Name	Access	Description	Reset
			1:900M	
1	reg_dsi_ssc_syn_src_en	R/W	pll synthesizer clock enable	0x0
2	reg_apll_ssc_syn_src_en	R/W	pll synthesizer clock enable	0x0
3	reg_disppll_ssc_syn_src_en	R/W	pll synthesizer clock enable	0x0
4	reg_cam0pll_ssc_syn_src_en	R/W	pll synthesizer clock enable	0x0
5	reg_cam1pll_ssc_syn_src_en	R/W	pll synthesizer clock enable	0x0
31:6	Reserved			

apll_ssc_syn_ctrl

Offset Address: 0x050

Bits	Name	Access	Description	Reset
0	reg_apll_ssc_syn_sw_up	W1T	pll synthesizer software update	
5:1	Reserved			
6	reg_apll_ssc_syn_fix_div	R/W		0x0
31:7	Reserved			

apll_ssc_syn_set

Offset Address: 0x054

Bits	Name	Access	Description	Reset
31:0	reg_apll_ssc_syn_set	R/W	pll synthesizer fraction setting: [31:26] integer 6 bits [26:0] decimal 26 bits	0x0

disppll_ssc_syn_ctrl

Offset Address: 0x060

Bits	Name	Access	Description	Reset
0	reg_disppll_ssc_syn_sw_up	W1T	pll synthesizer software update	
5:1	Reserved			
6	reg_disppll_ssc_syn_fix_div	R/W		0x0
31:7	Reserved			

disppll_ssc_syn_set

Offset Address: 0x064

Bits	Name	Access	Description	Reset
31:0	reg_disppll_ssc_syn_set	R/W	pll synthesizer fraction setting: [31:26] integer 6 bits [26:0] decimal 26 bits	0x0

cam0pll_ssc_syn_ctrl

Offset Address: 0x070

Bits	Name	Access	Description	Reset
0	reg_cam0pll_ssc_syn_sw_up	W1T	pll synthesizer software update	
5:1	Reserved			
6	reg_cam0pll_ssc_syn_fix_div	R/W		0x0
31:7	Reserved			

cam0pll_ssc_syn_set

Offset Address: 0x074

Bits	Name	Access	Description	Reset
31:0	reg_cam0pll_ssc_syn_set	R/W	pll synthesizer fraction setting: [31:26] integer 6 bits [26:0] decimal 26 bits	0x0

cam1pll_ssc_syn_ctrl

Offset Address: 0x080

Bits	Name	Access	Description	Reset
0	reg_cam1pll_ssc_syn_sw_up	W1T	pll synthesizer software update	
5:1	Reserved			
6	reg_cam1pll_ssc_syn_fix_div	R/W		0x0
31:7	Reserved			

cam1pll_ssc_syn_set

Offset Address: 0x084

Bits	Name	Access	Description	Reset
31:0	reg_cam1pll_ssc_syn_set	R/W	pll synthesizer fraction setting: [31:26] integer 6 bits [26:0] decimal 26 bits	0x0

apll_frac_div_ctrl

Offset Address: 0x090

Bits	Name	Access	Description	Reset
0	reg_apll_frac_div_clk_en	R/W	a24m clock src enable	0x0
1	reg_apll_frac_div_en	R/W	a24m clock div enable	0x0
2	reg_apll_frac_div_up	W1T		
3	reg_apll_frac_reg_out_en	R/W	a24m clock output enable	0x0
31:4	Reserved			

apll_frac_div_m

Offset Address: 0x094

Bits	Name	Access	Description	Reset
21:0	reg_apll_frac_div_m	R/W	a24m clock freq is 900*N/M/2 (MHz)	0x0
31:22	Reserved			

apll_frac_div_n

Offset Address: 0x098

Bits	Name	Access	Description	Reset
21:0	reg_apll_frac_div_n	R/W	a24m clock freq is 900*N/M/2 (MHz)	0x0
31:22	Reserved			

mipimpll_clk_csr

Offset Address: 0x0a0

Bits	Name	Access	Description	Reset

Bits	Name	Access	Description	Reset
0	reg_mipimpll_pdiv_pd	R/W	pd post div	0x0
1	reg_mipimpll_d2_pd	R/W	pd div2 div	0x1
2	reg_mipimpll_d3_pd	R/W	pd div3 div	0x1
3	reg_mipimpll_d5_pd	R/W	pd div5 div	0x1
4	reg_mipimpll_d7_pd	R/W	pd div7 div	0x1
7:5	Reserved			
8	reg_mipimpll_pdiv_auto_pd	R/W	auto pd pdiv clk	0x0
9	reg_mipimpll_d2_auto_pd	R/W	auto pd div2 clk	0x1
10	reg_mipimpll_d3_auto_pd	R/W	auto pd div3 clk	0x1
11	reg_mipimpll_d5_auto_pd	R/W	auto pd div5 clk	0x1
12	reg_mipimpll_d7_auto_pd	R/W	auto pd div7 clk	0x1
31:13	Reserved			

a0pll_clk_csr

Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
0	reg_a0pll_pdiv_pd	R/W	pd post div	0x0
1	reg_a0pll_d2_pd	R/W	pd div2 div	0x1
2	reg_a0pll_d3_pd	R/W	pd div3 div	0x1
3	reg_a0pll_d5_pd	R/W	pd div5 div	0x1
4	reg_a0pll_d7_pd	R/W	pd div7 div	0x1
7:5	Reserved			
8	reg_a0pll_pdiv_auto_pd	R/W	auto pd pdiv clk	0x0
9	reg_a0pll_d2_auto_pd	R/W	auto pd div2 clk	0x1
10	reg_a0pll_d3_auto_pd	R/W	auto pd div3 clk	0x1
11	reg_a0pll_d5_auto_pd	R/W	auto pd div5 clk	0x1
12	reg_a0pll_d7_auto_pd	R/W	auto pd div7 clk	0x1
31:13	Reserved			

disppll_clk_csr

Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
0	reg_disppll_pdiv_pd	R/W	pd post div	0x0
1	reg_disppll_d2_pd	R/W	pd div2 div	0x1
2	reg_disppll_d3_pd	R/W	pd div3 div	0x1
3	reg_disppll_d5_pd	R/W	pd div5 div	0x1
4	reg_disppll_d7_pd	R/W	pd div7 div	0x1
7:5	Reserved			
8	reg_disppll_pdiv_auto_pd	R/W	auto pd pdiv clk	0x0
9	reg_disppll_d2_auto_pd	R/W	auto pd div2 clk	0x1
10	reg_disppll_d3_auto_pd	R/W	auto pd div3 clk	0x1
11	reg_disppll_d5_auto_pd	R/W	auto pd div5 clk	0x1
12	reg_disppll_d7_auto_pd	R/W	auto pd div7 clk	0x1
31:13	Reserved			

cam0pll_clk_csr

Offset Address: 0x0ac

Bits	Name	Access	Description	Reset
0	reg_cam0pll_pdiv_pd	R/W	pd post div	0x0
1	reg_cam0pll_d2_pd	R/W	pd div2 div	0x1

Bits	Name	Access	Description	Reset
2	reg_cam0pll_d3_pd	R/W	pd div3 div	0x1
3	reg_cam0pll_d5_pd	R/W	pd div5 div	0x1
4	reg_cam0pll_d7_pd	R/W	pd div7 div	0x1
7:5	Reserved			
8	reg_cam0pll_pdiv_auto_pd	R/W	auto pd pdiv clk	0x0
9	reg_cam0pll_d2_auto_pd	R/W	auto pd div2 clk	0x1
10	reg_cam0pll_d3_auto_pd	R/W	auto pd div3 clk	0x1
11	reg_cam0pll_d5_auto_pd	R/W	auto pd div5 clk	0x1
12	reg_cam0pll_d7_auto_pd	R/W	auto pd div7 clk	0x1
31:13	Reserved			

cam1pll_clk_csr

Offset Address: 0x0b0

Bits	Name	Access	Description	Reset
0	reg_cam1pll_pdiv_pd	R/W	pd post div	0x0
1	reg_cam1pll_d2_pd	R/W	pd div2 div	0x1
2	reg_cam1pll_d3_pd	R/W	pd div3 div	0x1
3	reg_cam1pll_d5_pd	R/W	pd div5 div	0x1
4	reg_cam1pll_d7_pd	R/W	pd div7 div	0x1
7:5	Reserved			
8	reg_cam1pll_pdiv_auto_pd	R/W	auto pd pdiv clk	0x0
9	reg_cam1pll_d2_auto_pd	R/W	auto pd div2 clk	0x1
10	reg_cam1pll_d3_auto_pd	R/W	auto pd div3 clk	0x1
11	reg_cam1pll_d5_auto_pd	R/W	auto pd div5 clk	0x1
12	reg_cam1pll_d7_auto_pd	R/W	auto pd div7 clk	0x1
31:13	Reserved			

clk_cam0_src_div

Offset Address: 0x0c0

Bits	Name	Access	Description	Reset
0	reg_cam0_div_rstn	R/W	[0] Divider Reset Control 0: Assert Reset	0x1
3:1	Reserved			
4	reg_cam0_div_dis	R/W	[4] Divider Reset Control 0: Assert Reset	0x0
7:5	Reserved			
9:8	reg_cam0_src	R/W	[9:8] Clock source 0: cam0pll 1: cam0pll_d2 2: cam0pll_d3 3: mipimpll_d3	0x0
15:10	Reserved			
21:16	reg_cam0_div	R/W	[21:16] Clock Divider Factor	0x20
31:22	Reserved			

clk_cam1_src_div

Offset Address: 0x0c4

Bits	Name	Access	Description	Reset
0	reg_cam1_div_rstn	R/W	[0] Divider Reset Control 0: Assert Reset	0x1

Bits	Name	Access	Description	Reset
3:1	Reserved			
4	reg_cam1_div_dis	R/W	[4] divider disable	0x0
7:5	Reserved			
9:8	reg_cam1_src	R/W	[9:8] Clock source 0: cam0pll 1: cam0pll_d2 2: cam0pll_d3 3: mipimpll_d3	0x0
15:10	Reserved			
21:16	reg_cam1_div	R/W	[21:16] Clock Divider Factor	0x20
31:22	Reserved			

3.2.7.2 PLL_G6 Register Overview

pll_g6_ctrl

Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	mpll_pwd	R/W	pll power down	0x0
3:1	Reserved			
4	tpll_pwd	R/W	pll power down	0x0
7:5	Reserved			
8	fpll_pwd	R/W	pll power down	0x0
31:9	Reserved			

pll_g6_status

Offset Address: 0x004

Bits	Name	Access	Description	Reset
0	updating_mpll_val	RO	pll setting update status	
1	updating_tpll_val	RO	pll setting update status	
2	updating_fpll_val	RO	pll setting update status	
15:3	Reserved			
16	mpll_lock	RO	pll lock status	
17	tpll_lock	RO	pll lock status	
18	fpll_lock	RO	pll lock status	
31:19	Reserved			

mpll_csr

Offset Address: 0x008

Bits	Name	Access	Description	Reset
6:0	mpll_pre_div_sel	R/W	pll pre_div_sel setting	0x0
7	Reserved			
14:8	mpll_post_div_sel	R/W	pll post_div_sel setting	0x0

Bits	Name	Access	Description	Reset
16:15	mpll_sel_mode	R/W	pll mode setting	0x0
23:17	mpll_div_sel	R/W	pll div_sel setting	0x0
26:24	mpll_ictrl	R/W	pll ictrl setting	0x0
31:27	Reserved			

tpll_csr

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
6:0	tpll_pre_div_sel	R/W	pll pre_div_sel setting	0x0
7	Reserved			
14:8	tpll_post_div_sel	R/W	pll post_div_sel setting	0x0
16:15	tpll_sel_mode	R/W	pll mode setting	0x0
23:17	tpll_div_sel	R/W	pll div_sel setting	0x0
26:24	tpll_ictrl	R/W	pll ictrl setting	0x0
31:27	Reserved			

fpll_csr

Offset Address: 0x010

Bits	Name	Access	Description	Reset
6:0	fpll_pre_div_sel	R/W	pll pre_div_sel setting	0x0
7	Reserved			
14:8	fpll_post_div_sel	R/W	pll post_div_sel setting	0x0
16:15	fpll_sel_mode	R/W	pll mode setting	0x0
23:17	fpll_div_sel	R/W	pll div_sel setting	0x0
26:24	fpll_ictrl	R/W	pll ictrl setting	0x0
31:27	Reserved			

pll_g6_ssc_syn_ctrl

Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	reg_fpll_sel_syn_clk	R/W	fpll gen synthesizer clock source 0:750M 1:1.5G	0x1
1	reg_ddr_ssc_syn_src_en	R/W	ddr pll synthesizer clock enable	0x1
2	reg_mpll_ssc_syn_src_en	R/W	mpll synthesizer clock enable	0x0
3	reg_tpll_ssc_syn_src_en	R/W	tpll synthesizer clock enable	0x0
31:4	Reserved			

dpll_ssc_syn_ctrl

Offset Address: 0x050

Bits	Name	Access	Description	Reset
0	reg_dpll_ssc_syn_sw_up	W1T	pll synthesizer software update	
1	reg_dpll_ssc_syn_en_ssc	R/W	pll synthesizer ssc enable	0x0
3:2	reg_dpll_ssc_syn_ssc_mode	R/W		0x0
4	reg_dpll_ssc_syn_bypass	R/W		0x0
5	reg_dpll_ssc_syn_extpulse	R/W		0x0
6	reg_dpll_ssc_syn_fix_div	R/W		0x0
31:7	Reserved			

dpll_ssc_syn_set

Offset Address: 0x054

Bits	Name	Access	Description	Reset
31:0	reg_dpll_ssc_syn_set	R/W	pll synthesizer fraction setting: [31:26] integer 6 bits [26:0] decimal 26 bits	0x0

dpll_ssc_syn_span

Offset Address: 0x058

Bits	Name	Access	Description	Reset
15:0	reg_dpll_ssc_syn_span	R/W		0x0
31:16	Reserved			

dpll_ssc_syn_step

Offset Address: 0x05c

Bits	Name	Access	Description	Reset
23:0	reg_dpll_ssc_syn_step	R/W		0x0
31:24	Reserved			

mpll_ssc_syn_ctrl

Offset Address: 0x060

Bits	Name	Access	Description	Reset
0	reg_mpll_ssc_syn_sw_up	W1T	pll synthesizer software update	
1	reg_mpll_ssc_syn_en_ssc	R/W	pll synthesizer ssc enable	0x0
3:2	reg_mpll_ssc_syn_ssc_mode	R/W		0x0
4	reg_mpll_ssc_syn_bypass	R/W		0x1
5	reg_mpll_ssc_syn_extpulse	R/W		0x0
6	reg_mpll_ssc_syn_fix_div	R/W		0x0
31:7	Reserved			

mpll_ssc_syn_set

Offset Address: 0x064

Bits	Name	Access	Description	Reset
31:0	reg_mpll_ssc_syn_set	R/W	pll synthesizer fraction setting: [31:26] integer 6 bits [26:0] decimal 26 bits	0x0

mpll_ssc_syn_span

Offset Address: 0x068

Bits	Name	Access	Description	Reset
15:0	reg_mpll_ssc_syn_span	R/W		0x0
31:16	Reserved			

mpll_ssc_syn_step

Offset Address: 0x06c

Bits	Name	Access	Description	Reset
23:0	reg_mpll_ssc_syn_step	R/W		0x0

Bits	Name	Access	Description	Reset
31:24	Reserved			

tpll_ssc_syn_ctrl

Offset Address: 0x070

Bits	Name	Access	Description	Reset
0	reg_tpll_ssc_syn_sw_up	W1T	pll synthesizer software update	
1	reg_tpll_ssc_syn_en_ssc	R/W	pll synthesizer ssc enable	0x0
3:2	reg_tpll_ssc_syn_ssc_mode	R/W		0x0
4	reg_tpll_ssc_syn_bypass	R/W		0x1
5	reg_tpll_ssc_syn_extpulse	R/W		0x0
6	reg_tpll_ssc_syn_fix_div	R/W		0x0
31:7	Reserved			

tpll_ssc_syn_set

Offset Address: 0x074

Bits	Name	Access	Description	Reset
31:0	reg_tpll_ssc_syn_set	R/W	pll synthesizer fraction setting: [31:26] integer 6 bits [26:0] decimal 26 bits	0x0

tpll_ssc_syn_span

Offset Address: 0x078

Bits	Name	Access	Description	Reset
15:0	reg_tpll_ssc_syn_span	R/W		0x0
31:16	Reserved			

tpll_ssc_syn_step

Offset Address: 0x07c

Bits	Name	Access	Description	Reset
23:0	reg_tpll_ssc_syn_step	R/W		0x0
31:24	Reserved			

3.2.8 CLK_DIV CRG Register Overview

Clock Gen base address : 0x03002000

Name	Address Offset	Description
clk_en_0	0x000	clock enable register 0
clk_en_1	0x004	clock enable register 1
clk_en_2	0x008	clock enable register 2
clk_en_3	0x00c	clock enable register 3
clk_en_4	0x010	clock enable register 4

Name	Address Offset	Description
clk_sel_0	0x020	clock source selection register 0
clk_byp_0	0x030	clock bypass to xtal register 0
clk_byp_1	0x034	clock bypass to xtal register 1
div_clk_cpu_axi0	0x048	divider register of clk_cpu_axi0
div_clk_tpu	0x054	divider register of clk_tpu
div_clk_sd0	0x070	divider register of clk_sd0
div_clk_100k_sd0	0x078	divider register of clk_100k_sd0
div_clk_sd1	0x07c	divider register of clk_sd1
div_clk_100k_sd1	0x084	divider register of clk_100k_sd1
div_clk_spi_nand	0x088	divider register of clk_spi_nand
div_clk_500m_eth0	0x08c	divider register of clk_500m_eth0
div_clk_gpio_db	0x094	divider register of clk_gpio_db
div_clk_sdma_aud0	0x098	divider register of clk_sdma_aud0
div_clk_sdma_aud1	0x09c	divider register of clk_sdma_aud1
div_clk_sdma_aud2	0x0a0	divider register of clk_sdma_aud2
div_clk_sdma_aud3	0x0a4	divider register of clk_sdma_aud3
div_clk_cam0_200	0x0a8	divider register of clk_cam0_200
div_clk_axi4	0x0b8	divider register of clk_axi4
div_clk_axi6	0x0bc	divider register of clk_axi6
div_clk_dsi_esc	0x0c4	divider register of clk_dsi_esc
div_clk_axi_vip	0x0c8	divider register of clk_axi_vip
div_clk_src_vip_sys_0	0x0d0	divider register of clk_src_vip_sys_0
div_clk_src_vip_sys_1	0x0d8	divider register of clk_src_vip_sys_1
div_clk_disp_src_vip	0x0e0	divider register of clk_disp_src_vip
div_clk_axi_video_codec	0x0e4	divider register of clk_axi_video_codec
div_clk_vc_src0	0x0ec	divider register of clk_vc_src0
div_clk_1m	0x0fc	divider register of clk_1m
div_clk_spi	0x100	divider register of clk_spi
div_clk_i2c	0x104	divider register of clk_i2c
div_clk_src_vip_sys_2	0x110	divider register of clk_src_vip_sys_2
div_clk_audsrc	0x118	divider register of clk_audsrc
div_clk_pwm_src_0	0x120	divider register of clk_pwm_src_0
div_clk_ap_debug	0x128	divider register of clk_ap_debug
div_clk_rtcsys_src_0	0x12c	divider register of clk_rtcsys_src_0
div_clk_c906_0_0	0x130	divider register of clk_c906_0_0
div_clk_c906_0_1	0x134	divider register of clk_c906_0_1
div_clk_c906_1_0	0x138	divider register of clk_c906_1_0
div_clk_c906_1_1	0x13c	divider register of clk_c906_1_1
div_clk_src_vip_sys_3	0x140	divider register of clk_src_vip_sys_3
div_clk_src_vip_sys_4	0x144	divider register of clk_src_vip_sys_4

3.2.9 CLK_DIV CRG Register Overview

clk_en_0

Offset Address: 0x000

Bits	Name	Access	Description	Reset

Bits	Name	Access	Description	Reset
0	Reserved	R/W		0x1
1	clk_en_0_1	R/W	Clock Enable for clk_cpu_axi0 (1: Enable; 0: Gate)	0x1
2	Reserved	R/W		0x1
3	clk_en_0_3	R/W	Clock Enable for clk_xtal_ap (1: Enable; 0: Gate)	0x1
4	clk_en_0_4	R/W	Clock Enable for clk_tpu (1: Enable; 0: Gate)	0x1
5	Reserved			0x1
6	clk_en_0_6	R/W	Clock Enable for clk_ahb_rom (1: Enable; 0: Gate)	0x1
7	clk_en_0_7	R/W	Clock Enable for clk_ddr_axi_reg (1: Enable; 0: Gate)	0x1
8	clk_en_0_8	R/W	Clock Enable for clk_rtc_25m (1: Enable; 0: Gate)	0x1
9	clk_en_0_9	R/W	Clock Enable for clk_tempsen (1: Enable; 0: Gate)	0x1
10	clk_en_0_10	R/W	Clock Enable for clk_saradc (1: Enable; 0: Gate)	0x1
11	clk_en_0_11	R/W	Clock Enable for clk_efuse (1: Enable; 0: Gate)	0x1
12	clk_en_0_12	R/W	Clock Enable for clk_apb_efuse (1: Enable; 0: Gate)	0x1
13	Reserved			
14	clk_en_0_14	R/W	Clock Enable for clk_xtal_misc (1: Enable; 0: Gate)	0x1
15	Reserved	R/W		0x1
16	Reserved	R/W		0x1
17	Reserved	R/W		0x1
18	clk_en_0_18	R/W	Clock Enable for clk_axi4_sd0 (1: Enable; 0: Gate)	0x1
19	clk_en_0_19	R/W	Clock Enable for clk_sd0 (1: Enable; 0: Gate)	0x1
20	clk_en_0_20	R/W	Clock Enable for clk_100k_sd0 (1: Enable; 0: Gate)	0x1
21	clk_en_0_21	R/W	Clock Enable for clk_axi4_sd1 (1: Enable; 0: Gate)	0x1
22	clk_en_0_22	R/W	Clock Enable for clk_sd1 (1: Enable; 0: Gate)	0x1
23	clk_en_0_23	R/W	Clock Enable for clk_100k_sd1 (1: Enable; 0: Gate)	0x1
24	clk_en_0_24	R/W	Clock Enable for clk_spi_nand (1: Enable; 0: Gate)	0x1
25	clk_en_0_25	R/W	Clock Enable for clk_500m_eth0 (1: Enable; 0: Gate)	0x1
26	clk_en_0_26	R/W	Clock Enable for clk_axi4_eth0 (1: Enable; 0: Gate)	0x1
28:27	Reserved			
29	clk_en_0_29	R/W	Clock Enable for clk_apb_gpio (1: Enable; 0: Gate)	0x1
30	clk_en_0_30	R/W	Clock Enable for	0x1

Bits	Name	Access	Description	Reset
			clk_apb_gpio_intr (1: Enable; 0: Gate)	
31	clk_en_0_31	R/W	Clock Enable for clk_gpio_db (1: Enable; 0: Gate)	0x1

clk_en_1

Offset Address: 0x004

Bits	Name	Access	Description	Reset
0	clk_en_1_0	R/W	Clock Enable for clk_ahb_sf (1: Enable; 0: Gate)	0x1
1	clk_en_1_1	R/W	Clock Enable for clk_sdma_axi (1: Enable; 0: Gate)	0x1
2	clk_en_1_2	R/W	Clock Enable for clk_sdma_aud0 (1: Enable; 0: Gate)	0x1
3	clk_en_1_3	R/W	Clock Enable for clk_sdma_aud1 (1: Enable; 0: Gate)	0x1
4	clk_en_1_4	R/W	Clock Enable for clk_sdma_aud2 (1: Enable; 0: Gate)	0x1
5	clk_en_1_5	R/W	Clock Enable for clk_sdma_aud3 (1: Enable; 0: Gate)	0x1
6	clk_en_1_6	R/W	Clock Enable for clk_apb_i2c (1: Enable; 0: Gate)	0x1
7	clk_en_1_7	R/W	Clock Enable for clk_apb_wdt (1: Enable; 0: Gate)	0x1
8	clk_en_1_8	R/W	Clock Enable for clk_apb_pwm (1: Enable; 0: Gate)	0x1
9	clk_en_1_9	R/W	Clock Enable for clk_apb_spi0 (1: Enable; 0: Gate)	0x1
10	clk_en_1_10	R/W	Clock Enable for clk_apb_spi1 (1: Enable; 0: Gate)	0x1
11	clk_en_1_11	R/W	Clock Enable for clk_apb_spi2 (1: Enable; 0: Gate)	0x1
12	clk_en_1_12	R/W	Clock Enable for clk_apb_spi3 (1: Enable; 0: Gate)	0x1
13	clk_en_1_13	R/W	Clock Enable for clk_187p5m (1: Enable; 0: Gate)	0x1
14	clk_en_1_14	R/W	Clock Enable for clk_uart0 (1: Enable; 0: Gate)	0x1
15	clk_en_1_15	R/W	Clock Enable for clk_apb_uart0 (1: Enable; 0: Gate)	0x1
16	clk_en_1_16	R/W	Clock Enable for clk_uart1 (1: Enable; 0: Gate)	0x1
17	clk_en_1_17	R/W	Clock Enable for clk_apb_uart1 (1: Enable; 0: Gate)	0x1
18	clk_en_1_18	R/W	Clock Enable for clk_uart2 (1: Enable; 0: Gate)	0x1
19	clk_en_1_19	R/W	Clock Enable for clk_apb_uart2 (1: Enable; 0: Gate)	0x1
20	clk_en_1_20	R/W	Clock Enable for clk_uart3 (1: Enable; 0: Gate)	0x1
21	clk_en_1_21	R/W	Clock Enable for clk_apb_uart3 (1: Enable; 0: Gate)	0x1
22	clk_en_1_22	R/W	Clock Enable for clk_uart4 (1:	0x1

Bits	Name	Access	Description	Reset
			Enable; 0: Gate)	
23	clk_en_1_23	R/W	Clock Enable for clk_apb_uart4 (1: Enable; 0: Gate)	0x1
24	clk_en_1_24	R/W	Clock Enable for clk_apb_i2s0 (1: Enable; 0: Gate)	0x1
25	clk_en_1_25	R/W	Clock Enable for clk_apb_i2s1 (1: Enable; 0: Gate)	0x1
26	clk_en_1_26	R/W	Clock Enable for clk_apb_i2s2 (1: Enable; 0: Gate)	0x1
27	clk_en_1_27	R/W	Clock Enable for clk_apb_i2s3 (1: Enable; 0: Gate)	0x1
28	clk_en_1_28	R/W	Clock Enable for clk_axi4_usb (1: Enable; 0: Gate)	0x1
29	clk_en_1_29	R/W	Clock Enable for clk_apb_usb (1: Enable; 0: Gate)	0x1
31:30	Reserved			

clk_en_2

Offset Address: 0x008

Bits	Name	Access	Description	Reset
0	Reserved			
1	clk_en_2_1	R/W	Clock Enable for clk_axi4 (1: Enable; 0: Gate)	0x1
2	clk_en_2_2	R/W	Clock Enable for clk_axi6 (1: Enable; 0: Gate)	0x1
3	clk_en_2_3	R/W	Clock Enable for clk_dsi_esc (1: Enable; 0: Gate)	0x1
4	clk_en_2_4	R/W	Clock Enable for clk_axi_vip (1: Enable; 0: Gate)	0x1
5	clk_en_2_5	R/W	Clock Enable for clk_src_vip_sys_0 (1: Enable; 0: Gate)	0x1
6	clk_en_2_6	R/W	Clock Enable for clk_src_vip_sys_1 (1: Enable; 0: Gate)	0x1
7	clk_en_2_7	R/W	Clock Enable for clk_disp_src_vip (1: Enable; 0: Gate)	0x1
8	clk_en_2_8	R/W	Clock Enable for clk_axi_video_codec (1: Enable; 0: Gate)	0x1
9	clk_en_2_9	R/W	Clock Enable for clk_vc_src0 (1: Enable; 0: Gate)	0x1
10	clk_en_2_10	R/W	Clock Enable for clk_h264c (1: Enable; 0: Gate)	0x1
11	clk_en_2_11	R/W	Clock Enable for clk_h265c (1: Enable; 0: Gate)	0x1
12	clk_en_2_12	R/W	Clock Enable for clk_jpeg (1: Enable; 0: Gate)	0x1
13	clk_en_2_13	R/W	Clock Enable for clk_apb_jpeg (1: Enable; 0: Gate)	0x1
14	clk_en_2_14	R/W	Clock Enable for clk_apb_h264c	0x1

Bits	Name	Access	Description	Reset
			(1: Enable; 0: Gate)	
15	clk_en_2_15	R/W	Clock Enable for clk_apb_h265c (1: Enable; 0: Gate)	0x1
16	clk_en_2_16	R/W	Clock Enable for clk_cam0 (1: Enable; 0: Gate)	0x1
17	clk_en_2_17	R/W	Clock Enable for clk_cam1 (1: Enable; 0: Gate)	0x1
18	clk_en_2_18	R/W	Clock Enable for clk_csi_mac0_vip (1: Enable; 0: Gate)	0x1
19	clk_en_2_19	R/W	Clock Enable for clk_csi_mac1_vip (1: Enable; 0: Gate)	0x1
20	clk_en_2_20	R/W	Clock Enable for clk_isp_top_vip (1: Enable; 0: Gate)	0x1
21	clk_en_2_21	R/W	Clock Enable for clk_img_d_vip (1: Enable; 0: Gate)	0x1
22	clk_en_2_22	R/W	Clock Enable for clk_img_v_vip (1: Enable; 0: Gate)	0x1
23	clk_en_2_23	R/W	Clock Enable for clk_sc_top_vip (1: Enable; 0: Gate)	0x1
24	clk_en_2_24	R/W	Clock Enable for clk_sc_d_vip (1: Enable; 0: Gate)	0x1
25	clk_en_2_25	R/W	Clock Enable for clk_sc_v1_vip (1: Enable; 0: Gate)	0x1
26	clk_en_2_26	R/W	Clock Enable for clk_sc_v2_vip (1: Enable; 0: Gate)	0x1
27	clk_en_2_27	R/W	Clock Enable for clk_sc_v3_vip (1: Enable; 0: Gate)	0x1
28	clk_en_2_28	R/W	Clock Enable for clk_ldc_vip (1: Enable; 0: Gate)	0x1
29	clk_en_2_29	R/W	Clock Enable for clk_bt_vip (1: Enable; 0: Gate)	0x1
30	clk_en_2_30	R/W	Clock Enable for clk_disp_vip (1: Enable; 0: Gate)	0x1
31	clk_en_2_31	R/W	Clock Enable for clk_dsi_mac_vip (1: Enable; 0: Gate)	0x1

clk_en_3

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
0	clk_en_3_0	R/W	Clock Enable for clk_lvds0_vip (1: Enable; 0: Gate)	0x1
1	clk_en_3_1	R/W	Clock Enable for clk_lvds1_vip (1: Enable; 0: Gate)	0x1
2	clk_en_3_2	R/W	Clock Enable for clk_csi0_rx_vip (1: Enable; 0: Gate)	0x1
3	clk_en_3_3	R/W	Clock Enable for clk_csi1_rx_vip (1: Enable; 0: Gate)	0x1
4	clk_en_3_4	R/W	Clock Enable for clk_pad_vi_vip (1: Enable; 0: Gate)	0x1
5	clk_en_3_5	R/W	Clock Enable for clk_1m (1: Enable; 0: Gate)	0x1

Bits	Name	Access	Description	Reset
6	clk_en_3_6	R/W	Clock Enable for clk_spi (1: Enable; 0: Gate)	0x1
7	clk_en_3_7	R/W	Clock Enable for clk_i2c (1: Enable; 0: Gate)	0x1
8	clk_en_3_8	R/W	Clock Enable for clk_pm (1: Enable; 0: Gate)	0x1
9	clk_en_3_9	R/W	Clock Enable for clk_timer0 (1: Enable; 0: Gate)	0x1
10	clk_en_3_10	R/W	Clock Enable for clk_timer1 (1: Enable; 0: Gate)	0x1
11	clk_en_3_11	R/W	Clock Enable for clk_timer2 (1: Enable; 0: Gate)	0x1
12	clk_en_3_12	R/W	Clock Enable for clk_timer3 (1: Enable; 0: Gate)	0x1
13	clk_en_3_13	R/W	Clock Enable for clk_timer4 (1: Enable; 0: Gate)	0x1
14	clk_en_3_14	R/W	Clock Enable for clk_timer5 (1: Enable; 0: Gate)	0x1
15	clk_en_3_15	R/W	Clock Enable for clk_timer6 (1: Enable; 0: Gate)	0x1
16	clk_en_3_16	R/W	Clock Enable for clk_timer7 (1: Enable; 0: Gate)	0x1
17	clk_en_3_17	R/W	Clock Enable for clk_apb_i2c0 (1: Enable; 0: Gate)	0x1
18	clk_en_3_18	R/W	Clock Enable for clk_apb_i2c1 (1: Enable; 0: Gate)	0x1
19	clk_en_3_19	R/W	Clock Enable for clk_apb_i2c2 (1: Enable; 0: Gate)	0x1
20	clk_en_3_20	R/W	Clock Enable for clk_apb_i2c3 (1: Enable; 0: Gate)	0x1
21	clk_en_3_21	R/W	Clock Enable for clk_apb_i2c4 (1: Enable; 0: Gate)	0x1
22	clk_en_3_22	R/W	Clock Enable for clk_wgn (1: Enable; 0: Gate)	0x1
23	clk_en_3_23	R/W	Clock Enable for clk_wgn0 (1: Enable; 0: Gate)	0x1
24	clk_en_3_24	R/W	Clock Enable for clk_wgn1 (1: Enable; 0: Gate)	0x1
25	clk_en_3_25	R/W	Clock Enable for clk_wgn2 (1: Enable; 0: Gate)	0x1
26	clk_en_3_26	R/W	Clock Enable for clk_keyscan (1: Enable; 0: Gate)	0x1
27	clk_en_3_27	R/W	Clock Enable for clk_ahb_sf1 (1: Enable; 0: Gate)	0x1
28	Reserved			
29	clk_en_3_29	R/W	Clock Enable for clk_src_vip_sys_2 (1: Enable; 0: Gate	0x1
30	clk_en_3_30	R/W	Clock Enable for clk_pad_vi1_vip (1: Enable; 0: Gate)	0x1
31	clk_en_3_31	R/W	Clock Enable for clk_cfg_reg_vip (1: Enable; 0: Gate)	0x1

clk_en_4

Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	clk_en_4_0	R/W	Clock Enable for clk_cfg_reg_vc (1: Enable; 0: Gate)	0x1
1	clk_en_4_1	R/W	Clock Enable for clk_audsrc (1: Enable; 0: Gate)	0x1
2	clk_en_4_2	R/W	Clock Enable for clk_apb_audsrc (1: Enable; 0: Gate)	0x1
3	Reserved			
4	clk_en_4_4	R/W	Clock Enable for clk_pwm_src (1: Enable; 0: Gate)	0x1
5	clk_en_4_5	R/W	Clock Enable for clk_ap_debug(1: Enable; 0: Gate)	0x1
6	clk_en_4_6	R/W	Clock Enable for clk_rtcsys_src_0 (1: Enable; 0: Gate)	0x1
7	clk_en_4_7	R/W	Clock Enable for clk_pad_vi2_vip (1: Enable; 0: Gate)	0x1
8	clk_en_4_8	R/W	Clock Enable for clk_csi_be_vip (1: Enable; 0: Gate)	0x1
9	clk_en_4_9	R/W	Clock Enable for clk_vip_ip0_en	0x1
10	clk_en_4_10	R/W	Clock Enable for clk_vip_ip1_en	0x1
11	clk_en_4_11	R/W	Clock Enable for clk_vip_ip2_en	0x1
12	clk_en_4_12	R/W	Clock Enable for clk_vip_ip3_en	0x1
13	clk_en_4_13	R/W	Clock Enable for clk_c906_0_en	0x1
14	clk_en_4_14	R/W	Clock Enable for clk_c906_1_en	0x1
15	clk_en_4_15	R/W	Clock Enable for clk_src_vip_sys_3_en	0x1
16	clk_en_4_16	R/W	Clock Enable for clk_src_vip_sys_4_en	0x1
17	clk_en_4_17	R/W	Clock Enable for clk_ive_vip_en	0x1
18	clk_en_4_18	R/W	Clock Enable for clk_raw_vip_en	0x1
19	clk_en_4_19	R/W	Clock Enable for clk_osdc_vip_en	0x1
20	clk_en_4_20	R/W	Clock Enable for clk_fbc_vip_en	0x1
21	clk_en_4_21	R/W	Clock Enable for clk_cam0_vip_en	0x1
31:22	Reserved			

clk_sel_0

Offset Address: 0x020

Bits	Name	Access	Description	Reset
22:0	Reserved			
23	clk_sel_0_23	R/W	Clock Select for C906's clock clk_c906_0 1: Select div_clk_c906_0_0 as clock source 0: Select div_clk_c906_0_1 as clock source	0x0
24	clk_sel_0_24	R/W	Clock Select for C906's clock clk_c906_1 1: Select div_clk_c906_1_0 as clock source 0: Select div_clk_c906_1_1 as	0x0

Bits	Name	Access	Description	Reset
			clock source	
31:25	Reserved			

clk_byp_0

Offset Address: 0x030

Bits	Name	Access	Description	Reset
0	Reserved			0x1
1	clk_byp_0_1	R/W	Clock Bypass to xtal for clock clk_cpu_axi0	0x1
2	Reserved	R/W		0x1
3	clk_byp_0_3	R/W	Clock Bypass to xtal for TPU's clock clk_tpu	0x1
4	Reserved			0x1
5	Reserved			0x1
6	clk_byp_0_6	R/W	Clock Bypass to xtal for SD's clock clk_sd0	0x1
7	clk_byp_0_7	R/W	Clock Bypass to xtal for SD's clock clk_sd1	0x1
8	clk_byp_0_8	R/W	Clock Bypass to xtal for SPI_NAND's clock clk_spi_nand	0x1
9	clk_byp_0_9	R/W	Clock Bypass to xtal for ETH0's clock clk_500m_eth0	0x1
10	Reserved			
11	clk_byp_0_11	R/W	Clock Bypass to xtal for AUDIO's clock clk_aud0	0x1
12	clk_byp_0_12	R/W	Clock Bypass to xtal for AUDIO's clock clk_aud1	0x1
13	clk_byp_0_13	R/W	Clock Bypass to xtal for AUDIO's clock clk_aud2	0x1
14	clk_byp_0_14	R/W	Clock Bypass to xtal for AUDIO's clock clk_aud3	0x1
15	clk_byp_0_15	R/W	Clock Bypass to xtal for PWM's clock clk_pwm_src	0x1
16	clk_byp_0_16	R/W	Clock Bypass to xtal for TOP's clock clk_cam0_200	0x1
18:17	Reserved			
19	clk_byp_0_19	R/W	Clock Bypass to xtal for FABRIC_AXI4's clock clk_axi4	0x1
20	clk_byp_0_20	R/W	Clock Bypass to xtal for FABRIC_AXI6's clock clk_axi6	0x1
21	clk_byp_0_21	R/W	Clock Bypass to xtal for VIP_SYS's clock clk_dsi_esc	0x1
22	clk_byp_0_22	R/W	Clock Bypass to xtal for VIP_SYS's clock clk_axi_vip	0x1
23	clk_byp_0_23	R/W	Clock Bypass to xtal for VIP_SYS's clock clk_src_vip_sys_0	0x1
24	clk_byp_0_24	R/W	Clock Bypass to xtal for VIP_SYS's clock clk_src_vip_sys_1	0x1
25	clk_byp_0_25	R/W	Clock Bypass to xtal for	0x1

Bits	Name	Access	Description	Reset
			VIP_SYS's clock clk_disp_src_vip	
26	clk_byp_0_26	R/W	Clock Bypass to xtal for Video_subsys's clock clk_axi_video_codec	0x1
27	clk_byp_0_27	R/W	Clock Bypass to xtal for Video_subsys's clock clk_vc_src0	0x1
29:28	Reserved			
30	clk_byp_0_30	R/W	Clock Bypass to xtal for SPI's clock clk_spi	0x1
31	clk_byp_0_31	R/W	Clock Bypass to xtal for IIC's clock clk_i2c	0x1

clk_byp_1

Offset Address: 0x034

Bits	Name	Access	Description	Reset
0	Reserved			
1	clk_byp_1_1	R/W	Clock Bypass to xtal for VIP_SYS's clock clk_src_vip_sys_2	0x1
2	clk_byp_1_2	R/W	Clock Bypass to xtal for AUDSRC's clock clk_audsrc	0x1
3	clk_byp_1_3	R/W	Clock Bypass to xtal for Video_subsys's clock clk_vc_src2	0x1
4	clk_byp_1_4	R/W	Clock Bypass to xtal for clk_ap_debug	0x1
5	clk_byp_1_5	R/W	Clock Bypass to xtal for clk_src_rtc_sys_0	0x1
6	clk_byp_1_6	R/W	Clock Bypass to xtal for c906_0	0x1
7	clk_byp_1_7	R/W	Clock Bypass to xtal for c906_1	0x1
8	clk_byp_1_8	R/W	Clock Bypass to xtal for VIP_SYS's clock clk_src_vip_sys_3	0x1
9	clk_byp_1_9	R/W	Clock Bypass to xtal for VIP_SYS's clock clk_src_vip_sys_4	0x1
31:10	Reserved			

div_clk_cpu_axi0

Offset Address: 0x048

Bits	Name	Access	Description	Reset
31:0	div_clk_cpu_axi0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : fpll 1 : disppll	0x00000001

div_clk_tpu

Offset Address: 0x054

Bits	Name	Access	Description	Reset
31:0	div_clk_tpu	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : tpll 1 : apll 2 : mipimpll 3 : fpll	0x00000301

div_clk_sd0

Offset Address: 0x070

Bits	Name	Access	Description	Reset
31:0	div_clk_sd0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : fpll 1 : disppll	0x00000001

div_clk_100k_sd0

Offset Address: 0x078

Bits	Name	Access	Description	Reset
31:0	div_clk_100k_sd0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x00000001

div_clk_sd1

Offset Address: 0x07c

Bits	Name	Access	Description	Reset
31:0	div_clk_sd1	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : fpll 1 : disppll	0x00000001

div_clk_100k_sd1

Offset Address: 0x084

Bits	Name	Access	Description	Reset
31:0	div_clk_100k_sd1	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x00000001

div_clk_spi_nand

Offset Address: 0x088

Bits	Name	Access	Description	Reset
31:0	div_clk_spi_nand	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : fpll 1 : disppll	0x00000001

div_clk_500m_eth0

Offset Address: 0x08c

Bits	Name	Access	Description	Reset
31:0	div_clk_500m_eth0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x00000001

div_clk_gpio_db

Offset Address: 0x094

Bits	Name	Access	Description	Reset
31:0	div_clk_gpio_db	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x00000001

div_clk_sdma_aud0

Offset Address: 0x098

Bits	Name	Access	Description	Reset
31:0	div_clk_sdma_aud0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from	0x00000001

Bits	Name	Access	Description	Reset
			Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : apll 1 : a24k	

div_clk_sdma_aud1

Offset Address: 0x09c

Bits	Name	Access	Description	Reset
31:0	div_clk_sdma_aud1	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : apll 1 : a24k	0x00000001

div_clk_sdma_aud2

Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
31:0	div_clk_sdma_aud2	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : apll 1 : a24k	0x00000001

div_clk_sdma_aud3

Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
31:0	div_clk_sdma_aud3	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : apll 1 : a24k	0x00000001

div_clk_cam0_200

Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
31:0	div_clk_cam0_200	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : xtal 1 : disppll	0x00000001

div_clk_axi4

Offset Address: 0x0b8

Bits	Name	Access	Description	Reset
31:0	div_clk_axi4	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : fpll 1 : disppll	0x00000001

div_clk_axi6

Offset Address: 0x0bc

Bits	Name	Access	Description	Reset
31:0	div_clk_axi6	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x00000001

div_clk_dsi_esc

Offset Address: 0x0c4

Bits	Name	Access	Description	Reset
31:0	div_clk_dsi_esc	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x00000001

div_clk_axi_vip

Offset Address: 0x0c8

Bits	Name	Access	Description	Reset
31:0	div_clk_axi_vip	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset	0x00000001

Bits	Name	Access	Description	Reset
			[3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : mipimpll 1 : cam0pll 2 : disppll 3 : fpll	

div_clk_src_vip_sys_0

Offset Address: 0x0d0

Bits	Name	Access	Description	Reset
31:0	div_clk_src_vip_sys_0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : mipimpll 1 : cam0pll 2 : disppll 3 : fpll	0x00000301

div_clk_src_vip_sys_1

Offset Address: 0x0d8

Bits	Name	Access	Description	Reset
31:0	div_clk_src_vip_sys_1	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : mipimpll 1 : cam0pll 2 : disppll 3 : fpll	0x00000301

div_clk_disp_src_vip

Offset Address: 0x0e0

Bits	Name	Access	Description	Reset
31:0	div_clk_disp_src_vip	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x00000001

div_clk_axi_video_codec

Offset Address: 0x0e4

Bits	Name	Access	Description	Reset
31:0	div_clk_axi_video_codec	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : apll 1 : mipimpll 2 : camlpll 3 : fpll	0x00000101

div_clk_vc_src0

Offset Address: 0x0ec

Bits	Name	Access	Description	Reset
31:0	div_clk_vc_src0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : apll 1 : mipimpll 2 : camlpll 3 : fpll	0x00000101

div_clk_1m

Offset Address: 0x0fc

Bits	Name	Access	Description	Reset
31:0	div_clk_1m	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x00000001

div_clk_spi

Offset Address: 0x100

Bits	Name	Access	Description	Reset
31:0	div_clk_spi	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x00000001

div_clk_i2c

Offset Address: 0x104

Bits	Name	Access	Description	Reset
31:0	div_clk_i2c	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x00000001

div_clk_src_vip_sys_2

Offset Address: 0x110

Bits	Name	Access	Description	Reset
31:0	div_clk_src_vip_sys_2	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : mipimpll 1 : cam0pll 2 : dispplll 3 : fpll	0x00000201

div_clk_audsrc

Offset Address: 0x118

Bits	Name	Access	Description	Reset
31:0	div_clk_audsrc	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : apll 1 : a24k	0x00000001

div_clk_pwm_src_0

Offset Address: 0x120

Bits	Name	Access	Description	Reset
31:0	div_clk_pwm_src	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : fpll 1 : dispplll	0x00000001

div_clk_ap_debug

Offset Address: 0x128

Bits	Name	Access	Description	Reset
31:0	div_clk_ap_debug	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x00000001

div_clk_rtcsys_src_0

Offset Address: 0x12c

Bits	Name	Access	Description	Reset
31:0	div_clk_src_rtc_sys_0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x00000001

div_clk_c906_0_0

Offset Address: 0x130

Bits	Name	Access	Description	Reset
31:0	div_clk_c906_0_0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [1] High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider [2] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [3] Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : tpll 1 : apll 2 : mipimpll 3 : mpll	0x00000201

div_clk_c906_0_1

Offset Address: 0x134

Bits	Name	Access	Description	Reset
31:0	div_clk_c906_0_1	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from	0x00000001

Bits	Name	Access	Description	Reset
			this register [20:16] Clock Divider Factor [9:8] clk_src 0 : fpll	

div_clk_c906_1_0

Offset Address: 0x138

Bits	Name	Access	Description	Reset
31:0	div_clk_c906_1_0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [1] High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider [2] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [3] Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : tpll 1 : apll 2 : mipimpll 3 : mpll	0x00000001

div_clk_c906_1_1

Offset Address: 0x13c

Bits	Name	Access	Description	Reset
31:0	div_clk_c906_1_1	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : fpll	0x00000001

div_clk_src_vip_sys_3

Offset Address: 0x140

Bits	Name	Access	Description	Reset
31:0	div_clk_src_vip_sys_3	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src	0x00000001

Bits	Name	Access	Description	Reset
			0 : mipimpll 1 : cam0pll 2 : disppll 3 : fpll	

div_clk_src_vip_sys_4

Offset Address: 0x144

Bits	Name	Access	Description	Reset
31:0	div_clk_src_vip_sys_4	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : mipimpll 1 : cam0pll 2 : disppll 3 : fpll	0x00000201

3.3 Processor Subsystem

The chip adopts RISCV C906 processor, which has the following characteristics.

- The maximum operating frequency of the processor can reach 1.0 GHz
- Integrated with a vector execution unit and a floating-point coprocessor.
- Integrate L1 Cache which includes 32KB Instruction Cache and 64KB Data Cache
- Support MMU（Memory Management Unit）
- Integrated interrupt controller inside the processor.
- Support JTAG debugging interface

The coprocessor RISCV C906 @ 700Mhz

- Integrated with a floating-point unit (FPU).

3.4 Interrupt System

The chip' s interrupt source is shown in the following table.

Table 3- 5 Interrupt Number and Interrupt Source Mapping Table

中断号	中断源	中断号	中断源	中断号	中断源
16	TEMPSENS 中断	48	UART4 中断	80	Timer1 中断
17	RTC Alarm 中断	49	I2C0 中断	81	Timer2 中断
18	RTC Longpress 中断	50	I2C1 中断	82	Timer3 中断
19	VBAT DET 中断	51	I2C2 中断	83	Timer4 中断
20	JPEG 中断	52	I2C3 中断	84	Timer5 中断
21	H264 中断	53	I2C4 中断	85	Timer6 中断
22	H265 中断	54	SPI1 中断	86	Timer7 中断
23	VC SBM 中断	55	SPI2 中断	87	peri_firewall 中断
24	ISP 中断	56	SPI3 中断	88	hsperi_firewall 中断
25	SC_TOP 中断	57	SPI4 中断	89	ddr_fw 中断
26	CSI_MAC0 中断	58	Watch Dog1 中断	90	rom_firewall 中断
27	CSI_MAC1 中断	59	KEYSCAN 中断	91	SPACC 中断
28	LDC 中断	60	GPIO0 中断	92	TRNG 中断
29	System DMA 中断	61	GPIO1 中断	93	ddr_axi_mon 中断
30	USB 中断	62	GPIO2 中断	94	ddr_pi_phy 中断
31	Ethnet0 中断	63	GPIO3 中断	95	SPI_NOR 中断
32	Ethnet0 中断	64	Wiegand0 中断	96	EPHY 中断
33	保留	65	Wiegand1 中断	97	IVE 中断
34	保留	66	Wiegand2 中断	98	保留
35	SD0 Wakup 中断	67	RTC MBOX 中断	99	保留
36	SD0 中断	68		100	SARADC 中断
37	SD1 Wakup 中断	69	RTC IRRX 中断	101	mbox 中断
38	SD1 中断	70	RTC GPIO 中断		
39	SPI_NAND 中断	71	RTC UART 中断		
40	I2S0 中断	72	RTC SPI_NOR 中断		
41	I2S1 中断	73	RTC I2C 中断		
42	I2S2 中断	74	RTC WDG 中断		
43	I2S3 中断	75	TPU 中断		
44	UART0 中断	76	TDMA 中断		

中断号	中断源	中断号	中断源	中断号	中断源
45	UART1 中断	77	保留		
46	UART2 中断	78	保留		
47	UART3 中断	79	Timer0 中断		

3.5 System Controller

3.5.1 Overview

The system controller controls the chip through registers, including system soft reset, clock control and so on. Reset and clock have been described in other chapters. This chapter describes the configuration and status registers of some other system function modules.

3.5.2 Function Description

3.5.2.1 Global Reset Enablement

System global soft reset, debug reset and watch dog reset could be issued by configuring reg_sw_root_reset_en register. Details are explained in reg_sw_root_reset_en.

3.5.2.2 System DMA Channel Mapping

There are 8 channels in this DMA, and 0 ~7 dma request interfaces are configured respectively. The dma requests interfaces from 0 to 7 could be mapped to one of the peripheral interfaces in the following table by the system control registers sdma_dma_ch_remap0 and sdma_dma_ch_remap1. A peripheral interface should not be assigned to multiple channels.
Configuration steps:

Configure DMA channel image register sdma_dma_ch_remap0, sdma_dma_ch_remap1. Then write 1 to update_dma_remap_0_3 and update_dma_remap_4_7 to make the mapping effective.

编号	DMA 界面	编号	DMA 界面
0	dma_rx_req_i2s0	24	dma_rx_req_i2c0
1	dma_tx_req_i2s0	25	dma_tx_req_i2c0
2	dma_rx_req_i2s1	26	dma_rx_req_i2c1
3	dma_tx_req_i2s1	27	dma_tx_req_i2c1
4	dma_rx_req_i2s2	28	dma_rx_req_i2c2
5	dma_tx_req_i2s2	29	dma_tx_req_i2c2
6	dma_rx_req_i2s3	30	dma_rx_req_i2c3
7	dma_tx_req_i2s3	31	dma_tx_req_i2c3
8	dma_rx_req_n_uart0	32	dma_rx_req_i2c4
9	dma_tx_req_n_uart0	33	dma_tx_req_i2c4
10	dma_rx_req_n_uart1	34	dma_rx_req_tdm0
11	dma_tx_req_n_uart1	35	dma_tx_req_tdm0
12	dma_rx_req_n_uart2	36	dma_rx_req_tdm1
13	dma_tx_req_n_uart2	37	dma_req_audsrc
14	dma_rx_req_n_uart3	38	dma_req_spi_nand
15	dma_tx_req_n_uart3	39	dma_req_spi_nor
16	dma_rx_req_spi0	40	dma_rx_req_n_uart4
17	dma_tx_req_spi0	41	dma_tx_req_n_uart4
18	dma_rx_req_spi1	42	dma_req_spi_nor1
19	dma_tx_req_spi1		
20	dma_rx_req_spi2		
21	dma_tx_req_spi2		
22	dma_rx_req_spi3		
23	dma_tx_req_spi3		

3.5.2.3 DDR AXI Urgent/Qos Configuration

The DDR AXI priority could be controlled by ddr_axi_urgent_o, ddr_axi_urgen, ddr_axi_qos_0 and ddr_axi_qos_1. Refer to the DDR controller section for details.

3.5.3 System Control Register

3.5.3.1 System Control Register Overview

Bass address 0x03000000

Name	Address Offset	Description
conf_info	0x004	conf_info
sys_ctrl_reg	0x008	sys_ctrl_reg
usb_phy_ctrl_reg	0x048	usb_phy_ctrl_reg
sdma_dma_ch_remap0	0x154	sdma_dma_ch_remap0
sdma_dma_ch_remap1	0x158	sdma_dma_ch_remap1
top_timer_clk_sel	0x1a0	top_timer_clk_sel
top_wdt_ctrl	0x1a8	top_timer_clk_sel
ddr_axi_urgent_ow	0x1b8	ddr_axi_urgent_ow
ddr_axi_urgent	0x1bc	ddr_axi_urgent
ddr_axi_qos_0	0x1d8	ddr_axi_qos_0
ddr_axi_qos_1	0x1dc	ddr_axi_qos_1
sd_pwrsw_ctrl	0x1f4	sd_pwrsw_ctrl
sd_pwrsw_time	0x1f8	sd_pwrsw_time
ddr_axi_qos_ow	0x23c	ddr_axi_qos_ow
sd_ctrl_opt	0x294	additional control register for sd
sdma_dma_int_mux	0x298	Mux sdma channel interrupt to different processors

3.5.3.2 System Control Register Overview

conf_info

Offset Address: 0x004

Bits	Name	Access	Description	Reset
7:0	boot_sel	RO	[2:0] boot device selection 0: SPI_NAND 1: reserved 2: SPI_NOR 3: reserved [7:3] : resreved	
8	io_sta_usbid	RO	IO status from USBID PAD	
9	io_sta_usbvbus	RO	IO status from USB_VBUS_DET PAD	
23:10	Reserved			
31:24	io_sta_trap	RO	io_sta_trap[0] : io_boot_rom_din io_sta_trap[1] : io_boot_dev0_din io_sta_trap[2] : io_boot_dev1_din io_sta_trap[3] : io_trap_sd0_pwr_din io_sta_trap[4] : io_pkg_type0_din io_sta_trap[5] : io_pkg_type1_din io_sta_trap[6] : io_pkg_type2_din io_sta_trap[7] : io_trap_zq_din	

sys_ctrl_reg

Offset Address: 0x008

Bits	Name	Access	Description	Reset
1:0	Reserved			
5:2	reg_sw_root_reset_en	R/W	bit0 : wdt reset enable bit1 : cdbgrstreq enable	0x0

Bits	Name	Access	Description	Reset
			bit2 : reserved bit3 : reg_soft_reset_x_system enable	
31:6	Reserved			

usb_phy_ctrl_reg

Offset Address: 0x048

Bits	Name	Access	Description	Reset
0	reg_usb_phy_external_vbusvalid	R/W	external vbus status	0x0
1	reg_usb_drive_vbus	R/W	drive vbus power	0x0
4:2	Reserved			
5	toreg_usb_id_en	RO	usb id pullup status	
6	reg_usb_phy_idpad_c_ow	R/W	usb id overwrite enable	0x0
7	reg_usb_phy_idpad_c_sw	R/W	usb id overwrite value	0x0
8	io_usb_phy_idpad_c	RO	usb id external IO pin status	
9	toreg_usb_phy_idpad_c	RO	usb id pin status	
31:10	Reserved			

sdma_dma_ch_remap0

Offset Address: 0x154

Bits	Name	Access	Description	Reset
5:0	reg_dma_remap_ch0	R/W	dma channel 0 mapping	0x0
7:6	Reserved			
13:8	reg_dma_remap_ch1	R/W	dma channel 1 mapping	0x0
15:14	Reserved			
21:16	reg_dma_remap_ch2	R/W	dma channel 2 mapping	0x0
23:22	Reserved			
29:24	reg_dma_remap_ch3	R/W	dma channel 3 mapping	0x0
30	Reserved			
31	update_dma_remp_0_3	W1T	write 1 to update dma channel0~3 mapping	

sdma_dma_ch_remap1

Offset Address: 0x158

Bits	Name	Access	Description	Reset
5:0	reg_dma_remap_ch4	R/W	dma channel 4 mapping	0x0
7:6	Reserved			
13:8	reg_dma_remap_ch5	R/W	dma channel 5 mapping	0x0
15:14	Reserved			
21:16	reg_dma_remap_ch6	R/W	dma channel 6 mapping	0x0
23:22	Reserved			
29:24	reg_dma_remap_ch7	R/W	dma channel 7 mapping	0x0
30	Reserved			
31	update_dma_remp_4_7	W1T	write 1 to update dma channel4~7 mapping	

top_timer_clk_sel

Offset Address: 0x1a0

Bits	Name	Access	Description	Reset
7:0	reg_timer_clk_sel	R/W	timer0~7 clock selection. 0: xtal	0x0

Bits	Name	Access	Description	Reset
			clock,1:32k clock	
31:8	Reserved			

top_wdt_ctrl

Offset Address: 0x1a8

Bits	Name	Access	Description	Reset
2:0	reg_wdt_rst_sys_en	R/W	enable wdt0~wdt2 to reset system	0x7
3	Reserved			
6:4	reg_wdt_rst_cpu_en	R/W	enable wdt0~wdt2 to reset cpu	0x0
7	Reserved			
10:8	reg_wdt_clk_sel	R/W	top_wdt clock selection. 0: xtal clock,1:32k clock	0x0
31:11	Reserved			

ddr_axi_urgent_ow

Offset Address: 0x1b8

Bits	Name	Access	Description	Reset
0	reg_awurgent_m1_ow	R/W	ddr axi port1 awurgent overwrite enable	0x1
1	reg_arurgent_m1_ow	R/W	ddr axi port1 arurgent overwrite enable	0x1
2	reg_awurgent_m2_ow	R/W	ddr axi port2 awurgent overwrite enable	0x1
3	reg_arurgent_m2_ow	R/W	ddr axi port2 arurgent overwrite enable	0x1
4	reg_awurgent_m3_ow	R/W	ddr axi port3 awurgent overwrite enable	0x1
5	reg_arurgent_m3_ow	R/W	ddr axi port3 arurgent overwrite enable	0x1
6	reg_awurgent_m4_ow	R/W	ddr axi port4 awurgent overwrite enable	0x1
7	reg_arurgent_m4_ow	R/W	ddr axi port4 arurgent overwrite enable	0x1
8	reg_awurgent_m5_ow	R/W	ddr axi port5 awurgent overwrite enable	0x1
9	reg_arurgent_m5_ow	R/W	ddr axi port5 arurgent overwrite enable	0x1
10	reg_awurgent_m6_ow	R/W	ddr axi port6 awurgent overwrite enable	0x1
11	reg_arurgent_m6_ow	R/W	ddr axi port6 arurgent overwrite enable	0x1
31:12	Reserved			

ddr_axi_urgent

Offset Address: 0x1bc

Bits	Name	Access	Description	Reset
0	reg_awurgent_m1	R/W	ddr axi port1 awurgent overwrite value	0x0
1	reg_arurgent_m1	R/W	ddr axi port1 arurgent overwrite value	0x0
2	reg_awurgent_m2	R/W	ddr axi port2 awurgent overwrite value	0x0
3	reg_arurgent_m2	R/W	ddr axi port2 arurgent overwrite value	0x0
4	reg_awurgent_m3	R/W	ddr axi port3 awurgent overwrite value	0x0
5	reg_arurgent_m3	R/W	ddr axi port3 arurgent overwrite value	0x0
6	reg_awurgent_m4	R/W	ddr axi port4 awurgent overwrite value	0x0
7	reg_arurgent_m4	R/W	ddr axi port4 arurgent overwrite value	0x0
8	reg_awurgent_m5	R/W	ddr axi port5 awurgent overwrite value	0x0
9	reg_arurgent_m5	R/W	ddr axi port5 arurgent overwrite value	0x0
10	reg_awurgent_m6	R/W	ddr axi port6 awurgent overwrite value	0x0
11	reg_arurgent_m6	R/W	ddr axi port6 arurgent overwrite value	0x0

Bits	Name	Access	Description	Reset
31:12	Reserved			

ddr_axi_qos_0

Offset Address: 0x1d8

Bits	Name	Access	Description	Reset
3:0	reg_awqos_m1	R/W	ddr axi port1 awqos setting	0x0
7:4	reg_arqos_m1	R/W	ddr axi port1 arqos setting	0x0
11:8	reg_awqos_m2	R/W	ddr axi port2 awqos setting	0x0
15:12	reg_arqos_m2	R/W	ddr axi port2 arqos setting	0x0
19:16	reg_awqos_m3	R/W	ddr axi port3 awqos setting	0x0
23:20	reg_arqos_m3	R/W	ddr axi port3 arqos setting	0x0
27:24	reg_awqos_m4	R/W	ddr axi port4 awqos setting	0x0
31:28	reg_arqos_m4	R/W	ddr axi port4 arqos setting	0x0

ddr_axi_qos_1

Offset Address: 0x1dc

Bits	Name	Access	Description	Reset
3:0	reg_awqos_m5	R/W	ddr axi port5 awqos setting	0x0
7:4	reg_arqos_m5	R/W	ddr axi port5 arqos setting	0x0
11:8	reg_awqos_m6	R/W	ddr axi port6 awqos setting	0x0
15:12	reg_arqos_m6	R/W	ddr axi port6 arqos setting	0x0
31:16	Reserved			

sd_pwrsw_ctrl

Offset Address: 0x1f4

Bits	Name	Access	Description	Reset
0	reg_en_pwrsw	R/W	18/33 IO power switch enable	0x0
1	reg_pwrsw_vsel	R/W	18/33 IO power switch enable 0: 3.3v 1: 1.8v	0x1
2	reg_pwrsw_disc	R/W	18/33 IO power switch discharge enable	0x0
3	reg_pwrsw_auto	R/W	18/33 IO power switch auto protect enable	0x1
31:4	Reserved			

sd_pwrsw_time

Offset Address: 0x1f8

Bits	Name	Access	Description	Reset
15:0	reg_tpwrup	R/W	18/33 IO power switch, power up protection time is 500x40ns = 20us	0x1f4
31:16	reg_tpwrdn	R/W	18/33 IO power switch, power down protection time is 500x40ns = 20us	0x1f4

ddr_axi_qos_ow

Offset Address: 0x23c

Bits	Name	Access	Description	Reset
0	reg_awqos_m1_ow	R/W	ddr axi port1 awqos overwrite enable	0x1
1	reg_arqos_m1_ow	R/W	ddr axi port1 arqos overwrite enable	0x1
2	reg_awqos_m2_ow	R/W	ddr axi port2 awqos overwrite enable	0x1
3	reg_arqos_m2_ow	R/W	ddr axi port2 arqos overwrite enable	0x1
4	reg_awqos_m3_ow	R/W	ddr axi port3 awqos overwrite enable	0x1
5	reg_arqos_m3_ow	R/W	ddr axi port3 arqos overwrite enable	0x1
6	reg_awqos_m4_ow	R/W	ddr axi port4 awqos overwrite enable	0x1

Bits	Name	Access	Description	Reset
7	reg_arqos_m4_ow	R/W	ddr axi port4 arqos overwrite enable	0x1
8	reg_awqos_m5_ow	R/W	ddr axi port5 awqos overwrite enable	0x1
9	reg_arqos_m5_ow	R/W	ddr axi port5 arqos overwrite enable	0x1
10	reg_awqos_m6_ow	R/W	ddr axi port6 awqos overwrite enable	0x1
11	reg_arqos_m6_ow	R/W	ddr axi port6 arqos overwrite enable	0x1
31:12	Reserved			

sd_ctrl_opt

Offset Address: 0x294

Bits	Name	Access	Description	Reset
0	reg_sd0_carddet_ow	R/W	sd0 card detect over write enable	0x0
1	reg_sd0_carddet_sw	R/W	sd0 card detect over write value	0x0
7:2	Reserved			
8	reg_sd1_carddet_ow	R/W	sd1 card detect over write enable	0x0
9	reg_sd1_carddet_sw	R/W	sd1 card detect over write value	0x0
15:10	Reserved			
16	reg_sd0_pwr_en_polarity	R/W	off chip sd0 pwr en polarity 0: SD_LDO power ctrl high is power on , low is power off 1: SD_LDO power ctrl high is power off , low is power on	0x0
31:17	Reserved			

sdma_dma_int_mux

Offset Address: 0x298

Bits	Name	Access	Description	Reset
8:0	reg_dma_int_mux_cpu0	R/W	This register is used to mux separate sdma channel interrupts to CPU0. These are enable bits corresponding to {intr_cmnreg,intr_ch[7:0]}	0x1FF
9	Reserved			
18:10	reg_dma_int_mux_cpu1	R/W	This register is used to mux separate sdma channel interrupts to CPU1. These are enable bits corresponding to {intr_cmnreg,intr_ch[7:0]}	0x0
19	Reserved			
28:20	reg_dma_int_mux_cpu2	R/W	This register is used to mux separate sdma channel interrupts to CPU2. These are enable bits corresponding to {intr_cmnreg,intr_ch[7:0]}	0x0
31:29	Reserved			

3.6 DMA Controller

3.6.1 Overview

DMA (Direct Memory Access) can transfer data directly between the memory and the device. This mechanism can greatly reduce CPU access time and improve data transmission rate. It is very suitable for big data transmission. When the chip works, it often needs multi-channel data transmission. Each channel needs a DMA hardware to support, and the DMAC (DMA controller) is responsible for the control of multi-channel. The following figure shows the DMAC hardware control flow. The source and destination could be from different AXI Buses.

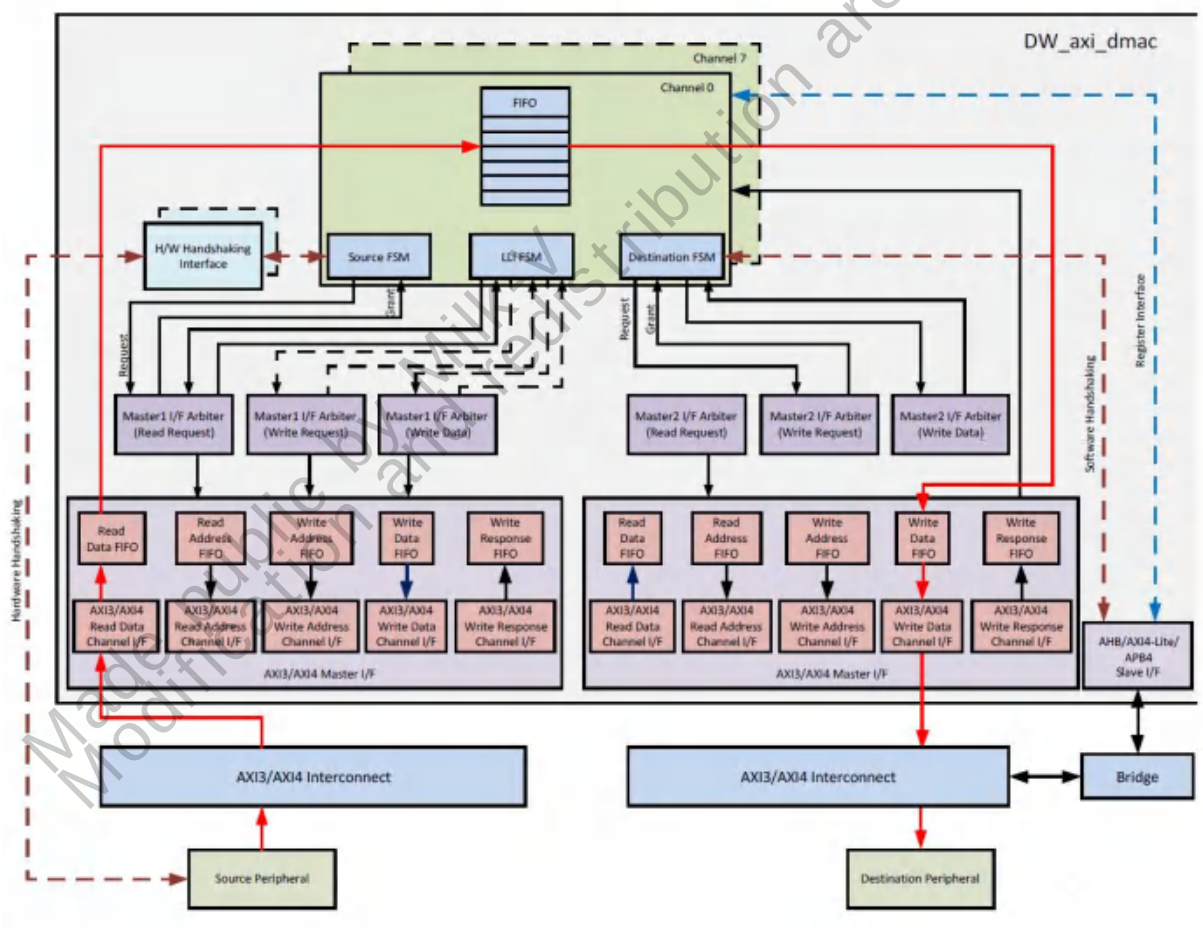


Figure 3- 4 DMAC Hardware Control Fow Diagram

3.6.2 Characteristics

The characteristics of DMAC are as follows.

a. Up to 8 DMA channels can be established at the same time.
b. The data source and data destination can be set as memory or device.
c. Only one way transport configuration is allowed.
d. Provide DMA transmission pause, resume, and cancellation.。
e. Support DMA Burst length configuration.
f. Provide DMA channel priority configuration.
g. When channel data is transmitted between devices, flow control can be controlled by devices.
h. Support hardware linked list function.
i. Channel locking is supported. Other channel requests will be ignored before channel locking is completed.

3.6.3 Function Description

3.6.3.1 Peripheral Request Line

8 groups of DMA channels are built in DMA. Peripheral requests of each channel need to be configured to map to peripheral devices. Please refer to 3.5.2.4 System DMA Channel Mapping to configure before DMA channel is enabled.

3.6.3.2 Access Space

Table 3- 6 DMAC Access Space Type

Space Type	Description
Memory	SRAM
	Non secure DDR space
Peripherals	UART0~UART3
	I2C0~I2C4
	SPI0~SPI3
	SPI_NAND
	I2S0~I2S3

3.6.3.3 Basic Transmission

DMA data transmission is set by block transmission, which is completed by burst transmission. The length of burst transmission can be set. However, what often happens is that the block data volume is not perfectly an integer multiple of the burst transmission length. The length of the last transaction of transmission is less than the set burst transmission length. In this case, it will need to use a single transmission request to complete.

The source and destination of the maximum 8 DMA channels can be in the following four combinations. :

a. memory to memory
b. memory to device
c. Device to memory
d. Device to device

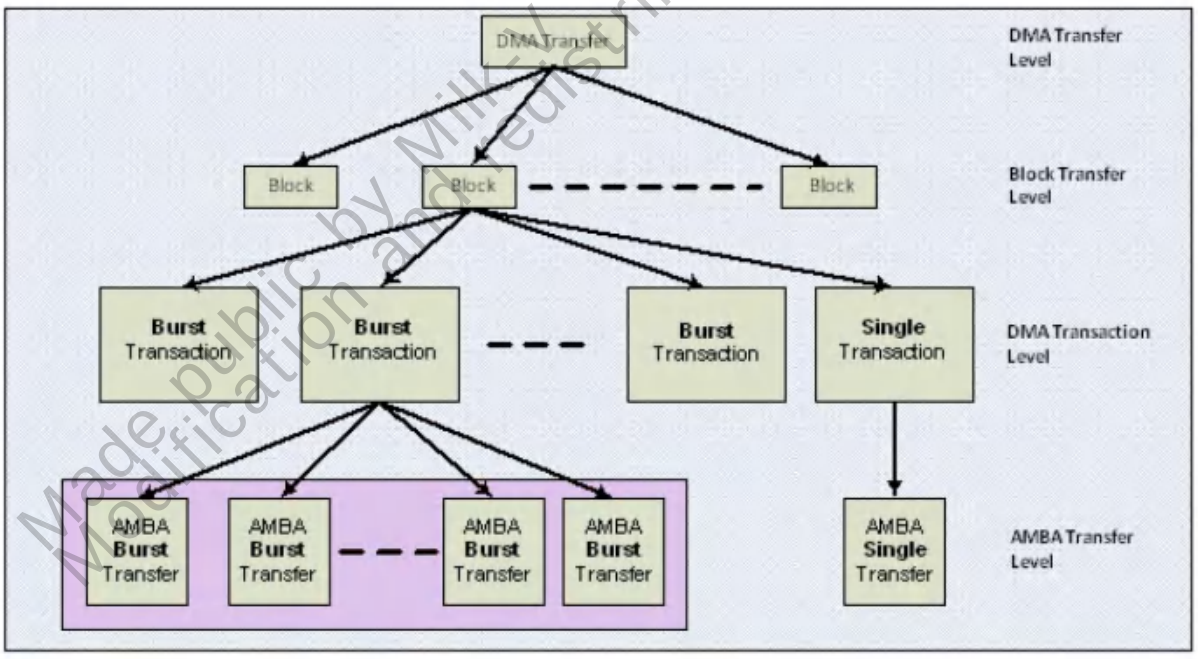


Figure 3- 5 DMA Transmission Structure

The individual data transmission amount can be calculated from the values written by the following registers.

- Transmission data amount from the source (bytes):

src_single_size_bytes = CHx_CTL.SRC_TR_WIDTH/8

- Burst transmission data amount from the source(bytes):
  src_burst_size_bytes = CHx_CTL.SRC_MSIZE * src_single_size_bytes
- Target transmission data amount(bytes):
  dst_single_size_bytes = CHx_CTL.DST_TR_WIDTH/8
- Target burst transmission data amount(bytes):
  dst_burst_size_bytes = CHx_CTL.DST_MSIZE * dst_single_size_bytes

The control right of transmission process can be controlled by DMA controller or source device or destination device. When block of data is transmitted, the amount of data transmitted is calculated as follows.

- The DMA controller controls the transmission process:
  blk_size_bytes_dma = CHx_BLOCK_TS.BLOCK_TS * src_single_size_bytes
- The source device controls the transmission process:
  blk_size_bytes_src = (number of block burst transmissions from source device * src_burst_size_bytes) + (number of independent transmission of source device block * src_single_size_bytes)
- The target device controls the transmission process:
  blk_size_bytes_dst = (number of block burst transmissions from target device * dst_burst_size_bytes) + (number of independent transmission of target device block * dst_single_size_bytes)

3.6.3.4 Linked-List Transmission

Linked list transmission is used in block transmission which needs to carry out multiple discontinuous addresses. After each block data, there will be a linked list information to store the information of the next node, so that the data transmission can directly carry out the block transmission of the next discontinuous space without the intervention of CPU. Figure 3-6 shows the configuration format of linked list information, which must conform to the information format to enter the linked list transmission work.

	31		0
0x3C	Reserved		
0x38	Reserved		
0x34	CHx_LLP_STATUS [63:32]		
0x30	CHx_LLP_STATUS [31:0]		
0x2C	Write back for CHx_DSTAT		
0x28	Write back for CHx_SSTAT		
0x24	CHx_CTL [63:32]		
0x20	CHx_CTL [31:0]		
0x1C	CHx_LLP [63:32]		
0x18	CHx_LLP [31:5]	6 5 Reserved	
0x14	Reserved		
0x10	CHx_BLOCK_TS [31:0]		
0x0C	CHx_DAR [63:32]		
0x08	CHx_DAR [31:0]		
0x04	CHx_SAR [63:32]		
0x00	CHx_SAR [31:0]		

Figure 3- 6 Linked List Relative Address and Data Format

3.6.3.5 Interrupt and Status

The interrupt sources of DMAC are as follows.

a. the completion of DMA transfer
b. the completion of block transfer
c. the completion of single transfer
d. internal error
e. the halt in channel or the termination of channels

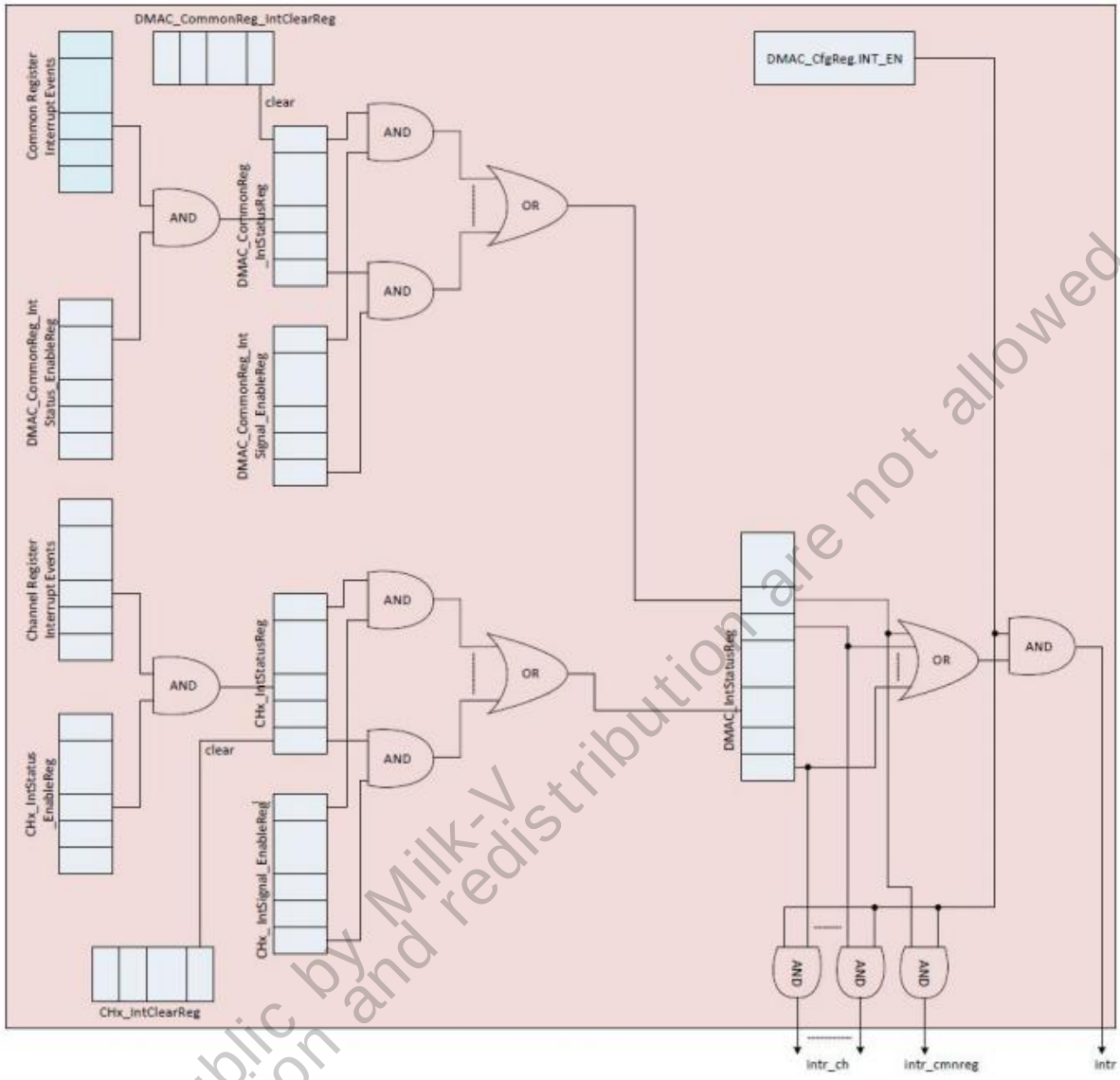


Figure 3- 7 Interrupt Status and Source Diagram

3.6.3.6 Channel Security Configuration

Channel security can be realized by awprot value and arprot value of each channel. According to AXI protocol, when the channel is a secure channel, arprot or awprot value should be 0x0, otherwise it is a non secure channel.

3.6.4 Working Mode

3.6.4.1 Clock and Reset

The clock of DMAC passes through CLK_EN_1[1] after writing 0x1; the clock can work normally. Writing 0x0 in REG_SOFT_RESET_X_SDMA_INIT can reset DMAC and writing 0x1 to release reset.

3.6.4.2 Initialization

After reset, it can be initialized following the steps below.

1. Peripheral Configuration: in the chapter of System DMA Channel Mapping, the configuration method of DMA peripheral request line is described, and the mapping should be configurated according to the scenario.
2. Confirm that the channel is closed: write 0x0 to DMA_ChEnReg and confirm that the channel is closed.
3. Confirm the interrupt source: writing 0x0 to the register DMAC_COMMONREG_INTSIGNAL_ENABLEREG and CHx_INSTATUS_ENABLEREG to turn off all interrupt sources, and then writes 0x1 to the required interrupt source to enable.
4. Configure the channel priority: when multiple channels transmit data at the same time, it will determine the passing order based on the priority level. The higher value configed in register CH_PRIOR , the higher priority the transmission is.

3.6.4.3 Basic Transmission

Up to 8 channels can be transmitted at the same time. After initialization, DMAC channel must be enabled before data transmission starts. Refer to the following steps for data transmission from internal memory to internal memory.

- Read register DMAC_ChEnReg to get the idle channel.
- Write 0x0 to channel register SRC_MULTBLK_TYPE and DST_MULTBLK_TYPE respectively to configure for continuous block transmission.

- Write 0x0 to the register TT_FC to configure the channel for memory to memory data transmission.
- Write the transmitted information to the register CHx_SAR, CHx_ADR and CHx_BLOCK_TS, CHx_CTL.
- Write 0x1 to the register DMAC_ChEnReg to enable the selected DMA channel.
- The software can obtain the status of BLOCK_TFR_DONE by interrupting or polling. When its value rises to 1, it means that the data transmission has been completed. Afterwards, write 0x0 to DMAC_ChEnReg to close the channel and restore it to an idle channel.

3.6.4.4 Linked List Transmission

Linked list transmission does not limit the number of nodes. Except for the ending node, each node must have information pointing to the next node. The linked list transmission can be completed by referring to the following steps.

1. Read register DMAC_ChEnReg to get the idle channel.
2. Write 0x3 to channel register SRC_MULTBLK_TYPE and DST_MULTBLK_TYPE respectively to configure for linked list transmission.
3. Configure register CHx_LLP, CHx_CTL.ShadowReg_Or_LLI_Valid and CHx_CTL.LLI_Last. Write the information needed to point to the first node.
4. Write 0x1 to the register DMAC_ChEnReg to enable the selected DMA channel.
5. The software can obtain the status of BLOCK_TFR_DONE by interrupting or polling. When its value rises to 1, it means that the data transmission of the ending node has been completed. Afterwards, write 0x0 to DMAC_ChEnReg to close the channel and restore it to an idle channel.

3.6.4.5 Interrupt Handling

The processing after the interrupt is triggered are as follows.

1. Find out the interrupt source: read the register CHx_INTSTATUS and DMAC_INSTATUSREG to get the interrupt source whose value is 0x1. When an interrupt occurs, it will be recorded as 0x1 in the corresponding selected

bit. If multiple interrupts occur at the same time, the software will serve based on its priority.

2. Clear interrupt: write 0x1 to **CHx_INTCLEARREG** or the selected bit of **DMAC_INTCLEARREG**. At this time, the recorded interrupt status of CHx_INTSTATUS and DMAC_INTSTATUSREG will return to 0x0 to record the next interrupt condition.

3.6.5 DMAC Register

DMAC_IDREG

Offset Address: 0x000

Bits	Name	Access	Description	Reset
63:0	DMAC_IDREG	RO	DMAC ID Number	

DMAC_COMPVERREG

Offset Address: 0x008

Bits	Name	Access	Description	Reset
31:0	DMAC_COMPVER	RO	DMAC Component Version Number.	

DMAC_CFGREG

Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	DMAC_EN	R/W	This bit is used to enable the DW_axi_dmac. ■ 0: DW_axi_dmac disabled ■ 1: DW_axi_dmac enabled NOTE: If this bit DMAC_EN bit is cleared while any channel is still active, then this bit still returns 1 to indicate that there are channels still active until DW_axi_dmac hardware has terminated all activity on all channels, at which point this bit returns zero (0).	0x0
1	INT_EN	R/W	This bit is used to globally enable the interrupt generation. ■ 0: DW_axi_dmac Interrupts are disabled ■ 1: DW_axi_dmac Interrupt logic is enabled.	0x0
31:2	Reserved			

DMAC_CHENREG

Offset Address: 0x018

Bits	Name	Access	Description	Reset
0	CH1_EN	R/W	This bit is used to enable the DW_axi_dmac Channel-1. ■ 0: DW_axi_dmac Channel-1 is disabled ■ 1: DW_axi_dmac Channel-1 is enabled The bit 'DMAC_ChEnReg.CH1_EN' is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.	0x0
1	CH2_EN	R/W	This bit is used to enable the DW_axi_dmac Channel-2. ■ 0: DW_axi_dmac Channel-2 is disabled ■ 1: DW_axi_dmac Channel-2 is enabled The bit 'DMAC_ChEnReg.CH2_EN' is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.	0x0
2	CH3_EN	R/W	This bit is used to enable the DW_axi_dmac Channel-3. ■ 0: DW_axi_dmac Channel-3 is disabled ■ 1: DW_axi_dmac Channel-3 is enabled The bit 'DMAC_ChEnReg.CH3_EN' is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.	0x0
3	CH4_EN	R/W	This bit is used to enable the DW_axi_dmac Channel-4. ■ 0: DW_axi_dmac Channel-4 is disabled ■ 1: DW_axi_dmac Channel-4 is enabled	0x0

Bits	Name	Access	Description	Reset
			The bit 'DMAC_ChEnReg.CH4_EN' is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.	
4	CH5_EN	R/W	This bit is used to enable the DW_axi_dmac Channel-5. ■ 0: DW_axi_dmac Channel-5 is disabled ■ 1: DW_axi_dmac Channel-5 is enabled The bit 'DMAC_ChEnReg.CH5_EN' is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.	0x0
5	CH6_EN	R/W	This bit is used to enable the DW_axi_dmac Channel-6. ■ 0: DW_axi_dmac Channel-6 is disabled ■ 1: DW_axi_dmac Channel-6 is enabled The bit 'DMAC_ChEnReg.CH6_EN' is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.	0x0
6	CH7_EN	R/W	This bit is used to enable the DW_axi_dmac Channel-7. ■ 0: DW_axi_dmac Channel-7 is disabled ■ 1: DW_axi_dmac Channel-7 is enabled The bit 'DMAC_ChEnReg.CH7_EN' is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new	0x0

Bits	Name	Access	Description	Reset
			DMA transfer.	
7	CH8_EN	R/W	This bit is used to enable the DW_axi_dmac Channel-8. ■ 0: DW_axi_dmac Channel-8 is disabled ■ 1: DW_axi_dmac Channel-8 is enabled The bit 'DMAC_ChEnReg.CH8_EN' is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.	0x0
8	CH1_EN_WE	WO	DW_axi_dmac Channel-1 Enable Write Enable bit. Read back value of this register bit is always '0'.	0x0
9	CH2_EN_WE	WO	DW_axi_dmac Channel-2 Enable Write Enable bit. Read back value of this register bit is always '0'.	0x0
10	CH3_EN_WE	WO	DW_axi_dmac Channel-3 Enable Write Enable bit. Read back value of this register bit is always '0'.	0x0
11	CH4_EN_WE	WO	DW_axi_dmac Channel-4 Enable Write Enable bit. Read back value of this register bit is always '0'.	0x0
12	CH5_EN_WE	WO	DW_axi_dmac Channel-5 Enable Write Enable bit. Read back value of this register bit is always '0'.	0x0
13	CH6_EN_WE	WO	DW_axi_dmac Channel-6 Enable Write Enable bit. Read back value of this register bit is always '0'.	0x0
14	CH7_EN_WE	WO	DW_axi_dmac Channel-7 Enable Write Enable bit. Read back value of this register bit is always '0'.	0x0
15	CH8_EN_WE	WO	DW_axi_dmac Channel-8 Enable Write Enable bit. Read back value of this register bit is	0x0

Bits	Name	Access	Description	Reset
			always '0'.	
16	CH1_SUSP	R/W	Channel-1 Suspend Request. Software sets this bit to 1 to request channel suspend. If this bit is set to 1, DW_axi_dmac suspends all DMA data transfers from the source gracefully until this bit is cleared. There is no guarantee that the current dma transaction will complete. This bit can also be used in conjunction with CH1_Status.CH_SUSPENDED to cleanly disable the channel without losing any data. In this case, software first sets CH1_SUSP bit to 1 and polls CH1_Status.CH_SUSPENDED till it is set to 1. Software can then clear CH1_EN bit to 0 to disable the channel. ■ 0: No Channel Suspend Request. ■ 1: Request for Channel Suspend. Software can clear CH1_SUSP bit to 0, after DW_axi_dmac sets CH1_Status.CH_SUSPENDED bit to 1, to exit the channel suspend mode. Note: CH_SUSP is cleared when channel is disabled.	0x0
17	CH2_SUSP	R/W	Channel-2 Suspend Request. Software sets this bit to 1 to request channel suspend. If this bit is set to 1, DW_axi_dmac suspends all DMA data transfers from the source gracefully until this bit is cleared. There is no guarantee that the current dma transaction will complete. This bit can also be used in conjunction with CH2_Status.CH_SUSPENDED to cleanly disable the channel without losing any data. In this case, software first sets CH2_SUSP bit to 1 and polls CH2_Status.CH_SUSPENDED till it is set to 1. Software can then clear CH2_EN bit to 0 to disable the channel. ■ 0: No Channel Suspend Request. ■ 1: Request for Channel Suspend. Software can clear CH2_SUSP bit to 0, after DW_axi_dmac sets CH2_Status.CH_SUSPENDED bit to 1, to exit the	0x0

Bits	Name	Access	Description	Reset
			channel suspend mode. Note: CH_SUSP is cleared when channel is disabled.	
18	CH3_SUSP	R/W	Channel-3 Suspend Request. Software sets this bit to 1 to request channel suspend. If this bit is set to 1, DW_axi_dmac suspends all DMA data transfers from the source gracefully until this bit is cleared. There is no guarantee that the current dma transaction will complete. This bit can also be used in conjunction with CH3_Status.CH_SUSPENDED to cleanly disable the channel without losing any data. In this case, software first sets CH3_SUSP bit to 1 and polls CH3_Status.CH_SUSPENDED till it is set to 1. Software can then clear CH3_EN bit to 0 to disable the channel. ■ 0: No Channel Suspend Request. ■ 1: Request for Channel Suspend. Software can clear CH3_SUSP bit to 0, after DW_axi_dmac sets CH3_Status.CH_SUSPENDED bit to 1, to exit the channel suspend mode. Note: CH_SUSP is cleared when channel is disabled.	0x0
19	CH4_SUSP	R/W	Channel-4 Suspend Request. Software sets this bit to 1 to request channel suspend. If this bit is set to 1, DW_axi_dmac suspends all DMA data transfers from the source gracefully until this bit is cleared. There is no guarantee that the current dma transaction will complete. This bit can also be used in conjunction with CH4_Status.CH_SUSPENDED to cleanly disable the channel without losing any data. In this case, software first sets CH4_SUSP bit to 1 and polls CH4_Status.CH_SUSPENDED till it is set to 1. Software can then clear CH4_EN bit to 0 to disable the channel. ■ 0: No Channel Suspend Request. ■ 1: Request for Channel Suspend. Software can clear CH4_SUSP bit to 0, after DW_axi_dmac sets CH4_Status.CH_SUSPENDED bit to	0x0

Bits	Name	Access	Description	Reset
			1, to exit the channel suspend mode. Note: CH_SUSP is cleared when channel is disabled.	
20	CH5_SUSP	R/W	Channel-5 Suspend Request. Software sets this bit to 1 to request channel suspend. If this bit is set to 1, DW_axi_dmac suspends all DMA data transfers from the source gracefully until this bit is cleared. There is no guarantee that the current dma transaction will complete. This bit can also be used in conjunction with CH5_Status.CH_SUSPENDED to cleanly disable the channel without losing any data. In this case, software first sets CH5_SUSP bit to 1 and polls CH5_Status.CH_SUSPENDED till it is set to 1. Software can then clear CH5_EN bit to 0 to disable the channel. ■ 0: No Channel Suspend Request. ■ 1: Request for Channel Suspend. Software can clear CH5_SUSP bit to 0, after DW_axi_dmac sets CH5_Status.CH_SUSPENDED bit to 1, to exit the channel suspend mode. Note: CH_SUSP is cleared when channel is disabled.	0x0
21	CH6_SUSP	R/W	Channel-6 Suspend Request. Software sets this bit to 1 to request channel suspend. If this bit is set to 1, DW_axi_dmac suspends all DMA data transfers from the source gracefully until this bit is cleared. There is no guarantee that the current dma transaction will complete. This bit can also be used in conjunction with CH6_Status.CH_SUSPENDED to cleanly disable the channel without losing any data. In this case, software first sets CH6_SUSP bit to 1 and polls CH6_Status.CH_SUSPENDED till it is set to 1. Software can then clear CH6_EN bit to 0 to disable the channel. ■ 0: No Channel Suspend Request. ■ 1: Request for Channel Suspend. Software can clear CH6_SUSP bit to 0, after DW_axi_dmac	0x0

Bits	Name	Access	Description	Reset
			sets CH6_Status.CH_SUSPENDED bit to 1, to exit the channel suspend mode. Note: CH_SUSP is cleared when channel is disabled.	
22	CH7_SUSP	R/W	Channel-7 Suspend Request. Software sets this bit to 1 to request channel suspend. If this bit is set to 1, DW_axi_dmac suspends all DMA data transfers from the source gracefully until this bit is cleared. There is no guarantee that the current dma transaction will complete. This bit can also be used in conjunction with CH7_Status.CH_SUSPENDED to cleanly disable the channel without losing any data. In this case, software first sets CH7_SUSP bit to 1 and polls CH7_Status.CH_SUSPENDED till it is set to 1. Software can then clear CH7_EN bit to 0 to disable the channel. ■ 0: No Channel Suspend Request. ■ 1: Request for Channel Suspend. Software can clear CH7_SUSP bit to 0, after DW_axi_dmac sets CH7_Status.CH_SUSPENDED bit to 1, to exit the channel suspend mode. Note: CH_SUSP is cleared when channel is disabled.	0x0
23	CH8_SUSP	R/W	Channel-8 Suspend Request. Software sets this bit to 1 to request channel suspend. If this bit is set to 1, DW_axi_dmac suspends all DMA data transfers from the source gracefully until this bit is cleared. There is no guarantee that the current dma transaction will complete. This bit can also be used in conjunction with CH8_Status.CH_SUSPENDED to cleanly disable the channel without losing any data. In this case, software first sets CH8_SUSP bit to 1 and polls CH8_Status.CH_SUSPENDED till it is set to 1. Software can then clear CH8_EN bit to 0 to disable the channel. ■ 0: No Channel Suspend Request. ■ 1: Request for Channel Suspend. Software can clear CH8_SUSP bit to 0,	0x0

Bits	Name	Access	Description	Reset
			after DW_axi_dmac sets CH8_Status.CH_SUSPENDED bit to 1, to exit the channel suspend mode. Note: CH_SUSP is cleared when channel is disabled.	
24	CH1_SUSP_WE	WO	This bit is used as a write enable to the Channel-1 Suspend bit. The read back value of this register bit is always 0.	0x0
25	CH2_SUSP_WE	WO	This bit is used as a write enable to the Channel-2 Suspend bit. The read back value of this register bit is always 0.	0x0
26	CH3_SUSP_WE	WO	This bit is used as a write enable to the Channel-3 Suspend bit. The read back value of this register bit is always 0.	0x0
27	CH4_SUSP_WE	WO	This bit is used as a write enable to the Channel-4 Suspend bit. The read back value of this register bit is always 0.	0x0
28	CH5_SUSP_WE	WO	This bit is used as a write enable to the Channel-5 Suspend bit. The read back value of this register bit is always 0.	0x0
29	CH6_SUSP_WE	WO	This bit is used as a write enable to the Channel-6 Suspend bit. The read back value of this register bit is always 0.	0x0
30	CH7_SUSP_WE	WO	This bit is used as a write enable to the Channel-7 Suspend bit. The read back value of this register bit is always 0.	0x0
31	CH8_SUSP_WE	WO	This bit is used as a write enable to the Channel-8 Suspend bit. The read back value of this register bit is always 0.	0x0
32	CH1_ABORT	R/W	Channel-1 Abort Request. Software sets this bit to 1 to request channel abort. If this bit is set to 1, DW_axi_dmac disables the channel immediately. Aborting the channel might result in AXI Protocol violation as DW_axi_dmac does not make sure that all AXI transfers initiated on the master interface are completed.Aborting the channel is not recommended and should be used only in situations where a particular channel hangs due to no response from the corresponding AXI slave interface and software wants to disable the channel without resetting the	0x0

Bits	Name	Access	Description	Reset
			entire DW_axi_dmac. It is recommended to try channel disabling first and then only opt for channel aborting. ■ 0: No Channel Abort Request. ■ 1: Request for Channel Abort. DW_axi_dmac clears this bit to 0 once the channel is aborted (when it sets CH1_Status.CH_ABORTED bit to 1).	
33	CH2_ABORT	R/W	Channel-2 Abort Request. Software sets this bit to 1 to request channel abort. If this bit is set to 1, DW_axi_dmac disables the channel immediately. Aborting the channel might result in AXI Protocol violation as DW_axi_dmac does not make sure that all AXI transfers initiated on the master interface are completed.Aborting the channel is not recommended and should be used only in situations where a particular channel hangs due to no response from the corresponding AXI slave interface and software wants to disable the channel without resetting the entire DW_axi_dmac. It is recommended to try channel disabling first and then only opt for channel aborting. ■ 0: No Channel Abort Request. ■ 1: Request for Channel Abort. DW_axi_dmac clears this bit to 0 once the channel is aborted (when it sets CH2_Status.CH_ABORTED bit to 1).	0x0
34	CH3_ABORT	R/W	Channel-3 Abort Request. Software sets this bit to 1 to request channel abort. If this bit is set to 1, DW_axi_dmac disables the channel immediately. Aborting the channel might result in AXI Protocol violation as DW_axi_dmac does not make sure that all AXI transfers initiated on the master interface are completed.Aborting the channel is not recommended and should be used only in situations where a particular channel hangs due to no response from the corresponding AXI slave interface and software wants to disable the channel	0x0

Bits	Name	Access	Description	Reset
			without resetting the entire DW_axi_dmac. It is recommended to try channel disabling first and then only opt for channel aborting. ■ 0: No Channel Abort Request. ■ 1: Request for Channel Abort. DW_axi_dmac clears this bit to 0 once the channel is aborted (when it sets CH3_Status.CH_ABORTED bit to 1).	
35	CH4_ABORT	R/W	Channel-4 Abort Request. Software sets this bit to 1 to request channel abort. If this bit is set to 1, DW_axi_dmac disables the channel immediately. Aborting the channel might result in AXI Protocol violation as DW_axi_dmac does not make sure that all AXI transfers initiated on the master interface are completed.Aborting the channel is not recommended and should be used only in situations where a particular channel hangs due to no response from the corresponding AXI slave interface and software wants to disable the channel without resetting the entire DW_axi_dmac. It is recommended to try channel disabling first and then only opt for channel aborting. ■ 0: No Channel Abort Request. ■ 1: Request for Channel Abort. DW_axi_dmac clears this bit to 0 once the channel is aborted (when it sets CH4_Status.CH_ABORTED bit to 1).	0x0
36	CH5_ABORT	R/W	Channel-5 Abort Request. Software sets this bit to 1 to request channel abort. If this bit is set to 1, DW_axi_dmac disables the channel immediately. Aborting the channel might result in AXI Protocol violation as DW_axi_dmac does not make sure that all AXI transfers initiated on the master interface are completed.Aborting the channel is not recommended and should be used only in situations where a particular channel hangs due to no response from the corresponding AXI slave interface and	0x0

Bits	Name	Access	Description	Reset
			software wants to disable the channel without resetting the entire DW_axi_dmac. It is recommended to try channel disabling first and then only opt for channel aborting. ■ 0: No Channel Abort Request. ■ 1: Request for Channel Abort. DW_axi_dmac clears this bit to 0 once the channel is aborted (when it sets CH5_Status.CH_ABORTED bit to 1).	
37	CH6_ABORT	R/W	Channel-6 Abort Request. Software sets this bit to 1 to request channel abort. If this bit is set to 1, DW_axi_dmac disables the channel immediately. Aborting the channel might result in AXI Protocol violation as DW_axi_dmac does not make sure that all AXI transfers initiated on the master interface are completed.Aborting the channel is not recommended and should be used only in situations where a particular channel hangs due to no response from the corresponding AXI slave interface and software wants to disable the channel without resetting the entire DW_axi_dmac. It is recommended to try channel disabling first and then only opt for channel aborting. ■ 0: No Channel Abort Request. ■ 1: Request for Channel Abort. DW_axi_dmac clears this bit to 0 once the channel is aborted (when it sets CH6_Status.CH_ABORTED bit to 1).	0x0
38	CH7_ABORT	R/W	Channel-7 Abort Request. Software sets this bit to 1 to request channel abort. If this bit is set to 1, DW_axi_dmac disables the channel immediately. Aborting the channel might result in AXI Protocol violation as DW_axi_dmac does not make sure that all AXI transfers initiated on the master interface are completed.Aborting the channel is not recommended and should be used only in situations where a particular channel hangs due to no response from the corresponding AXI	0x0

Bits	Name	Access	Description	Reset
			slave interface and software wants to disable the channel without resetting the entire DW_axi_dmac. It is recommended to try channel disabling first and then only opt for channel aborting. ■ 0: No Channel Abort Request. ■ 1: Request for Channel Abort. DW_axi_dmac clears this bit to 0 once the channel is aborted (when it sets CH7_Status.CH_ABORTED bit to 1).	
39	CH8_ABORT	R/W	Channel-8 Abort Request. Software sets this bit to 1 to request channel abort. If this bit is set to 1, DW_axi_dmac disables the channel immediately. Aborting the channel might result in AXI Protocol violation as DW_axi_dmac does not make sure that all AXI transfers initiated on the master interface are completed.Aborting the channel is not recommended and should be used only in situations where a particular channel hangs due to no response from the corresponding AXI slave interface and software wants to disable the channel without resetting the entire DW_axi_dmac. It is recommended to try channel disabling first and then only opt for channel aborting. ■ 0: No Channel Abort Request. ■ 1: Request for Channel Abort. DW_axi_dmac clears this bit to 0 once the channel is aborted (when it sets CH8_Status.CH_ABORTED bit to 1).	0x0
40	CH1_ABORT_WE	R/W	This bit is used to write enable the Channel-1 Abort bit. The read back value of this register bit is always 0.	0x0
41	CH2_ABORT_WE	R/W	This bit is used to write enable the Channel-2 Abort bit. The read back value of this register bit is always 0.	0x0
42	CH3_ABORT_WE	R/W	This bit is used to write enable the Channel-3 Abort bit. The read back value of this register bit is always 0.	0x0
43	CH4_ABORT_WE	R/W	This bit is used to write enable the Channel-4 Abort bit. The read back value of this register bit is	0x0

Bits	Name	Access	Description	Reset
			always 0.	
44	CH5_ABORT_WE	R/W	This bit is used to write enable the Channel-5 Abort bit. The read back value of this register bit is always 0.	0x0
45	CH6_ABORT_WE	R/W	This bit is used to write enable the Channel-6 Abort bit. The read back value of this register bit is always 0.	0x0
46	CH7_ABORT_WE	R/W	This bit is used to write enable the Channel-7 Abort bit. The read back value of this register bit is always 0.	0x0
47	CH8_ABORT_WE	R/W	This bit is used to write enable the Channel-8 Abort bit. The read back value of this register bit is always 0.	0x0
63:48	RSVD_DMAC_CHENREG	RO	DMAC_CHENREG Reserved bits	

DMAC_INTSTATUSREG

Offset Address: 0x030

Bits	Name	Access	Description	Reset
0	CH1_IntStat	RO	Channel 1 Interrupt Status Bit.	
1	CH2_IntStat	RO	Channel 2 Interrupt Status Bit.	
2	CH3_IntStat	RO	Channel 3 Interrupt Status Bit.	
3	CH4_IntStat	RO	Channel 4 Interrupt Status Bit.	
4	CH5_IntStat	RO	Channel 5 Interrupt Status Bit.	
5	CH6_IntStat	RO	Channel 6 Interrupt Status Bit.	
6	CH7_IntStat	RO	Channel 7 Interrupt Status Bit.	
7	CH8_IntStat	RO	Channel 8 Interrupt Status Bit.	
15:8	Reserved			
16	CommonReg_IntStat	RO	Common Register Interrupt Status Bit.	
31:17	Reserved			

DMAC_COMMONREG_INTCLEARREG

Offset Address: 0x038

Bits	Name	Access	Description	Reset
0	Clear_SLVIF_CommonReg_DEC_ERR_IntStat	WO	Slave Interface Common Register Decode Error Interrupt clear Bit. This bit is used to clear the corresponding channel interrupt status bit (SLVIF_CommonReg_DEC_ERR_IntStat in DMAC_CommonReg_IntStatusReg.	0x0
1	Clear_SLVIF_CommonReg_WR2RO_ERR_IntStat	WO	Slave Interface Common Register Write to Read only Error Interrupt clear Bit. This bit is used to clear the corresponding channel interrupt	0x0

Bits	Name	Access	Description	Reset
			status bit(SLVIF_CommonReg_WR2RO_ERR_IntStat in DMAC_CommonReg_IntStatusReg.	
2	Clear_SLVIF_CommonReg_RD2WO_ERR_IntStat	WO	Slave Interface Common Register Read to Write only Error Interrupt clear Bit. This bit is used to clear the corresponding channel interrupt status bit(SLVIF_CommonReg_RD2WO_ERR_IntStat in DMAC_CommonReg_IntStatusReg.	0x0
3	Clear_SLVIF_CommonReg_WrOnHold_ERR_IntStat	WO	Slave Interface Common Register Write On Hold Error Interrupt clear Bit. This bit is used to clear the corresponding channel interrupt status bit(SLVIF_CommonReg_WrOnHold_ERR_IntStat in DMAC_CommonReg_IntStatusReg.	0x0
7:4	Reserved			
8	Clear_SLVIF_UndefinedReg_DEC_ERR_IntStat	WO	Slave Interface Undefined register Decode Error Interrupt clear Bit. This bit is used to clear the corresponding channel interrupt status bit(SLVIF_UndefinedReg_DEC_ERR_IntStat in DMAC_CommonReg_IntStatusReg.	0x0
31:9	Reserved			

DMAC_COMMONREG_INTSTATUS_ENABLEREG

Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	Enable_SLVIF_CommonReg_DEC_ERR_IntStat	R/W	Slave Interface Common Register Decode Error Interrupt Status Enable Bit. This bit is used to enable the corresponding channel interrupt status bit (SLVIF_CommonReg_DEC_ERR_IntStat in DMAC_CommonReg_IntStatusReg.	0x0
1	Enable_SLVIF_CommonReg_WR2RO_ERR_IntStat	R/W	Slave Interface Common Register Write to Read only Error Interrupt Status Enable Bit. This bit is used to enable the corresponding channel interrupt status bit (SLVIF_CommonReg_WR2RO_ERR_IntStat in DMAC_CommonReg_IntStatusReg.	0x0

Bits	Name	Access	Description	Reset
2	Enable_SLVIF_CommonReg_RD2WO_ERR_IntStat	R/W	Slave Interface Common Register Read to Write only Error Interrupt Status Enable Bit. This bit is used to enable the corresponding channel interrupt status bit (SLVIF_CommonReg_RD2WO_ERR_IntStat in DMAC_CommonReg_IntStatusReg.	0x0
3	Enable_SLVIF_CommonReg_WrOnHold_ERR_IntStat	R/W	Slave Interface Common Register Write On Hold Error Interrupt Status Enable Bit. This bit is used to enable the corresponding channel interrupt status bit (SLVIF_CommonReg_WrOnHold_ERR_IntStat in DMAC_CommonReg_IntStatusReg.	0x0
7:4	Reserved			
8	Enable_SLVIF_UndefinedReg_DEC_ERR_IntStat	R/W	Slave Interface Undefined register Decode Error Interrupt Status enable Bit. This bit is used to enable the corresponding channel interrupt status bit (SLVIF_UndefinedReg_DEC_ERR_IntStat in DMAC_CommonReg_IntStatusReg.	0x0
31:9	Reserved			

DMAC_COMMONREG_INTSIGNAL_ENABLEREG

Offset Address: 0x048

Bits	Name	Access	Description	Reset
0	Enable_SLVIF_CommonReg_DEC_ERR_IntSignal	R/W	Slave Interface Common Register Decode Error Interrupt Signal Enable Bit. This bit is used to enable the propagation of corresponding channel interrupt status bit (SLVIF_CommonReg_DEC_ERR_IntStat in DMAC_CommonReg_IntStatusReg) to generate a port level interrupt.	0x0
1	Enable_SLVIF_CommonReg_WR2RO_ERR_IntSignal	R/W	Slave Interface Common Register Write to Read only Error Interrupt Signal Enable Bit. This bit is used to enable the propagation of corresponding channel interrupt status bit (SLVIF_CommonReg_WR2RO_ERR_IntStat in DMAC_CommonReg_IntStatusReg) to generate a port level	0x0

Bits	Name	Access	Description	Reset
			interrupt.	
2	Enable_SLVIF_CommonReg_RD2WO_ERR_IntSignal	R/W	Slave Interface Common Register Read to Write only Error Interrupt Signal Enable Bit. This bit is used to enable the propagation of corresponding channel interrupt status bit (SLVIF_CommonReg_RD2WO_ERR_IntStat in DMAC_CommonReg_IntStatusReg) to generate a port level interrupt.	0x0
3	Enable_SLVIF_CommonReg_WrOnHold_ERR_IntSignal	R/W	Slave Interface Common Register Write On Hold Error Interrupt Signal Enable Bit. This bit is used to enable the propagation of corresponding channel interrupt status bit(SLVIF_CommonReg_WrOnHold_ERR_IntStat in DMAC_CommonReg_IntStatusReg) to generate a port level interrupt.	0x0
7:4	Reserved			
8	Enable_SLVIF_UndefinedReg_DEC_ERR_IntSignal	R/W	Slave Interface Undefined register Decode Error Interrupt Signal Enable Bit. This bit is used to enable the propagation of corresponding channel interrupt status bit(SLVIF_UndefinedReg_DEC_ERR_IntStat in DMAC_CommonReg_IntStatusReg) to generate a port level interrupt.	0x0
31:9	Reserved			

DMAC_COMMONREG_INTSTATUSREG

Offset Address: 0x050

Bits	Name	Access	Description	Reset
0	SLVIF_CommonReg_DEC_ERR_IntStat	RO	Slave Interface Common Register Decode Error Interrupt Status Bit. Decode Error generated by DW_axi_dmac during register access. This error occurs if the register access is to an invalid address in the common register space (0x000 to 0x0FF) resulting in error response by DW_axi_dmac slave interface. ■ 0: No Slave Interface Decode Errors. ■ 1: Slave Interface Decode Error detected.	

Bits	Name	Access	Description	Reset
			The Error Interrupt status is generated if the corresponding Status Enable bit in DMAC_CommonReg_IntStatus_Enable register bit is set to 1. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in DMAC_COMMONREG_INTCLEARREG on enabling the channel (required when the interrupt is not enabled).	
1	SLVIF_CommonReg_WR2RO_ERR_IntStat	RO	Slave Interface Common Register Write to Read Only Error Interrupt Status bit. This error occurs if write operation is performed to a Read Only register in the common register space (0x000 to 0x0FF). ■ 0: No Slave Interface Write to Read Only Errors. ■ 1: Slave Interface Write to Read Only Error detected. Error Interrupt status is generated if the corresponding Status Enable bit in DMAC_CommonReg_IntStatus_Enable register bit is set to 1. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in DMAC_COMMONREG_INTCLEARREG on enabling the channel (required when the interrupt is not enabled).	
2	SLVIF_CommonReg_RD2WO_ERR_IntStat	RO	Slave Interface Common Register Read to Write only Error Interrupt Status bit. This error occurs if Read operation is performed to a Write Only register in the common register space (0x000 to 0x0FF). ■ 0: No Slave Interface Read to Write Only Errors. ■ 1: Slave Interface Read to Write Only Error detected. Error Interrupt status is generated if the corresponding Status Enable bit in DMAC_CommonReg_IntStatus_Enable register bit is set to 1. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in DMAC_COMMONREG_INTCLEARREG on	

Bits	Name	Access	Description	Reset
			enabling the channel (required when the interrupt is not enabled). Values: ■ 0x1 (Active_CommonReg_RD2WO_ERR): Slave Interface Read to Write Only Error detected ■ 0x0 (Inactive_CommonReg_RD2WO_ERR): No Slave Interface Read to Write Only Errors	
3	SLVIF_CommonReg_WrOnHold_ERR_IntStat	RO	Slave Interface Common Register Write On Hold Error Interrupt Status Bit. This error occurs if an illegal write operation is performed on a common register; this happens if a write operation is performed on a common register except DMAC_RESETREG with DMAC_RST field set to 1 when DW_axi_dmac is in Hold mode. ■ 0: No Slave Interface Common Register Write On Hold Errors. ■ 1: Slave Interface Common Register Write On Hold Error detected. Error Interrupt Status is generated if the corresponding Status Enable bit in DMAC_CommonReg_IntStatus_Enable register bit is set to 1. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in DMAC_COMMONREG_INTCLEARREG on enabling the channel (required when the interrupt is not enabled).	
7:4	Reserved			
8	SLVIF_UndefinedReg_DEC_ERR_IntStat	RO	Slave Interface Undefined register Decode Error Interrupt Signal Enable Bit. Decode Error generated by DW_axi_dmac during register access. This error occurs if the register access is to undefined address range (>0x8FF if 8 channels are configured, >0x4FF if 4 channels are configured etc.) resulting in error response by DW_axi_dmac slave interface. ■ 0: No Slave Interface Decode Errors.	

Bits	Name	Access	Description	Reset
			■ 1: Slave Interface Decode Error detected. Error Interrupt Status is generated if the corresponding Status Enable bit in DMAC_CommonReg_IntStatus_Enable register bit is set to 1. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in DMAC_COMMONREG_INTCLEARREG on enabling the channel (required when the interrupt is not enabled).	
31:9	Reserved			

DMAC_RESETREG

Offset Address: 0x058

Bits	Name	Access	Description	Reset
0	DMAC_RST	R/W	DMAC Reset Request bit Software writes 1 to this bit to reset the DW_axi_dmac and polls this bit to see it as 0. DW_axi_dmac resets all the modules except the slave bus interface module and clears this bit to 0. NOTE: Software is not allowed to write 0 to this bit.	0x0
31:1	Reserved			

CHx_SAR

Offset Address: 0x100

Bits	Name	Access	Description	Reset
63:0	SAR	R/W	Current Source Address of DMA transfer. Updated after each source transfer. The SINC fields in the CHx_CTL register determines whether the address increments or is left unchanged on every source transfer throughout the block transfer.	0x0

CHx_DAR

Offset Address: 0x108

Bits	Name	Access	Description	Reset
63:0	DAR	R/W	Current Destination Address of DMA transfer. Updated after each destination transfer. The DINC fields in the CHx_CTL register determines whether the address	0x0

Bits	Name	Access	Description	Reset
			increments or is left unchanged on every destination transfer throughout the block transfer.	

CHx_BLOCK_TS

Offset Address: 0x110

Bits	Name	Access	Description	Reset
21:0	BLOCK_TS	R/W	Block Transfer Size. The number programmed into BLOCK_TS field indicates the total number of data of width CHx_CTL.SRC_TR_WIDTH to be transferred in a DMA block transfer. Block Transfer Size = BLOCK_TS+1 When the transfer starts, the read-back value is the total number of data items already read from the source peripheral, regardless of who is the flow controller. When the source or destination peripheral is assigned as the flow controller, the value before the transfer starts saturates at DMAX_CHx_MAX_BLK_SIZE, but the actual block size can be greater.	0x0
31:22	Reserved			

CHx_CTL

Offset Address: 0x118

Bits	Name	Access	Description	Reset
0	SMS	R/W	Source Master Select. Identifies the Master Interface layer from which the source device (peripheral or memory) is accessed. ■ 0: AXI master 1 ■ 1: AXI Master 2	0x0
1	Reserved			
2	DMS	R/W	Destination Master Select. Identifies the Master Interface layer from which the destination device (peripheral or memory) is accessed. ■ 0: AXI master 1 ■ 1: AXI Master 2	0x0
3	Reserved			
4	SINC	R/W	Source Address Increment. Indicates whether to increment the source address on every source transfer. If the device is fetching data from a source peripheral FIFO with a fixed address,	0x0

Bits	Name	Access	Description	Reset
			then set this field to 'No change'. ■ 0: Increment ■ 1: No Change	
5	Reserved			
6	DINC	R/W	Destination Address Increment. Indicates whether to increment the destination address on every destination transfer. If the device is writing data from a source peripheral FIFO with a fixed address, then set this field to 'No change'. ■ 0: Increment ■ 1: No Change	0x0
7	Reserved			
10:8	SRC_TR_WIDTH	R/W	Source Transfer Width. Mapped to AXI bus arsize, this value must be less than or equal to DMAX_M_DATA_WIDTH.	0x0
13:11	DST_TR_WIDTH	R/W	Destination Transfer Width. Mapped to AXI bus awsize, this value must be less than or equal to DMAX_M_DATA_WIDTH.	0x0
17:14	SRC_MSIZE	R/W	Source Burst Transaction Length. Number of data items, each of width CHx_CTL.SRC_TR_WIDTH, to be read from the source every time a source burst transaction request is made from the corresponding hardware or software handshaking interface. The maximum value of DST_MSIZE is limited by DMAX_CHx_MAX_MULT_SIZE.	0x0
21:18	DST_MSIZE	R/W	Destination Burst Transaction Length. Number of data items, each of width CHx_CTL.DST_TR_WIDTH, to be written to the destination every time a destination burst transaction request is made from the corresponding hardware or software handshaking interface.Note: This Value is not related to the AXI awlen signal.	0x0
25:22	AR_CACHE	R/W	AXI 'ar_cache' signal	0x0
29:26	AW_CACHE	R/W	AXI 'aw_cache' signal	0x0
30	NonPosted_LastWrite_En	R/W	Non Posted Last Write Enable This bit decides whether posted writes can be used throughout the block transfer. ■ 0: Posted writes may be used throughout the block transfer. ■ 1: Posted writes may be used till the	0x0

Bits	Name	Access	Description	Reset
			end of the block (inside a block) and the last write in the block must be non-posted. This is to synchronize block completion interrupt generation to the last write data reaching the end memory/peripheral.	
31	Reserved			
34:32	AR_PROT	R/W	AXI 'ar_prot' signal	0x0
37:35	AW_PROT	R/W	AXI 'aw_prot' signal	0x0
38	ARLEN_EN	R/W	Source Burst Length Enable If this bit is set to 1, DW_axi_dmac uses the value of CHx_CTL.ARLEN as AXI Burst length for source data transfer till the extent possible; remaining transfers use maximum possible burst length. If this bit is set to 0, DW_axi_dmac uses any possible value that is less than or equal to DMAX_CHx_MAX_AMBA_BURST_LENGTH as AXI Burst length for source data transfer.	0x0
46:39	ARLEN	R/W	Source Burst Length AXI Burst length used for source data transfer. The specified burst length is used for source data transfer till the extent possible; remaining transfers use maximum possible value that is less than or equal to DMAX_CHx_MAX_AMBA_BURST_LENGTH. The maximum value of ARLEN is limited by DMAX_CHx_MAX_AMBA_BURST_LENGTH	0x0
47	AWLEN_EN	R/W	Destination Burst Length Enable If this bit is set to 1, DW_axi_dmac uses the value of CHx_CTL.AWLEN as AXI Burst length for destination data transfer till the extent possible; remaining transfers use maximum possible burst length. If this bit is set to 0, DW_axi_dmac uses any possible value which is less than or equal to DMAX_CHx_MAX_AMBA_BURST_LENGTH as AXI Burst length for destination data transfer.	0x0
55:48	AWLEN	RO	Destination Burst Length AXI Burst length used for destination data transfer. The specified burst length is used for	

Bits	Name	Access	Description	Reset
			destination data transfer till the extent possible; remaining transfers use maximum possible value that is less than or equal to DMAX_CHx_MAX_AMBA_BURST_LENGTH. The maximum value of AWLEN is limited by DMAX_CHx_MAX_AMBA_BURST_LENGTH.	
56	SRC_STAT_EN	R/W	Source Status Enable Enable the logic to fetch status from source peripheral of channel x pointed to by the content of CHx_SSTATAR register and stores it in CHx_SSTAT register. This value is written back to the CHx_SSTAT location of linked list at end of each block transfer if DMAX_CHx_LLI_WB_EN is set to 1 and if linked list based multi-block transfer is used by either source or destination peripheral.	0x0
57	DST_STAT_EN	R/W	Destination Status Enable Enable the logic to fetch status from destination peripheral of channel x pointed to by the content of CHx_DSTATAR register and stores it in CHx_DSTAT register. This value is written back to the CHx_DSTAT location of linked list at end of each block transfer if DMAX_CHx_LLI_WB_EN is set to 1 and if linked list based multi-block transfer is used by either source or destination peripheral.	0x0
58	IOC_BlkTfr	R/W	Interrupt On completion of Block Transfer This bit is used to control the block transfer completion interrupt generation on a block by block basis for shadow register or linked list based multi-block transfers. Writing 1 to this register field enables CHx_IntStatusReg.BLOCK_TFR_DONE_IntStat field if this interrupt generation is enabled in CHx_IntStatus_EnableReg register and the external interrupt output is is asserted if this interrupt generation is enabled in CHx_IntSignal_EnableReg register.	0x0

Bits	Name	Access	Description	Reset
61:59	Reserved			
62	SHADOWREG_OR_LLI_LAST	R/W	Last Shadow Register/Linked List Item. Indicates whether shadow register content or the linked list item fetched from the memory is the last one or not. ■ 0: Not last Shadow Register/LLI ■ 1: Last Shadow Register/LLI	0x0
63	SHADOWREG_OR_LLI_VALID	R/W	Shadow Register content/Linked List Item valid. Indicates whether the content of shadow register or the linked list item fetched from the memory is valid. ■ 0: Shadow Register content/LLI is invalid. ■ 1: Last Shadow Register/LLI is valid.	0x0

CHx_CFG

Offset Address: 0x120

Bits	Name	Access	Description	Reset
1:0	SRC_MULTBLK_TYPE	RO	Source Multi Block Transfer Type. These bits define the type of multi-block transfer used for source peripheral. ■ 00: Contiguous ■ 01: Reload ■ 10: Shadow Register ■ 11: Linked List If the type selected is Contiguous, the CHx_SAR register is loaded with the value of the end source address of previous block + 1 at the end of every block for multi-block transfers. A new block transfer is then initiated. If the type selected is Reload, the CHx_SAR register is reloaded from the initial value of SAR at the end of every block for multi-block transfers. A new block transfer is then initiated. If the type selected is Shadow Register, the CHx_SAR register is loaded from the content of its shadow register if CHx_CTL.ShadowReg_Or_LLI_Valid bit is set to 1 at the end of every block for multi-block transfers. A new block transfer is then initiated. If the type selected is Linked List, the CHx_SAR register is	

Bits	Name	Access	Description	Reset
			loaded from the Linked List if CTL.ShadowReg_Or_LLI_Valid bit is set to 1 at the end of every block for multi-block transfers. A new block transfer is then initiated. CHx_CTL and CHx_BLOCK_TS registers are loaded from their initial values or from the contents of their shadow registers (if CHx_CTL.ShadowReg_Or_LLI_Valid bit is set to 1) or from the linked list (if CTL.ShadowReg_Or_LLI_Valid bit is set to 1) at the end of every block for multi-block transfers based on the multi-block transfer type programmed for source and destination peripherals. Contiguous transfer on both source and destination peripheral is not a valid multi-block transfer configuration. This field does not exist if the configuration parameter DMAX_CHx_MULTI_BLK_EN is not selected; in that case, the read-back value is always 0. Values: ■ 0x0 (CONTINGUOUS): Contiguous Multiblock Type used for Source Transfer ■ 0x1 (RELOAD): Reload Multiblock Type used for Source Transfer ■ 0x2 (SHADOW_REGISTER): Shadow Register based Multiblock Type used for Source Transfer ■ 0x3 (LINKED_LIST): Linked List based Multiblock Type used for Source Transfer	
3:2	DST_MULTBLK_TYPE	R/W	Destination Multi Block Transfer Type. These bits define the type of multi-block transfer used for destination peripheral. ■ 00: Contiguous ■ 01: Reload ■ 10: Shadow Register ■ 11: Linked List If the type selected is Contiguous, the CHx_DAR register is loaded with the value of the end source address of previous block + 1 at the end of every block for multi-block transfers. A	0x0

Bits	Name	Access	Description	Reset
			new block transfer is then initiated. If the type selected is Reload, the CHx_DAR register is reloaded from the initial value of DAR at the end of every block for multi-block transfers. A new block transfer is then initiated. If the type selected is Shadow Register, the CHx_DAR register is loaded from the content of its shadow register if CHx_CTL.ShadowReg_Or_LLI_Valid bit is set to 1 at the end of every block for multi-block transfers. A new block transfer is then initiated. If the type selected is Linked List, the CHx_DAR register is loaded from the Linked List if CTL.ShadowReg_Or_LLI_Valid bit is set to 1 at the end of every block for multi-block transfers. A new block transfer is then initiated. CHx_CTL and CHx_BLOCK_TS registers are loaded from their initial values or from the contents of their shadow registers (if CHx_CTL.ShadowReg_Or_LLI_Valid bit is set to 1) or from the linked list (if CTL.ShadowReg_Or_LLI_Valid bit is set to 1) at the end of every block for multi-block transfers based on the multi-block transfer type programmed for source and destination peripherals. Contiguous transfer on both source and destination peripheral is not a valid multi-block transfer configuration. This field does not exist if the configuration parameter DMAX_CHx_MULTI_BLK_EN is not selected; in that case, the read-back value is always 0. Values: ■ 0x0 (CONTINGUOUS): Contiguous Multiblock Type used for Destination Transfer ■ 0x1 (RELOAD): Reload Multiblock Type used for Destination Transfer ■ 0x2 (SHADOW_REGISTER): Shadow Register based	

Bits	Name	Access	Description	Reset
			Multiblock Type used for Destination Transfer ■ 0x3 (LINKED_LIST): Linked List based Multiblock Type used for Destination Transfer	
31:4	Reserved			
34:32	TT_FC	R/W	Transfer Type and Flow Control. The following transfer types are supported. ■ Memory to Memory ■ Memory to Peripheral ■ Peripheral to Memory ■ Peripheral to Peripheral Flow Control can be assigned to the DW_axi_dmac, the source peripheral, or hte destination peripheral. Values: ■ 0x0 (MEM_TO_MEM_DMAC): Transfer Type is memory to memory and Flow Controller is DW_axi_dmac ■ 0x1 (MEM_TO_PER_DMAC): Transfer Type is memory to peripheral and Flow Controller is DW_axi_dmac ■ 0x2 (PER_TO_MEM_DMAC): Transfer Type is peripheral to memory and Flow Controller is DW_axi_dmac ■ 0x3 (PER_TO_PER_DMAC): Transfer Type is peripheral to peripheral and Flow Controller is DW_axi_dmac ■ 0x4 (PER_TO_MEM_SRC): Transfer Type is peripheral to Memory and Flow Controller is Source peripheral ■ 0x5 (PER_TO_PER_SRC): Transfer Type is peripheral to peripheral and Flow Controller is Source peripheral ■ 0x6 (MEM_TO_PER_DST): Transfer Type is memory to peripheral and Flow Controller is Destination peripheral ■ 0x7 (PER_TO_PER_DST): Transfer Type is peripheral to peripheral and Flow Controller is Destination peripheral	0x0
35	HS_SEL_SRC	R/W	Source Software or Hardware Handshaking Select. This register selects which of the handshaking interfaces (hardware or software) is active for source requests on this	0x0

Bits	Name	Access	Description	Reset
			channel. ■ 0: Hardware handshaking interface. Software-initiated transaction requests are ignored. ■ 1: Software handshaking interface. Hardware-initiated transaction requests are ignored. If the source peripheral is memory, then this bit is ignored.	
36	HS_SEL_DST	R/W	Destination Software or Hardware Handshaking Select. This register selects which of the handshaking interfaces (hardware or software) is active for destination requests on this channel. ■ 0: Hardware handshaking interface. Software-initiated transaction requests are ignored. ■ 1: Software handshaking interface. Hardware-initiated transaction requests are ignored. If the destination peripheral is memory, then this bit is ignored.	0x0
37	SRC_HWHS_POL	RO	Source Hardware Handshaking Interface Polarity. ■ 0: ACTIVE HIGH ■ 1: ACTIVE LOW	
38	DST_HWHS_POL	RO	Destination Hardware Handshaking Interface Polarity. ■ 0: ACTIVE HIGH ■ 1: ACTIVE LOW	
39	SRC_PER	R/W	Assigns a hardware handshaking interface (0 - DMAX_NUM_HS_IF-1) to the source of Channelx if the CHx_CFG.HS_SEL_SRC field is 0; otherwise, this field is ignored. The channel can then communicate with the source peripheral connected to that interface through the assigned hardware handshaking interface. Reset Value = 1 Note: For correct DW_axi_dmac operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.	0x0
43:40	Reserved			
44	DST_PER	R/W	Assigns a hardware handshaking interface (0 - DMAX_NUM_HS_IF-1) to the destination of Channelx if the CHx_CFG.HS_SEL_DST field is 0; otherwise, this field is	0x0

Bits	Name	Access	Description	Reset
			ignored. The channel can then communicate with the destination peripheral connected to that interface through the assigned hardware handshaking interface. Reset Value = 1 Note: For correct DW_axi_dmac operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.	
48:45	Reserved			
51:49	CH_PRIOR	R/W	Channel Priority A priority of 7 is the highest priority, and 0 is the lowest. This field must be programmed within the following range: 0: DMAX_NUM_CHANNELS-1 A programmed value outside this range will cause erroneous behavior.	0x0
52	LOCK_CH	R/W	Channel Lock bit When the channel is granted control of the master bus interface and if the CHx_CFG.LOCK_CH bit is asserted, then no other channels are granted control of the master bus interface for the duration specified in CHx_CFG.LOCK_CH_L. Indicates to the master bus interface arbiter that this channel wants exclusive access to the master bus interface for the duration specified in CHx_CFG.LOCK_CH_L. This field does not exist if the configuration parameter DMAX_CHx_LOCK_EN is set to False; in this case, the read-back value is always 0. Locking the channel locks AXI Read Address, Write Address and Write Data channels on the corresponding master interface. Note: Channel locking feature is supported only for memoryto-memory transfer at Block Transfer and DMA Transfer levels. Hardware does not check for the validity of channel locking setting, hence the software must take care of enabling the channel locking only for memory-to-memory	0x0

Bits	Name	Access	Description	Reset
			transfers at Block Transfer or DMA Transfer levels. Illegal programming of channel locking might result in unpredictable behavior.	
54:53	LOCK_CH_L	R/W	Channel Lock Level This bit indicates the duration over which CHx_CFG.LOCK_CH bit applies. ■ 00: Over complete DMA transfer ■ 01: Over DMA block transfer ■ 1x: Reserved This field does not exist if the configuration parameter DMAX_CHx_LOCK_EN is set to False; in that case, the read-back value is always 0.	0x0
58:55	SRC_OSR_LMT	R/W	Source Outstanding Request Limit ■ Maximum outstanding request supported is 16. ■ Source Outstanding Request Limit = SRC_OSR_LMT + 1	0x0
62:59	DST_OSR_LMT	R/W	Destination Outstanding Request Limit ■ Maximum outstanding request supported is 16. ■ Source Outstanding Request Limit = DST_OSR_LMT + 1	0x0
63	Reserved			

CHx_LLP

Offset Address: 0x128

Bits	Name	Access	Description	Reset
0	LMS	R/W	LLI master Select This bit identifies the AXI layer/interface where the memory device that stores the next linked list item resides. - 0: AXI Master 1 ■ 1: AXI Master 2 This field does not exist if the configuration parameter DMAX_CHx_LMS is not set to NO_HARDCODE.	0x0
5:1	Reserved			
63:6	LOC	R/W	Starting Address Memory of LLI block Starting Address In Memory of next LLI if block chaining is enabled. The six LSBs of the starting address are not stored because the address is assumed to be aligned to a 64-byte boundary. LLI access always uses the burst size (arsize/awsize) that is same as the data bus width and cannot	0x0

Bits	Name	Access	Description	Reset
			be changed or programmed to anything other than this. Burst length (awlen/arlen) is chosen based on the data bus width so that the access does not cross one complete LLI structure of 64 bytes. DW_axi_dmac will fetch the entire LLI (40 bytes) in one AXI burst if the burst length is not limited by other settings.	

CHx_STATUSREG

Offset Address: 0x130

Bits	Name	Access	Description	Reset
21:0	CMPLTD_BLK_TFR_SIZE	RO	Completed Block Transfer Size. This bit indicates the total number of data of width CHx_CTL.SRC_TR_WIDTH transferred for the previous block transfer.	
31:22	Reserved			
46:32	DATA_LEFT_IN_FIFO	RO	Data Left in FIFO. This bit indicates the total number of data left in DW_axi_dmac channel FIFO after completing the current block transfer.	
63:47	Reserved			

CHx_SWHSSRCREG

Offset Address: 0x138

Bits	Name	Access	Description	Reset
0	SWHS_REQ_SRC	R/W	Software Handshake Request for Channel Source. This bit is used to request dma source data transfer if software handshaking method is selected for the source of the corresponding channel. This bit is ignored if software handshaking is not enabled for the source of the Channelx. The functionality of this field depends on whether the peripheral is the flow controller or not.	0x0
1	SWHS_REQ_SRC_WE	WO	Write Enable bit for Software Handshake Request for Channel Source.	0x0
2	SWHS_SGLREQ_SRC	R/W	Software Handshake Single Request for Channel Source. This bit is used to request SINGLE (AXI	0x0

Bits	Name	Access	Description	Reset
			burst length = 1) dma source data transfer if software handshaking method is selected for the source of the corresponding channel. This bit is ignored if software handshaking is not enabled for the source of the Channelx. The functionality of this field depends on whether the peripheral is the flow controller.	
3	SWHS_SGLREQ_SRC_WE	WO	Write Enable bit for Software Handshake Single Request for Channel Source.	0x0
4	SWHS_LST_SRC	R/W	Software Handshake Last Request for Channel Source. This bit is used to request LAST dma source data transfer if software handshaking method is selected for the source of the corresponding channel. This bit is ignored if software handshaking is not enabled for the source of the Channelx or if the source of Channelx is not the flow controller.	0x0
5	SWHS_LST_SRC_WE	WO	Write Enable bit for Software Handshake Last Request for Channel Source.	0x0
31:6	Reserved			

CHx_SWHSDSTREG

Offset Address: 0x140

Bits	Name	Access	Description	Reset
0	SWHS_REQ_DST	R/W	Software Handshake Request for Channel Destination. This bit is used to request dma destination data transfer if software handshaking method is selected for the destination of the corresponding channel.	0x0
1	SWHS_REQ_DST_WE	WO	Write Enable bit for Software Handshake Request for Channel Destination.	0x0
2	SWHS_SGLREQ_DST	R/W	Software Handshake Single Request for Channel Destination. This bit is used to request SINGLE (AXI burst length = 1) dma destination data transfer if software handshaking method is selected for the destination of the corresponding channel.	0x0
3	SWHS_SGLREQ_DST_WE	WO	Write Enable bit for Software Handshake Single Request for	0x0

Bits	Name	Access	Description	Reset
			Channel Destination.	
4	SWHS_LST_DST	R/W	Software Handshake Last Request for Channel Destination. This bit is used to request LAST dma destination data transfer if software handshaking method is selected for the destination of the corresponding channel.	0x0
5	SWHS_LST_DST_WE	WO	Write Enable bit for Software Handshake Last Request for Channel Destination.	0x0
31:6	Reserved			

CHx_BLK_TFR_RESUMEREQREG

Offset Address: 0x148

Bits	Name	Access	Description	Reset
0	BLK_TFR_RESUMEREQ	WO	Block Transfer Resume Request during Linked-List or Shadow-Register-based multi-block transfer.	0x0
31:1	Reserved			

CHx_AXI_IDREG

Offset Address: 0x150

Bits	Name	Access	Description	Reset
14:0	AXI_READ_ID_SUFFIX	R/W	AXI Read ID Suffix These bits form part of the ARID output of AXI3/AXI4 master interface.	0x0
15	Reserved			
30:16	AXI_WRITE_ID_SUFFIX	R/W	AXI Write ID Suffix. These bits form part of the AWID output of AXI3/AXI4 master interface.	0x0
31	Reserved			

CHx_AXI_QOSREG

Offset Address: 0x158

Bits	Name	Access	Description	Reset
3:0	AXI_AWQOS	R/W	AXI AWQOS. These bits form the awqos output of AXI4 master interface.	0x0
7:4	AXI_ARQOS	R/W	AXI ARQOS. These bits form the arqos output of AXI4 master interface.	0x0
31:8	Reserved			

CHx_SSTAT

Offset Address: 0x160

Bits	Name	Access	Description	Reset

Bits	Name	Access	Description	Reset
31:0	SSTAT	RO	Source Status Source status information retrieved by hardware from the address pointed to by the contents of the CHx_SSTATAR register.	

CHx_DSTAT

Offset Address: 0x168

Bits	Name	Access	Description	Reset
31:0	DSTAT	RO	Destination Status Destination status information retrieved by hardware from the address pointed to by the contents of the CHx_DSTATAR register.	

CHx_SSTATAR

Offset Address: 0x170

Bits	Name	Access	Description	Reset
63:0	SSTATAR	R/W	Source Status Fetch Address Pointer from where hardware can fetch the source status information, which is registered in the CHx_SSTAT register and written out to the CHx_SSTAT register location of the LLI before the start of the next block if DMAX_CHx_LLI_WB_EN = 1 and linked list based multi-block transfer is enabled for either source or destination peripheral of the channel. Source peripheral should update the source status information, if any, at the location pointed to by CHx_SSTATAR to utilize this feature.	0x0

CHx_DSTATAR

Offset Address: 0x178

Bits	Name	Access	Description	Reset
63:0	DSTATAR	R/W	Destination Status Fetch Address Pointer from where hardware can fetch the Destination status information, which is registered in the CHx_DSTAT register and written out to the CHx_DSTAT register location of the LLI before the start of the next block if DMAX_CHx_LLI_WB_EN = 1 and linked list based multiblock	0x0

Bits	Name	Access	Description	Reset
			transfer is enabled for either source or destination peripheral of the channel. Destination peripheral should update the destination status information, if any, at the location pointed to by CHx_DSTATAR to utilize this feature.	

CHx_INTSTATUS_ENABLEREG

Offset Address: 0x180

Bits	Name	Access	Description	Reset
0	Enable_BLOCK_TFR_DONE_IntStat	R/W	Block Transfer Done Interrupt Status Enable. 0: Disable the generation of Block Transfer Done Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Block Transfer Done Interrupt in CHx_INTSTATUSREG	0x0
1	Enable_DMA_TFR_DONE_IntStat	R/W	DMA Transfer Done Interrupt Status Enable. 0: Disable the generation of DMA Transfer Done Interrupt in CHx_INTSTATUSREG 1: Enable the generation of DMA Transfer Done Interrupt in CHx_INTSTATUSREG	0x0
2	Reserved			
3	Enable_SRC_TRANSCOMP_IntStat	R/W	Source Transaction Completed Status Enable. 0: Disable the generation of Source Transaction Complete Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Source Transaction Complete Interrupt in CHx_INTSTATUSREG	0x0
4	Enable_DST_TRANSCOMP_IntStat	R/W	Destination Transaction Completed Status Enable. 0: Disable the generation of Destination Transaction complete Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Destination Transaction complete Interrupt in CHx_INTSTATUSREG	0x0
5	Enable_SRC_DEC_ERR_IntStat	R/W	Source Decode Error Status Enable. 0: Disable the generation of Source Decode Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Source Decode Error Interrupt in CHx_INTSTATUSREG	0x0
6	Enable_DST_DEC_ERR_IntStat	R/W	Destination Decode Error Status Enable.	0x0

Bits	Name	Access	Description	Reset
			0: Disable the generation of Destination Decode Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Destination Decode Error Interrupt in CHx_INTSTATUSREG	
7	Enable_SRC_SLV_ERR_IntStat	R/W	Source Slave Error Status Enable. 0: Disable the generation of Source Slave Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Source Slave Error Interrupt in CHx_INTSTATUSREG	0x0
8	Enable_DST_SLV_ERR_IntStat	R/W	Destination Slave Error Status Enable. 0: Disable the generation of Destination Slave Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Destination Slave Error Interrupt in CHx_INTSTATUSREG	0x0
9	Enable_LLI_RD_DEC_ERR_IntStat	R/W	LLI Read Decode Error Status Enable. 0: Disable the generation of LLI Read Decode Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of LLI Read Decode Error Interrupt in CHx_INTSTATUSREG	0x0
10	Enable_LLI_WR_DEC_ERR_IntStat	R/W	LLI WRITE Decode Error Status Enable. 0: Disable the generation of LLI WRITE Decode Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of LLI WRITE Decode Error Interrupt in CHx_INTSTATUSREG	0x0
11	Enable_LLI_RD_SLV_ERR_IntStat	R/W	LLI Read Slave Error Status Enable. 0: Disable the generation of LLI Read Slave Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of LLI Read Slave Error Interrupt in CHx_INTSTATUSREG	0x0
12	Enable_LLI_WR_SLV_ERR_IntStat	R/W	LLI WRITE Slave Error Status Enable. 0: Disable the generation of LLI WRITE Slave Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of LLI WRITE Slave Error Interrupt in CHx_INTSTATUSREG	0x0
13	Enable_SHADOWREG_OR_LLI_INVALID_ERR_IntStat	R/W	Shadow register or LLI Invalid Error Status Enable. 0: Disable the generation of Shadow Register or LLI Invalid Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Shadow Register or LLI Invalid Error Interrupt in	0x0

Bits	Name	Access	Description	Reset
			CHx_INTSTATUSREG	
14	Enable_SLVIF_MULTIBLKTYPE_ERR_IntStat	R/W	Slave Interface Multi Block type Error Status Enable. 0: Disable the generation of Slave Interface Multi Block type Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Slave Interface Multi Block type Error Interrupt in CHx_INTSTATUSREG	0x0
15	Reserved			
16	Enable_SLVIF_DEC_ERR_IntStat	R/W	Slave Interface Decode Error Status Enable. 0: Disable the generation of Slave Interface Decode Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Slave Interface Decode Error Interrupt in CHx_INTSTATUSREG	0x0
17	Enable_SLVIF_WR2RO_ERR_IntStat	R/W	Slave Interface Write to Read Only Error Status Enable. 0: Disable the generation of Slave Interface Write to Read only Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Slave Interface Write to Read Only Error Interrupt in CHx_INTSTATUSREG	0x0
18	Enable_SLVIF_RD2RWO_ERR_IntStat	R/W	Slave Interface Read to write Only Error Status Enable. 0: Disable the generation of Slave Interface Read to Write only Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Slave Interface Read to Write Only Error Interrupt in CHx_INTSTATUSREG	0x0
19	Enable_SLVIF_WRONCHEN_ERR_IntStat	R/W	Slave Interface Write On Channel Enabled Error Status Enable. 0: Disable the generation of Slave Interface Write On Channel enabled Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Slave Interface Write On Channel enabled Error Interrupt in CHx_INTSTATUSREG	0x0
20	Enable_SLVIF_SHADOWREG_WRON_VALID_ERR_IntStat	R/W	Shadow Register Write On Valid Error Status Enable. 0: Disable the generation of Shadow Register Write On Valid Error Interrupt in CHx_INTSTATUSREG	0x0

Bits	Name	Access	Description	Reset
			1: Enable the generation of Shadow register Write On Valid Error Interrupt in CHx_INTSTATUSREG	
21	Enable_SLVIF_WRONHOLD_ERR_IntStat	R/W	Slave Interface Write On Hold Error Status Enable. 0: Disable the generation of Slave Interface Write On Hold Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Slave Interface Write On Hold Error Interrupt in CHx_INTSTATUSREG	0x0
26:22	Reserved			
27	Enable_CH_LOCK_CLEARED_IntStat	R/W	Channel Lock Cleared Status Enable. 0: Disable the generation of Channel LOCK CLEARED Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Channel LOCK CLEARED Interrupt in CHx_INTSTATUSREG	0x0
28	Enable_CH_SRC_SUSPENDED_IntStat	R/W	Channel Source Suspended Status Enable. 0: Disable the generation of Channel Source Suspended Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Channel Source Suspended Interrupt in CHx_INTSTATUSREG	0x0
29	Enable_CH_SUSPENDED_IntStat	R/W	Channel Suspended Status Enable. 0: Disable the generation of Channel Suspended Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Channel Suspended Interrupt in CHx_INTSTATUSREG	0x0
30	Enable_CH_DISABLED_IntStat	R/W	Channel Disabled Status Enable. 0: Disable the generation of Channel Disabled Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Channel Disabled Interrupt in CHx_INTSTATUSREG	0x0
31	Enable_CH_ABORTED_IntStat	R/W	Channel Aborted Status Enable. 0: Disable the generation of Channel Aborted Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Channel Aborted Interrupt in CHx_INTSTATUSREG	0x0

CHx_INTSTATUS

Offset Address: 0x188

Bits	Name	Access	Description	Reset
0	BLOCK_TFR_DONE_IntStat	RO	Block Transfer Done. This indicates to the software that the	

Bits	Name	Access	Description	Reset
			DW_axi_dmac has completed the requested block transfer. The DW_axi_dmac sets this bit to 1 when the transfer is successfully completed. 0: Block Transfer not completed. 1: Block Transfer completed. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register.	
1	DMA_TFR_DONE_IntStat	RO	DMA Transfer Done. This indicates to the software that the DW_axi_dmac has completed the requested DMA transfer. The DW_axi_dmac sets this bit to 1 along with setting CHx_INTSTATUS.BLOCK_TFR_DONE bit to 1 when the last block transfer is completed. 0: DMA Transfer not completed. 1: DMA Transfer Completed This bit is cleared to 0 on writing 1	
2	Reserved			
3	SRC_TRANSCOMP_IntStat	RO	Source Transaction Completed. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register or on enabling the channel (needed when interrupt is not enabled.	
4	DST_TRANSCOMP_IntStat	RO	Destination Transaction Completed. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register or on enabling the channel (needed when interrupt is not enabled.	
5	SRC_DEC_ERR_IntStat	RO	Source Decode Error. Decode Error detected by Master Interface during source data transfer. This error occurs if the access is to invalid address and a Decode Error is returned from interconnect/slave. This error condition causes the DW_axi_dmac to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN bit corresponding to the channel which received the error is set to 0. 0: No Source Decode Errors. 1: Source Decode Error detected.	
6	DST_DEC_ERR_IntStat	RO	Destination Decode Error. Decode Error detected by Master	

Bits	Name	Access	Description	Reset
			Interface during destination data transfer. This error occurs if the access is to invalid address and a Decode Error is returned from interconnect/slave. This error condition causes the DW_axi_dmac to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN bit corresponding to the channel which received the error is set to 0. 0: No destination Decode Errors. 1: Destination Decode Error Detected	
7	SRC_SLV_ERR_IntStat	RO	Source Slave Error. Slave Error detected by Master Interface during source data transfer. This error occurs if the slave interface from which the data is read issues a Slave Error. This error condition causes the DW_axi_dmac to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN bit corresponding to the channel which received the error is set to 0. 0: No Source Slave Errors 1: Source Slave Error Detected	
8	DST_SLV_ERR_IntStat	RO	Destination Slave Error. Slave Error detected by Master Interface during destination data transfer. This error occurs if the slave interface to which the data is written issues a Slave Error. This error condition causes the DW_axi_dmac to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN bit corresponding to the channel which received the error is set to 0. 0: No Destination Slave Errors 1: Destination Slave Errors Detected	
9	LLI_RD_DEC_ERR_IntStat	RO	LLI Read Decode Error. Decode Error detected by Master Interface during LLI read operation. This error occurs if the access is to invalid address and a Decode Error is returned from interconnect/slave. This error condition causes the DW_axi_dmac to disable the	

Bits	Name	Access	Description	Reset
			corresponding channel gracefully; the DMAC_ChEnReg.CH_EN1 bit which received the error is set to 0. 0: NO LLI Read Decode Errors. 1: LLI Read Decode Error detected	
10	LLI_WR_DEC_ERR_IntStat	RO	LLI WRITE Decode Error. Decode Error detected by Master Interface during LLI writeback operation. This error occurs if the access is to invalid address and a Decode Error is returned from interconnect/slave. This error condition causes the DW_axi_dmac to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN1 bit which received the error is set to 0. 0: NO LLI Write Decode Errors. 1: LLI write Decode Error detected.	
11	LLI_RD_SLV_ERR_IntStat	RO	LLI Read Slave Error. Slave Error detected by Master Interface during LLI read operation. This error occurs if the slave interface on which LLI resides issues a Slave Error. This error condition causes the DW_axi_dmac to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN1 bit which received the error is set to 0. 0: No LLI Read Slave Errors. 1: LLI read Slave Error detected.	
12	LLI_WR_SLV_ERR_IntStat	RO	LLI WRITE Slave Error. Slave Error detected by Master Interface during LLI writeback operation. This error occurs if the slave interface on which LLI resides issues a Slave Error. This error condition causes the DW_axi_dmac to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN1 bit which received the error is set to 0. 0: No LLI write Slave Errors. 1: LLI Write SLAVE Error detected.	
13	SHADOWREG_OR_LLI_INVALID_ERR_IntStat	RO	Shadow register or LLI Invalid Error. This error occurs if CHx_CTL.ShadowReg_Or_LLI_Valid bit is seen to be 0 during DW_axi_dmac Shadow Register / LLI fetch phase. This error condition causes the DW_axi_dmac	

Bits	Name	Access	Description	Reset
			to halt the corresponding channel gracefully; Error Interrupt is generated if the corresponding channel error interrupt mask bit is set to 0 and the channel waits till software writes (any value) to CHx_BLK_TFR_ResumeReqReg to indicate valid Shadow Register availability. In the case of LLI pre-fetching, ShadowReg_Or_LLI_Invalid_ERR Interrupt is not generated even if ShadowReg_Or_LLI_Valid bit is seen to be 0 for the pre-fetched LLI. In this case, DW_axi_dmac re-attempts the LLI fetch operation after completing the current block transfer and generates ShadowReg_Or_LLI_Invalid_ERR Interrupt only if ShadowReg_Or_LLI_Valid bit is still seen to be 0. 0: No Shadow Register / LLI Invalid errors. 1: Shadow Register / LLI Invalid error detected.	
14	SLVIF_MULTIBLKTYPE_ERR_IntStat	RO	Slave Interface Multi Block type Error. This error occurs if multi-block transfer type programmed in CHx_CFG register (SRC_MLTBLK_TYPE and DST_MLTBLK_TYPE) is invalid. This error condition causes the DW_axi_dmac to halt the corresponding channel gracefully; Error Interrupt is generated if the corresponding channel error interrupt mask bit is set to 0 and the channel waits till software writes (any value) to CHx_BLK_TFR_ResumeReqReg to indicate valid multiblock transfer type availability. 0: No Multi-block transfer type Errors. 1: Multi-block transfer type Error detected.	
15	Reserved			
16	SLVIF_DEC_ERR_IntStat	RO	Slave Interface Decode Error. Decode Error generated by DW_axi_dmac during register access. This error occurs if the register access is to invalid address in Channelx register space resulting in error response by DW_axi_dmac slave	

Bits	Name	Access	Description	Reset
			interface. 0: No Slave Interface Decode errors. 1: Slave Interface Decode Error detected.	
17	SLVIF_WR2RO_ERR_IntStat	RO	Slave Interface Write to Read Only Error. This error occurs if write operation is performed to a Read Only register. 0: No Slave Interface Write to Read Only Errors. 1: Slave Interface Write to Read Only Error detected.	
18	SLVIF_RD2RWO_ERR_IntStat	RO	Slave Interface Read to write Only Error. This error occurs if read operation is performed to a Write Only register. 0: No Slave Interface Read to Write Only Errors. 1: Slave Interface Read to Write Only Error detected.	
19	SLVIF_WRONCHEN_ERR_IntStat	RO	Slave Interface Write On Channel Enabled Error. This error occurs if an illegal write operation is performed on a register; this happens if a write operation is performed on a register when the channel is enabled and if it is not allowed for the corresponding register as per the DW_axi_dmac specification. 0: No Slave Interface Write On Channel Enabled Errors. 1: Slave Interface Write On Channel Enabled Error detected.	
20	SLVIF_SHADOWREG_WRON_VALID_ERR_IntStat	RO	Shadow Register Write On Valid Error. This error occurs if shadow register based multi-block transfer is enabled and software tries to write to the shadow register when CHx_CTL.ShadowReg_Or_LLI_Valid bit is 1. 0: No Slave Interface Shadow Register Write On Valid Errors. 1: Slave Interface Shadow Register Write On Valid Error detected.	
21	SLVIF_WRONHOLD_ERR_IntStat	RO	Slave Interface Write On Hold Error. This error occurs if an illegal write operation is performed on a register; this happens if a write operation is performed on a channel register when DW_axi_dmac is in Hold mode.	

Bits	Name	Access	Description	Reset
			0: No Slave Interface Write On Hold Errors. 1: Slave Interface Write On Hold Error detected.	
26:22	Reserved			
27	CH_LOCK_CLEARED_IntStat	RO	Channel Lock Cleared. This indicates to the software that the locking of the corresponding channel in DW_axi_dmac is cleared. 0: Channel locking is not cleared. 1: Channel locking is cleared.	
28	CH_SRC_SUSPENDED_IntStat	RO	Channel Source Suspended. This indicates to the software that the corresponding channel source data transfer in DW_axi_dmac is suspended. 0: Channel source is not suspended 1: Channel Source is suspended.	
29	CH_SUSPENDED_IntStat	RO	Channel Suspended. This indicates to the software that the corresponding channel in DW_axi_dmac is suspended. 0: Channel is not suspended. 1: Channel is suspended.	
30	CH_DISABLED_IntStat	RO	Channel Disabled. This indicates to the software that the corresponding channel in DW_axi_dmac is disabled. 0: Channel is not disabled. 1: Channel is disabled. Error Interrupt is generated if the corresponding bit in CHx_INTSTATUS_ENABLEReg is enabled.	
31	CH_ABORTED_IntStat	RO	Channel Aborted. This indicates to the software that the corresponding channel in DW_axi_dmac is aborted. 0: Channel is not aborted 1: Channel is aborted	

CHx_INTSIGNAL_ENABLEREG

Offset Address: 0x190

Bits	Name	Access	Description	Reset
0	Enable_BLOCK_TFR_DONE_IntSignal	R/W	Block Transfer Done Interrupt Signal Enable. 0: Disable the propagation of Block Transfer Done Interrupt to generate a port level interrupt 1: Enable the propagation of Block Transfer Done Interrupt to generate a port level interrupt	0x0
1	Enable_DMA_TFR_DONE_IntSignal	R/W	DMA Transfer Done Interrupt Signal	0x0

Bits	Name	Access	Description	Reset
			Enable. 0: Disable the propagation of DMA Transfer Done Interrupt to generate a port level interrupt 1: Enable the propagation of DMA Transfer Done Interrupt to generate a port level interrupt	
2	Reserved			
3	Enable_SRC_TRANSCOMP_IntSignal	R/W	Source Transaction Completed Signal Enable. 0: Disable the propagation of Source Transaction Complete Interrupt to generate a port level interrupt 1: Enable the propagation of Source Transaction Complete Interrupt to generate a port level interrupt	0x0
4	Enable_DST_TRANSCOMP_IntSignal	R/W	Destination Transaction Completed Signal Enable. 0: Disable the propagation of Destination Transaction complete Interrupt to generate a port level interrupt 1: Enable the propagation of Destination Transaction complete Interrupt to generate a port level interrupt	0x0
5	Enable_SRC_DEC_ERR_IntSignal	R/W	Source Decode Error Signal Enable. 0: Disable the propagation of Source Decode Error Interrupt to generate a port level interrupt 1: Enable the propagation of Source Decode Error Interrupt to generate a port level interrupt	0x0
6	Enable_DST_DEC_ERR_IntSignal	R/W	Destination Decode Error Signal Enable. 0: Disable the propagation of Destination Decode Error Interrupt to generate a port level interrupt 1: Enable the propagation of Destination Decode Error Interrupt to generate a port level interrupt	0x0
7	Enable_SRC_SLV_ERR_IntSignal	R/W	Source Slave Error Signal Enable. 0: Disable the propagation of Source Slave Error Interrupt to generate a port level interrupt 1: Enable the propagation of Source Slave Error Interrupt to generate a port level interrupt	0x0
8	Enable_DST_SLV_ERR_IntSignal	R/W	Destination Slave Error Signal Enable. 0: Disable the propagation of	0x0

Bits	Name	Access	Description	Reset
			Destination Slave Error Interrupt to generate a port level interrupt 1: Enable the propagation of Destination Slave Error Interrupt to generate a port level interrupt	
9	Enable_LLI_RD_DEC_ERR_IntSignal	R/W	LLI Read Decode Error Signal Enable. 0: Disable the propagation of LLI Read Decode Error Interrupt to generate a port level interrupt 1: Enable the propagation of LLI Read Decode Error Interrupt to generate a port level interrupt	0x0
10	Enable_LLI_WR_DEC_ERR_IntSignal	R/W	LLI WRITE Decode Error Signal Enable. 0: Disable the propagation of LLI WRITE Decode Error Interrupt to generate a port level interrupt 1: Enable the propagation of LLI WRITE Decode Error Interrupt to generate a port level interrupt	0x0
11	Enable_LLI_RD_SLV_ERR_IntSignal	R/W	LLI Read Slave Error Signal Enable. 0: Disable the propagation of LLI Read Slave Error Interrupt to generate a port level interrupt 1: Enable the propagation of LLI Read Slave Error Interrupt to generate a port level interrupt	0x0
12	Enable_LLI_WR_SLV_ERR_IntSignal	R/W	LLI WRITE Slave Error Signal Enable. 0: Disable the propagation of LLI WRITE Slave Error Interrupt to generate a port level interrupt 1: Enable the propagation of LLI WRITE Slave Error Interrupt to generate a port level interrupt	0x0
13	Enable_SHADOWREG_OR_LLI_INVALID_ERR_IntSignal	R/W	Shadow register or LLI Invalid Error Signal Enable. 0: Disable the propagation of Shadow Register or LLI Invalid Error Interrupt to generate a port level interrupt 1: Enable the propagation of Shadow Register or LLI Invalid Error Interrupt to generate a port level interrupt	0x0
14	Enable_SLVIF_MULTIBLKTYPE_ERR_IntSignal	R/W	Slave Interface Multi Block type Error Signal Enable. 0: Disable the propagation of Slave Interface Multi Block	0x0

Bits	Name	Access	Description	Reset
			type Error Interrupt to generate a port level interrupt 1: Enable the propagation of Slave Interface Multi Block type Error Interrupt to generate a port level interrupt	
15	Reserved			
16	Enable_SLVIF_DEC_ERR_IntSignal	R/W	Slave Interface Decode Error Signal Enable. 0: Disable the propagation of Slave Interface Decode Error Interrupt to generate a port level interrupt 1: Enable the propagation of Slave Interface Decode Error Interrupt to generate a port level interrupt	0x0
17	Enable_SLVIF_WR2RO_ERR_IntSignal	R/W	Slave Interface Write to Read Only Error Signal Enable. 0: Disable the propagation of Slave Interface Write to Read only Error Interrupt to generate a port level interrupt 1: Enable the propagation of Slave Interface Write to Read Only Error Interrupt to generate a port level interrupt	0x0
18	Enable_SLVIF_RD2RWO_ERR_IntSignal	R/W	Slave Interface Read to write Only Error Signal Enable. 0: Disable the propagation of Slave Interface Read to Write only Error Interrupt to generate a port level interrupt 1: Enable the propagation of Slave Interface Read to Write Only Error Interrupt to generate a port level interrupt	0x0
19	Enable_SLVIF_WRONCHEN_ERR_IntSignal	R/W	Slave Interface Write On Channel Enabled Error Signal Enable. 0: Disable the propagation of Slave Interface Write On Channel enabled Error Interrupt to generate a port level interrupt 1: Enable the propagation of Slave Interface Write On Channel enabled Error Interrupt to generate a port level interrupt	0x0
20	Enable_SLVIF_SHADOWREG_WRON_VALID_ERR_IntSignal	R/W	Shadow Register Write On Valid Error Signal Enable. 0: Disable the propagation of Shadow Register Write On Valid Error Interrupt to generate a port	0x0

Bits	Name	Access	Description	Reset
			level interrupt 1: Enable the propagation of Shadow register Write On Valid Error Interrupt to generate a port level interrupt	
21	Enable_SLVIF_WRONHOLD_ERR_IntSignal	R/W	Slave Interface Write On Hold Error Signal Enable. 0: Disable the propagation of Slave Interface Write On Hold Error Interrupt to generate a port level interrupt 1: Enable the propagation of Slave Interface Write On Hold Error Interrupt to generate a port level interrupt	0x0
26:22	Reserved			
27	Enable_CH_LOCK_CLEARED_IntSignal	R/W	Channel Lock Cleared Signal Enable. 0: Disable the propagation of Channel Lock Cleared Interrupt to generate a port level interrupt 1: Enable the propagation of Channel Lock Cleared Interrupt to generate a port level interrupt	0x0
28	Enable_CH_SRC_SUSPENDED_IntSignal	R/W	Channel Source Suspended Signal Enable. 0: Disable the propagation of Channel Source Suspended Interrupt to generate a port level interrupt 1: Enable the propagation of Channel Source Suspended Interrupt to generate a port level interrupt	0x0
29	Enable_CH_SUSPENDED_IntSignal	R/W	Channel Suspended Signal Enable. 0: Disable the propagation of Channel Suspended Interrupt to generate a port level interrupt 1: Enable the propagation of Channel Suspended Interrupt to generate a port level interrupt	0x0
30	Enable_CH_DISABLED_IntSignal	R/W	Channel Disabled Signal Enable. 0: Disable the propagation of Channel Disabled Interrupt to generate a port level interrupt 1: Enable the propagation of Channel Disabled Interrupt to generate a port level interrupt	0x0
31	Enable_CH_ABORTED_IntSignal	R/W	Channel Aborted Signal Enable. 0: Disable the propagation of Channel Aborted Interrupt to generate a port level interrupt 1: Enable the propagation of Channel Aborted Interrupt to	0x0

Bits	Name	Access	Description	Reset
			generate a port level interrupt	

CHx_INTCLEARREG

Offset Address: 0x198

Bits	Name	Access	Description	Reset
0	Clear_BLOCK_TFR_DONE_IntStat	WO	Block Transfer Done Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CH1_INTSTATUSREG	0x0
1	Clear_DMA_TFR_DONE_IntStat	WO	DMA Transfer Done Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
2	Reserved			
3	Clear_SRC_TRANSCOMP_IntStat	WO	Source Transaction Completed Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
4	Clear_DST_TRANSCOMP_IntStat	WO	Destination Transaction Completed Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
5	Clear_SRC_DEC_ERR_IntStat	WO	Source Decode Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
6	Clear_DST_DEC_ERR_IntStat	WO	Destination Decode Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
7	Clear_SRC_SLV_ERR_IntStat	WO	Source Slave Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
8	Clear_DST_SLV_ERR_IntStat	WO	Destination Slave Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
9	Clear_LLI_RD_DEC_ERR_IntStat	WO	LLI Read Decode Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
10	Clear_LLI_WR_DEC_ERR_IntStat	WO	LLI WRITE Decode Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
11	Clear_LLI_RD_SLV_ERR_IntStat	WO	LLI Read Slave Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt	0x0

Bits	Name	Access	Description	Reset
			status bit in CHx_INTSTATUSREG.	
12	Clear_LLI_WR_SLV_ERR_IntStat	WO	LLI WRITE Slave Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
13	Clear_SHADOWREG_OR_LLI_INVALID_ERR_IntStat	WO	Shadow register or LLI Invalid Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
14	Clear_SLVIF_MULTIBLKTYPE_ERR_IntStat	WO	Slave Interface Multi Block type Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
15	Reserved			
16	Clear_SLVIF_DEC_ERR_IntStat	WO	Slave Interface Decode Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
17	Clear_SLVIF_WR2RO_ERR_IntStat	WO	Slave Interface Write to Read Only Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
18	Clear_SLVIF_RD2RWO_ERR_IntStat	WO	Slave Interface Read to write Only Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
19	Clear_SLVIF_WRONCHEN_ERR_IntStat	WO	Slave Interface Write On Channel Enabled Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
20	Clear_SLVIF_SHADOWREG_WRON_VALID_ERR_IntStat	WO	Shadow Register Write On Valid Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
21	Clear_SLVIF_WRONHOLD_ERR_IntStat	WO	Slave Interface Write On Hold Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
26:22	Reserved			
27	Clear_CH_LOCK_CLEARED_IntStat	WO	Channel Lock Cleared Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
28	Clear_CH_SRC_SUSPENDED_IntStat	WO	Channel Source Suspended Interrupt Clear Bit. This bit is used to clear the	0x0

Bits	Name	Access	Description	Reset
			corresponding channel interrupt status bit in CHx_INTSTATUSREG.	
29	Clear_CH_SUSPENDED_IntStat	WO	Channel Suspended Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
30	Clear_CH_DISABLED_IntStat	WO	Channel Disabled Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
31	Reserved			

3.7 Timer

3.7.1 Overview

The system is equipped with 8 timer modules. It can be used as a timing or counting function, which can provide application program for timing and counting, and can also provide operating system for implementing system clock.

3.7.2 Characteristics

The timer has the following characteristics:

- 32bit count down timer / counter.
- Support two counting modes: free running mode and user-defined counting mode
- The system can read the current counter value.

When the counter value decreases to 0, an interrupt is generated.

3.7.3 Function description

Timer is based on a 32 bit count down counter. The value of the counter is subtracted by 1 on each rising edge of the counting clock. When the counter is counting down to zero, the timer generates an interrupt.

Timer has the following two counting modes:

- Free running mode

  The timer counts continuously. When the counter value is reduced to 0, it automatically turns back to its maximum value and continues to count down. The maximum initial counter value is 0xFFFF_ FFFF。
- User-defined counting mode

  The timer continues to count. When the counter value is reduced to 0, the timer reloads the initial counter value from the TimerNLoadCount (N = 1 ~ 8) register and continues to count down.

The method of loading the initial counter value to the timer is as follows：

By writing TimerNLoadCount (N = 1 ~ 8) register, the initial counter value of timer can

be loaded.

3.7.4 Operation mode

3.7.4.1 Initialization

Step 1 Write TimerNLoadCount (N = 1 ~ 8) register to load the initial counter value for timer.
Step 2 Set TimerNControlReg [2:0] (N = 1 ~ 8) register, select timer counting mode, mask timer interrupt and enable timer to start counting down.

3.7.4.2 Interupt processing

When the timer generates an interrupt, the operation steps are as follows:
Step 1 read TimerNEOI (N = 1 ~ 8) register and clear timerN interrupt.
Step 2 executes the process waiting for the interrupt.
Step 3 After the process of is completed, the interrupted program is resumed.

3.7.4.3 Clock selection

The system timer can use either a 25MHz or 32KHz clock for counting, which can be selected using the "reg_timer_clk_sel" register.

3.7.5 Timer register overview

The timer register is accessed through the bus.
An overview of the timer register is shown in table 3.7.

Table 3- 7 Timer register overview (address 0x030A0000)

Name	Address Offset	Description
Timer1LoadCount	0x000	Value to be loaded into Timer1
Timer1CurrentValue	0x004	Current Value of Timer1
Timer1ControlReg	0x008	Control Register for Timer1
Timer1EOI	0x00c	Clears the interrupt from Timer1
Timer1IntStatus	0x010	Contains the interrupt status for Timer1
Timer2 Registers	0x014~ 0x024	共 5 个寄存器，内容与 Timer1 相同。
Timer3 Registers	0x028~ 0x038	共 5 个寄存器，内容与 Timer1 相同。
Timer4 Registers	0x03c~ 0x04c	共 5 个寄存器，内容与 Timer1 相同。

Name	Address Offset	Description
Timer5 Registers	0x050~ 0x060	共 5 个寄存器，内容与 Timer1 相同。
Timer6 Registers	0x064~ 0x074	共 5 个寄存器，内容与 Timer1 相同。
Timer7 Registers	0x078~ 0x088	共 5 个寄存器，内容与 Timer1 相同。
Timer8 Registers	0x08c~ 0x09c	共 5 个寄存器，内容与 Timer1 相同。
TimersIntStatus	0x0a0	Contains the interrupt status of all timers in the component.
TimersEOI	0x0a4	Returns all zeroes (0) and clears all active interrupts.
TimersRawIntStatus	0x0a8	Contains the unmasked interrupt status of all timers in the component.

3.7.6 Timer register description

Timer1LoadCount

Offset Address: 0x000

Bits	Name	Access	Description	Reset
31:0	Timer1LoadCount	R/W	Timer1 Load Count Register Value to be loaded into Timer1. This is the value from which counting commences. Any value written to this register is loaded into the associated timer.	0x0

Timer1CurrentValue

Offset Address: 0x004

Bits	Name	Access	Description	Reset
31:0	Timer1CurrentValue	RO	Timer1 Current Value Current Value of Timer1. This register is supported only when timer_1_clk is synchronous to pclk. Reading this register when using independent clocks results in an undefined value.	

Timer1ControlReg

Offset Address: 0x008

Bits	Name	Access	Description	Reset
2:0	Timer1ControlReg	R/W	[2] Timer interrupt mask for Timer1 0 – not masked 1 – masked [1] Timer mode for Timer1 0 – free-running mode 1 – user-defined count mode [0] Timer enable bit for Timer1	0x0

Bits	Name	Access	Description	Reset
			0 – disable 1 – enable	
31:3	Reserved			

Timer1EOI

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
0	Timer1EOI	RO	Reading from this register returns all zeroes (0) and clears the interrupt from Timer1.	
31:1	Reserved			

Timer1IntStatus

Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	Timer1IntStatus	RO	Contains the interrupt status for Timer1.	
31:1	Reserved			

TimersIntStatus

Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
7:0	TimersIntStatus	RO	Contains the interrupt status of all timers. Reading from this register does not clear any active interrupts: 0 – either timer_intr or timer_intr_n is not active after masking 1 – either timer_intr or timer_intr_n is active after masking	
31:8	Reserved			

TimersEOI

Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
7:0	TimersEOI	RO	Reading this register returns all zeroes (0) and clears all active interrupts.	
31:8	Reserved			

TimersRawIntStatus

Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
7:0	TimersRawIntStatus	RO	The register contains the unmasked interrupt status of all timers. 0 – either timer_intr or timer_intr_n is not active prior to masking 1 – either timer_intr or timer_intr_n is active prior to masking	
31:8	Reserved			

3.8 Watchdog

3.8.1 Overview

The system is equipped with a watchdog module. It is used to send interrupt or reset signal to interrupt or reset the whole system after a certain time when the system is abnormal.

3.8.2 Charateristics

WatchDog has the following characteristics:

- a 32bit configurable decrement counter.
- The initial counter value (i.e. timeout period) can be configured.
- Support watchdog restart protection to prevent watchdog from being restarted due to misoperation.
- Support reset signal generation.
- Support timeout interrupt generation.

3.8.3 Function description

The system configures watchdog registers through system bus. In order to monitor system operation, watchdog regularly sends WDT_INTR interrupt request to the system, and WDT_SYS_RST signal is sent out to reset the system when the system does not respond to the interrupt (e.g. system hang).

3.8.3.1 Application block diagram

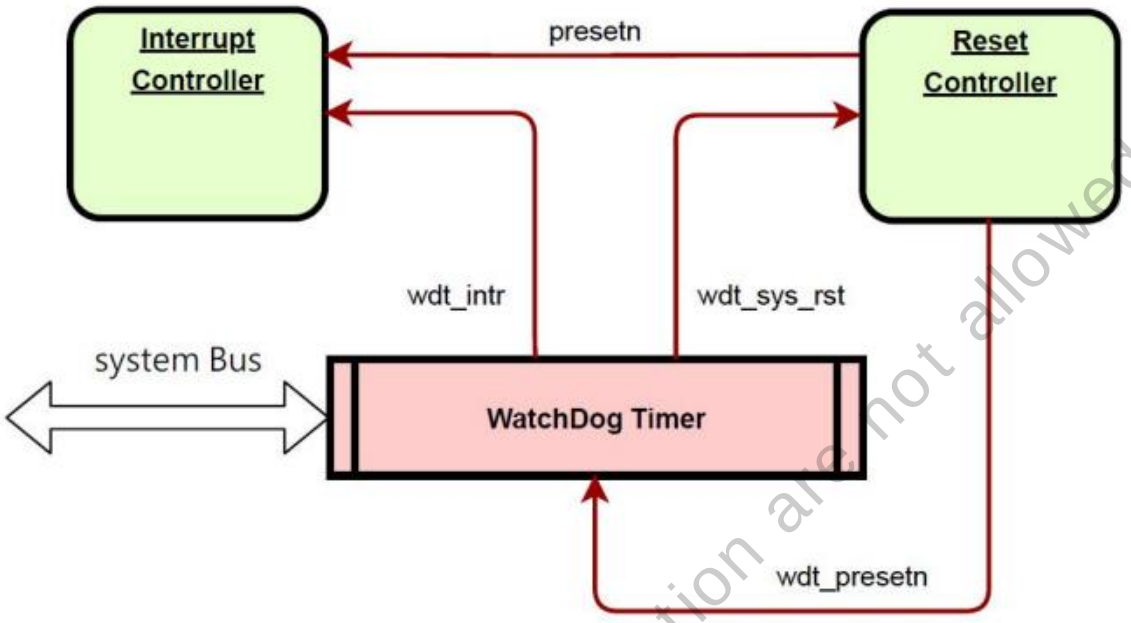


Figure 3- 8 WatchDog Application block diagram

3.8.3.2 Function description

The initial counter value of watchdog is loaded by register WDT_TORR and working based on a 32bit decrement counter. When the watchdog clock is enabled, the counter value is subtracted by 1 on each rising edge of the counting clock. When the counter is decremented to 0, watchdog will generate an interrupt. Then, at the next rising edge of the counting clock, the counter starts to reload the initial counter value from the register WDT_TORR and begins to decrement the counter.。

If the CPU has not cleared the watchdog interrupt when the counter value is decremented to 0 for the second time, the Watchdog will send a reset signal WDT_SYS_RST, the counter stops counting. The user can set the register WDT_CR[1] to decides whether to send the reset signal WDT_SYS_RST immediately when the counter value is decremented to 0 for the first time.

3.8.4 Working mode

3.8.4.1 Counting clock frequency configuration

The Watchdog counting clock can be either 25MHz or 32KHz. Use the "reg_wdt_clk_sel" to select.

3.8.4.2 System initialization configuration

After the system is powered on and reset, the watchdog counter is in the stop counting state, and it needs to be initialized and enabled during the system initialization. The initialization process of watchdog is as follows:
Step 1 write register WDT_TORR, set the initial value of watchdog counter.
Step 2 write register WDT_CR[1], set the watchdog counter timeout response mode.
Step 3 write register WDT_CR[0], start watchdog counting.。

3.8.4.3 Interrupt processing

After receiving the interrupt from watchdog, the system should clear its interrupt status in time.
The process of watchdog interrupt processing is as follows:
Step 1 read register WDT_EOI to clear the interrupt state of watchdog.
Step 2 write 0x76 to register WDT_CRR, restart watchdog.

3.8.4.4 Close WatchDog

Write register WDT_CR[0] to control the state of watchdog:
0 : WDT close。
1 : WDT startup. Only system reset can shut down WDT after startup.

3.8.5 WDT register overview

The WDT registers are accessed through the bus. The four base addresses for the WDT are:

WDT0	: 0x03010000
WDT1	: 0x03011000

WDT2 : 0x03012000
RTCSYS_WDT : 0x0502D000

An overview of the WDT register is shown in table 3-8.

Table 3- 8 WDT register overview (address 0x03010000)

Name	Address Offset	Description
WDT_CR	0x000	Control register
WDT_TORR	0x004	Timeout range register
WDT_CCVR	0x008	Current counter value register
WDT_CRR	0x00c	Counter restart register
WDT_STAT	0x010	Interrupt status register
WDT_EOI	0x014	Interrupt clear register
WDT_TOC	0x01C	Time Out Count

3.8.6 WDT register description

WDT_CR

Offset Address: 0x000

Bits	Name	Access	Description	Reset
4:0	WDT_CR	R/W	[4:2] Reset pulse length. This is used to select the number of pclk cycles for which the system reset stays asserted. The range of values available is 2 to 256 pclk cycles. 000 – 2 pclk cycles 001 – 4 pclk cycles 010 – 8 pclk cycles 011 – 16 pclk cycles 100 – 32 pclk cycles 101 – 64 pclk cycles 110 – 128 pclk cycles 111 – 256 pclk cycles [1] Response mode. Selects the output response generated to a timeout. 0 = Generate a system reset. 1 = First generate an interrupt and if it is not cleared by the time a second timeout occurs then generate a system reset.	0x0

Bits	Name	Access	Description	Reset
			[0] WDT enable. This bit is used to enable and disable the WatchDog. When disabled, the counter does not decrement. Thus, no interrupts or system resets are generated. Once this bit has been enabled, it can be cleared only by a system reset. 0 = WDT disabled. 1 = WDT enabled.	
6	TOR_MODE	R/W	The Mode of Timeout Period	0x0
7	ITOR_MODE	R/W	The Mode of Timeout Period for initialization	0x0
31:5	Reserved			

WDT_TORR

Offset Address: 0x004

Bits	Name	Access	Description	Reset
3:0	WDT_TORR	R/W	[3:0] TOP(TimeOut Period). This field is used to select the timeout period from which the watchdog counter restarts. A change of the timeout period takes effect only after the next counter restart (kick). The range of values is limited by 32-bit width. If TOP is programmed to select a range that is greater than the counter width, the timeout period is truncated to fit to the counter width. This affects only the non-user specified values as users are limited to these boundaries during configuration. The range of values available for a 32-bit watchdog counter are: TOR_MODE = 0 T = 2^(16 + WDT_TORR) TOR_MODE = 1 T = WDT_TOC <<(WDT_TORR +1)	0x0
7:4	WDT_ITORR		Initial TimeOut Period ITOR_MODE = 0 T = 2^(16 + WDT_ITORR) ITOR_MODE = 1 T = WDT_TOC <<(WDT_ITORR +1)	
31:4	Reserved			

WDT_CCVR

Offset Address: 0x008

Bits	Name	Access	Description	Reset
31:0	WDT_CCVR	RO	This register, when read, is the current value of the internal counter.	

WDT_CRR

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
7:0	WDT_CRR	R/W	[7:0] Counter Restart Register This register is used to restart the WDT counter. As a safety feature to prevent accidental restarts, the value 0x76 must be written. A restart also clears the WDT interrupt. Reading this register returns zero.	0x0
31:8	Reserved			

WDT_STAT

Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	WDT_STAT	RO	[0] Interrupt Status Register This register shows the interrupt status of the WDT. 1 = Interrupt is active regardless of polarity. 0 = Interrupt is inactive.	
31:1	Reserved			

WDT_EOI

Offset Address: 0x014

Bits	Name	Access	Description	Reset
0	WDT_EOI	RO	[0] Interrupt Clear Register Clears the watchdog interrupt. This can be used to clear the interrupt without restarting the watchdog counter.	
31:1	Reserved			

WDT_TOC

Offset Address: 0x01C

Bits	Name	Access	Description	Reset
15:0	WDT_TOC	R/W	Time out counter	0x0

3.9 Real time clock

3.9.1 Overview

Real time clock (RTC) is an independent power domain block in the chip. It contains a 32KHz oscillator and a Power-on-reset (POR) module and can be used for date clock display and alarm clock generation. In addition, the internal hardware finite state

machine provides the timing sequence control for triggering chip Power-on, Power-off and system reset.

3.9.2 Features

The features of RTC are listed as below:

Provide system reset source
Provide 32768 Hz clock source (mismatch < ± 1%)
Provide a 32-bit second counter and hardware calibration circuit, which can achieve a second accuracy level of 5ppm.
Supports alarm clock configuration and generates alarm interrupt.
Provides 2KB SRAM space for storing software code or temporary data
Supports battery low voltage detection and generates interrupt.
Supports waking up the chip from sleep by pressing a button.
Supports triggering chip sleep, reset, or overheating restart by software.
Supports Watchdog triggering system reset of the chip.
Supports waking up the chip from sleep by alarm triggering.
Power-up/down timing, reset time interval can be configured.
Provides 1 ultra-low-power analog clock counter (32 bit to count from 1970~2106 year).

3.9.3 Function description

RTC is an always-on power domain module. When RTC is powered-on at the first time, its POR circuit will generate a low-level pulse, and then the 32KHz oscillator starts to vibrate. After POR turns to high level, RTC enters the initial state and waits for event trigger.

When the state machine detects that the battery voltage is in a normal state, it starts to complete the chip power-on process according to the default timing and release the system reset signal. The software needs to initialize the RTC and configure the initial count value after the first boot. When the system needs to shut down or enter sleep mode, the RTC state machine can be triggered to complete the chip power-off process

according to the configured timing by configuring the system control register (RTC_CTRL).

When chip is at power-off or sleep-mode state, RTC will still keep operating and the necessary software code or user data can be stored in RTC SRAM and also information register (RTC_INFO0~3, RTC_NOPOR_INFO0~3); The second counter will keep counting, and at the same time the state machine keep detecting any key event trigger chip power-on or wake-up from sleep. After system resumes or reboots, software can judge the chip status by reading back RTC status registers or the contents written into information registers previously. Two status registers (RTC_ST_ON_REASON, RTC_ST_OFF_REASON) are provided to record the triggering condition of last time chip happens power-down, power-up, or reset respectively. It can provide more detail information, such as whether unexpected events have occurred like force reset, chip overheat (thermal shutdown), or battery low, power supply drop, etc. In addition, RTC will issue system reset while receives the watchdog event.

The RTC second counter is counting by 32KHz clock and is based on a 32-bit adder. The counter initial value can be loaded by register RTC_SET_SEC_CNTR_VALUE. The second value can be converted into specific year, month, day, hour and minutes.

The 32KHz clock and the second pulse period can be calibrated through a software process or by enabling a hardware module to perform automatic calibration periodically.

Software can specify the alarm time by configuring the 32-bit register RTC_ALARM_TIME and set register bit RTC_ALARM_ENABLE to 1 to enable Alarm function. When the second counter value RTC_SEC_CNTR_VALUE equals to RTC_ALARM_TIME, an alarm interrupt will be generated. The interrupt status will be kept until RTC_ ALARM_ENABLE is set to 0.

In addition, RTC provides battery low-voltage detection function. When the battery voltage is lower than a certain level, an interrupt will be generated. Software can immediately execute the shutdown procedure and trigger RTC to complete the power-down process after receiving the interrupt in order to prevent abnormal errors happen.

3.9.4 Operation

3.9.4.1 Counting clock

The maximum counting time of RTC second counter is:

2^32 = 49710 days = 136 years

3.9.4.2 Reset RTC

RTC serves as the power-on and power-off control unit of the chip, and it cannot be reset separately by software. Except for the POR at the first power-on, it supports forcing a full chip reset (including RTC) via the RSTN button in case of an exception. After the RSTN button is released, all RTC registers will return to their default values, and the state machine will return to the initial state. If the state machine detects that the battery voltage is in a normal state, it will begin to complete the chip power-on process and release the system reset signal according to the default timing.

3.9.4.3 RTC initialization

System needs to initialize RTC after chip is powered-on first time. The 32KHz clock and second time period need to be calibrated flirtly. The calibration circuit uses 25MHz crystal clock to sample 32KHz clock. In coarse tune mode, the 25MHz crystal clock samples one 32KHz clock cycle period and report the counting results. Then software adjusts the configuration register RTC_ANA_CALIB[8:0] to speed up or slow down the 32KHz clock rate depending on the counting results to improve the accuracy of 32KHz clock. The fine tune procedure can be further proceed after coarse tune complete. Uses 25MHz crystal clock to sample 256 32KHz clock cycles by default. Then software calculates the average value according to the counting result to obtain the number of pulses required for counting one second by 32KHz clock. The average value needs to be write to the register RTC_SEC_PULSE_GEN_INT and RTC_SEC_PULSE_GEN_FRAC to complete the second calibration process.

The coarse tune calibration process is summarized as follows:

1. Configure register RTC_ANA_SEL_FTUNE as 0, and set the initial value of RTC_ANA_CALIB as 0x100.
2. Implement calibration using binary search as follows:
    FTUNE = RTC_ANA_CALIB; offset = 0x80
3. Set RTC_FC_COARSE_EN to 1 to start coarse tuning. Poll the value of RTC_FC_COARSE_TIME until it is greater than the previous reading, then set RTC_FC_COARSE_EN to 0.
4. Read RTC_FC_COARSE_VALUE to obtain the count of one 32KHz clock cycle sampled by the 25MHz clock.
    if (RTC_FC_COARSE_VALUE > 770) FTUNE = FTUNE + offset;
    if (RTC_FC_COARSE_VALUE < 755) FTUNE = FTUNE - offset;
    Write the FTUNE value back to the RTC_ANA_CALIB register.
    offset = offset >> 1;
5. When the value of RTC_FC_COARSE_VALUE is between 755 and 770, the accuracy of the 32KHz clock has reached within ±1% of 32,768Hz, and coarse tuning is completed. Otherwise, wait for 0.5ms and repeat steps 3-5, up to a maximum of 8 times.

The fine tune calibration process is summarized as follows:

1. Configure the register RTC_SEL_SEC_PULSE to 0. Configure RTC_FC_FINE_EN to 1 to start fine-tuning.
2. Poll the value of RTC_FC_FINE_TIME until it is greater than the previous read value.
3. Read RTC_FC_FINE_VALUE to obtain the count of 256 32KHz clock cycles sampled by the 25MHz clock.
4. The frequency of the 32KHz clock can be obtained from the following equation:
    Frequency = 256 / (RTC_FC_FINE_VALUE x 40ns)
    For example: 256 / (195310 x 40) = 32768.4194357
5. Take the integer part, 32768, and write it to the register RTC_SEC_PULSE_GEN_INT. Take the fractional part, 8-bit = 0.4194357 x 256 = 107, and write it to the register RTC_SEC_PULSE_GEN_FRAC.
6. Configure RTC_FC_FINE_EN to 0 to end fine-tuning.

The clock calibration process can be performed by software as a one-time or periodic execution, depending on the system's needs. In addition to the software calibration process, RTC also supports automatic calibration performed periodically by hardware.

RTC needs to be further initialized after clock calibration. Only the necessary initialization are listed below. Most of the other parameter registers need to be configured only when the timing intervals of power sequence needs to be optimized. Otherwise, they are generally recommended to use the default value.

1. Configure the RTC_POR_DB_MAGIC_KEY register with the value 0x5AF0 to enable power-on reset (POR) debounce and prevent false triggering of POR caused by brief voltage drops in the RTC module power supply. The debounce time is about 1ms.

2. Set the RTC_SET_SEC_CNTR_VALUE register to initialize the RTC time counter.

3. Write 1 to RTC_SET_SEC_CNTR_TRIG to load the initial counter value into the RTC second counter.

4. Poll the RTC_SEC_CNTR_VALUE register until the read value equals the value in RTC_SET_SEC_CNTR_VALUE.

5. Set RTC_PWR_DET_COMP[0] to 1 to enable battery low voltage detection, and configure RTC_PWR_DET_SEL[0] to 1 to generate a low voltage detection interrupt when the battery voltage drops below the threshold value. The threshold value can be adjusted by configuring the RTC_PWR_DET_COMP[12:8] register.

6. After the first power-on, the RTC subsystem must be configured by setting the reg_rtcsys_rstn_src_sel register from the default value of 0 to 1 to maintain the working state of the RTC subsystem after the chip is powered down (suspend or powerdown). If this register is set to 0, the RTC subsystem will be reset when the chip is powered down.

7. After the first power-on, the RTC_EN_AUTO_POWER_UP register must be configured from the default value of 1 to 0. If the default value is maintained, the RTC will

automatically power up when the chip enters the power-down state (powerdown) and PWR_VBAT_DET is detected as high.

3.9.4.4 Analog Second Clock Initialization

1. Configure RTC_MACRO_RG_DEFD to 16'hC80 (32000 32KHz clock cycles)
2. Configure RTC_MACRO_DA_SOC_READY to 1
3. Configure RTC_MACRO_DA_CLEAR_ALL to 1
4. Configure RTC_MACRO_DA_CLEAR_ALL to 0
5. Configure RTC_MACRO_RG_SET_T to the desired Counter value
6. Configure RTC_MACRO_DA_LATCH_PASS to 1
7. Configure RTC_MACRO_DA_LATCH_PASS to 0
8. Configure RTC_MACRO_DA_SOC_READY to 0
9. Read RTC_MACRO_RO_T to obtain the counter value.

3.9.4.5 Interrupt handling

RTC can issue alarm interrupt and low voltage interrupt. While receives the alarm interrupt, set register bit RTC_ALARM_ENABLE to 0 to disable alarm and clear interrupt state. Specify the new value to register RTC_ALARM_TIME and set RTC_ALARM_ENABLE to 1 again if a new alarm time is required.

3.9.4.6 Suspend and wakeup

Set register bit req_suspend to 1 can trigger chip enter suspend/sleep mode. Specify register RTC_EN_PWR_WAKEUP can select the source that can trigger chip wake-up. Note that the register RTC_PG_REG must be written to 0 before setting req_suspend to retent DDR IO state in order to prevent the mis-operation of DDR interface and protect DDR contents during chip power-down or power-up period. After chip resumes, write the register RTC_PG_REG to 1 to release DDR IO retention before accessing DDR.

3.9.4.7 Power off and power on

By configuring the req_shdn register to 1, the system software can put the chip into a power-off state (poweroff) including DDR. The RTC_EN_PWR_UP register can be configured to select the source that triggers the chip to power up and boot (powerup).

3.9.5 RTC register overview

The RTC registers consist of multiple parts: RTC_CORE_REG, RTC_MACRO_REG, and RTC_CTRL_REG, with different base addresses, and are all accessed through the bus.

An overview of RTC_CORE_REG registers is listed in table 3-9.

Table 3- 9 RTC_REG register overview (base address: 0x05026000)

Name	Address Offset	Description
RTC_ANA_CALIB	0x000	32K 振荡器控制
RTC_SEC_PULSE_GEN	0x004	秒脉冲产生器整数位与小数位
RTC_ALARM_TIME	0x008	设定定时报警时间
RTC_ALARM_ENABLE	0x00c	使能报警
RTC_SET_SEC_CNTR_VALUE	0x010	设定秒计数器值
RTC_SET_SEC_CNTR_TRIG	0x014	加载秒计数器值
RTC_SEC_CNTR_VALUE	0x018	读取目前秒计数器值
RTC_INFO0	0x01c	信息寄存器 0
RTC_INFO1	0x020	信息寄存器 1
RTC_INFO2	0x024	信息寄存器 2
RTC_INFO3	0x028	信息寄存器 3
RTC_NOPOR_INFO0	0x02c	无复位信息寄存器 0
RTC_NOPOR_INFO1	0x030	无复位信息寄存器 1
RTC_NOPOR_INFO2	0x034	无复位信息寄存器 2
RTC_NOPOR_INFO3	0x038	无复位信息寄存器 3
RTC_DB_PWR_VBAT_DET	0x040	PWR_VBAT_DET 去抖动时间
RTC_DB_BUTTON1	0x048	PWR_BUTTON1 去抖动时间
RTC_DB_PWR_ON	0x04c	PWR_ON 去抖动时间
RTC_7SEC_RESET	0x050	设定长按 PWR_BUTTON 秒数强制 reset
RTC_THM_SHDN_AUTO_REBOOT	0x064	选择 REQ_THM_SHDN 动作
RTC_POR_DB_MAGIC_KEY	0x068	使能 POR 长时去抖动
RTC_DB_SEL_PWR	0x06c	选择 PWR_BUTTON 去抖动模式
RTC_UP_SEQ0	0x070	上电 PWR_SEQ0 输出时序
RTC_UP_SEQ1	0x074	上电 PWR_SEQ1 输出时序
RTC_UP_SEQ2	0x078	上电 PWR_SEQ2 输出时序
RTC_UP_SEQ3	0x07c	上电 PWR_SEQ3 输出时序
RTC_UP_IF_EN	0x080	上电 ISO 解除时序
RTC_UP_RSTN	0x084	上电系统复位解除时序

Name	Address Offset	Description
RTC_UP_MAX	0x088	上电流程完成时序
RTC_DN_SEQ0	0x090	下电 PWR_SEQ0 输出时序
RTC_DN_SEQ1	0x094	下电 PWR_SEQ1 输出时序
RTC_DN_SEQ2	0x098	下电 PWR_SEQ2 输出时序
RTC_DN_SEQ3	0x09c	下电 PWR_SEQ3 输出时序
RTC_DN_IF_EN	0x0a0	下电 ISO 打开时序
RTC_DN_RSTN	0x0a4	下电系统复位发出时序
RTC_DN_MAX	0x0a8	下电流程完成时序
RTC_PWR_CYC_MAX	0x0b0	Power-cycle 完成时序
RTC_WARM_RST_MAX	0x0b4	Warm-reset 完成时序
RTC_EN_7SEC_RST	0x0b8	设定 PWR_BUTTON1 7SEC reset 模式
RTC_EN_PWR_WAKEUP	0x0bc	设定休眠唤醒来源
RTC_EN_SHDN_REQ	0x0c0	使能 REQ_SHDN
RTC_EN_THM_SHDN	0x0c4	使能 REQ_THM_SHDN
RTC_EN_PWR_CYC_REQ	0x0c8	使能 REQ_PWR_CYC
RTC_EN_WARM_RST_REQ	0x0cc	使能 REQ_WARM_RST
RTC_EN_PWR_VBAT_DET	0x0d0	使能状态机参考 PWR_VBAT_DET
FSM_STATE	0x0d4	RTC 状态机值
RTC_EN_WDG_RST_REQ	0x0e0	使能 REQ_WDG_RST
RTC_EN_SUSPEND_REQ	0x0e4	使能 REQ_SUSPEND
RTC_DB_REQ_WDG_RST	0x0e8	REQ_WDG_RST 去抖动时间
RTC_DB_REQ_SUSPEND	0x0ec	REQ_SUSPEND 去抖动时间
RTC_PG_REG	0x0f0	Power Good 寄存器
RTC_ST_ON_REASON	0x0f8	上电状态寄存器
RTC_ST_OFF_REASON	0x0fc	下电状态寄存器
RTC_EN_WAKEUP_REQ	0x120	使能 REQ_WAKEUP
RTC_PWR_WAKEUP_POLARITY	0x128	选择 PWR_WAKEUP 低电平
RTC_DB_SEL_REQ	0x130	选择去抖动模式
RTC_PWR_DET_SEL	0x140	选择低电压检测信号来源

An overview of RTC_MACRO_REG registers is listed in table 3-10.

Table 3-10 RTC_MACRO_REG register overview(base address 0x05026400)

Name	Address Offset	Description
RTC_PWR_DET_COMP	0x044	低电压检测控制
RTC_MACRO_DA_CLEAR_ALL	0x080	DA_CLEAR_ALL
RTC_MACRO_DA_SET_ALL	0x084	DA_SEL_ALL
RTC_MACRO_DA_LATCH_PASS	0x088	DA_LATCH_PASS
RTC_MACRO_DA_SOC_READY	0x08c	DA_SOC_READY
RTC_MACRO_PD_SLDO	0x090	PD_SLDO
RTC_MACRO_RG_DEFD	0x094	RG_DEFD
RTC_MACRO_RG_SET_T	0x098	RG_SET_T
RTC_MACRO_RO_CLK_STOP	0x0a0	RO_CLK_STOP
RTC_MACRO_RO_DEFQ	0x0a4	RO_DEFQ
RTC_MACRO_RO_T	0x0a8	RO_T

An overview of RTC_CTRL registers is listed in table 3-11.

Table 3- 11 RTC_CTRL register overview (base address: 0x03004000)

Name	Address Offset	Description
rtc_ctrl_unlockkey	0x004	rtc_ctrl_unlockkey
rtc_ctrl0	0x008	rtc_ctrl0
rtc_ctrl_status0	0x00c	rtc_ctrl_status0
rtc_ctrl_status1	0x010	rtc_ctrl_status1
rtc_ctrl_status2gpio	0x014	rtc_ctrl_status2gpio
rtcsys_rst_ctrl	0x018	rtcsys_rst_ctrl
rtcsys_clkmux	0x01c	rtcsys_clkmux
rtcsys_mcu51_ctrl0	0x020	rtcsys_mcu51_ctrl0
rtcsys_mcu51_ctrl1	0x024	rtcsys_mcu51_ctrl1
rtcsys_pmu	0x028	rtcsys_pmu
rtcsys_status	0x02c	rtcsys_status
rtcsys_clkbyp	0x030	rtcsys_clkbyp
rtcsys_clk_en	0x034	rtcsys_clk_en
rtcsys_wkup_ctrl	0x038	rtcsys_wkup_ctrl
rtcsys_clkdiv	0x03c	rtcsys_clkdiv
fc_coarse_en	0x040	fc_coarse_en
fc_coarse_cal	0x044	fc_coarse_cal
fc_fine_en	0x048	fc_fine_en
fc_fine_period	0x04c	fc_fine_period
fc_fine_cal	0x050	fc_fine_cal
rtcsys_pmu2	0x054	rtcsys_pmu2
rtcsys_clkdiv1	0x058	rtcsys_clkdiv1
rtcsys_mcu51_dbg	0x05c	rtcsys_mcu51_dbg
sw_reg0	0x060	sw_reg0
sw_reg1_por	0x064	sw_reg1_por
fab_lp_ctrl	0x068	fab_lp_ctrl
fab_option	0x06c	fab_option
rtcsys_mcu51_ictrl1	0x07c	rtcsys_mcu51_ictrl1
rtc_ip_pwr_req	0x080	rtc_ip_pwr_req
rtc_ip_iso_ctrl	0x084	rtc_ip_iso_ctrl
rtcsys_spare_reg0	0x088	rtcsys_spare_reg0
rtcsys_spare_reg1	0x08c	rtcsys_spare_reg1
rtcsys_spare_ro	0x090	rtcsys_spare_ro
rtcsys_wkup_ctrl1	0x094	rtcsys_wkup_ctrl1
rtcsys_sram_ctrl	0x098	rtcsys_sram_ctrl
rtcsys_io_ctrl	0x09c	rtcsys_io_ctrl
rtcsys_wdt_ctrl	0x0a0	rtcsys_wdt_ctrl
rtcsys_irrx_clk_ctrl	0x0a4	rtcsys_irrx_clk_ctrl
rtcsys_rtc_wkup_ctrl	0x0a8	rtcsys_rtc_wkup_ctrl
rtcsys_por_rst_ctrl	0x0ac	rtcsys_por_rst_ctrl

3.9.6 RTC register description

3.9.6.1 RTC_CORE_REG

RTC_ANA_CALIB

Offset Address: 0x000

Bits	Name	Access	Description	Reset
15:0	RTC_ANA_CALIB	R/W	Adjusting the frequency of the analog module's 32K oscillator.	0x100
17:16	RTC_ANA_ISEL	R/W	Adjusting the current of the analog module's 32K XTAL oscillator. 00 = 2uA, 01 = 1.5uA, 11 = 0.5uA	0x3
30:18	Reserved			
31	RTC_ANA_SEL_FTUNE	R/W	Select 32K OSC calibration value source: 0 = controlled by RTC_ANA_CALIB register 1 = controlled by hardware circuitry	0x1

RTC_SEC_PULSE_GEN

Offset Address: 0x004

Bits	Name	Access	Description	Reset
7:0	RTC_SEC_PULSE_GEN_FRAC	R/W	the fractional part of the second pulse generator	0x0
23:8	RTC_SEC_PULSE_GEN_INT	R/W	the integer part of the second pulse generator When the counter increment value is greater than the integer part value, a second pulse signal is generated to increment the second counter.	0x8000
30:24	Reserved			
31	RTC_SEL_SEC_PULSE	R/W	Select the source of second pulse signal: 0 = Second pulse signal is generated internally. 1 = Second pulse signal is generated by external hardware circuit. When set to 1, the registers RTC_SEL_PULSE_GEN_FRAC and RTC_SEL_PULSE_GEN_INT have no effect.	0x1

RTC_ALARM_TIME

Offset Address: 0x008

Bits	Name	Access	Description	Reset
31:0	RTC_ALARM_TIME	R/W	Set the time for timed alarm.	0xffffffff

RTC_ALARM_ENABLE

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
0	RTC_ALARM_ENABLE	R/W	Alarm Enable Set to 1 to enable the alarm, set to 0 to	0x0

Bits	Name	Access	Description	Reset
			disable or clear the alarm interrupt status.	
31:1	Reserved			

RTC_SET_SEC_CNTR_VALUE

Offset Address: 0x010

Bits	Name	Access	Description	Reset
31:0	RTC_SET_SEC_CNTR_VALUE	R/W	Set the value of the second counter.	0x0

RTC_SET_SEC_CNTR_TRIG

Offset Address: 0x014

Bits	Name	Access	Description	Reset
0	RTC_SET_SEC_CNTR_TRIG	W1C	Load seconds counter value. Set to 1 to enable RTC_SET_SEC_CNTR_VALUE to take effect. The register will automatically clear to 0 after being written to 1.	
31:1	Reserved			

RTC_SEC_CNTR_VALUE

Offset Address: 0x018

Bits	Name	Access	Description	Reset
31:0	RTC_SEC_CNTR_VALUE	RO	Read current value of the second counter.	

RTC_INFO0

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
31:0	RTC_INFO0	R/W	Information register 0	0xABCD 1234

RTC_INFO1

Offset Address: 0x020

Bits	Name	Access	Description	Reset
31:0	RTC_INFO1	R/W	Information register 1	0xDEAD BEEF

RTC_INFO2

Offset Address: 0x024

Bits	Name	Access	Description	Reset
31:0	RTC_INFO2	R/W	Information register 2	0xABCD 1234

RTC_INFO3

Offset Address: 0x028

Bits	Name	Access	Description	Reset
31:0	RTC_INFO3	R/W	Information register 3	0xDEAD BEEF

RTC_NOPOR_INFO0

Offset Address: 0x02c

Bits	Name	Access	Description	Reset

Bits	Name	Access	Description	Reset
31:0	RTC_NOPOR_INFO0	R/W	No reset information register 0	Random

RTC_NOPOR_INFO1

Offset Address: 0x030

Bits	Name	Access	Description	Reset
31:0	RTC_NOPOR_INFO1	R/W	No reset information register 1	Random

RTC_NOPOR_INFO2

Offset Address: 0x034

Bits	Name	Access	Description	Reset
31:0	RTC_NOPOR_INFO2	R/W	No reset information register 2	Random

RTC_NOPOR_INFO3

Offset Address: 0x038

Bits	Name	Access	Description	Reset
31:0	RTC_NOPOR_INFO3	R/W	No reset information register 3	Random

RTC_APB_BUSY_SEL

Offset Address: 0x03c

Bits	Name	Access	Description	Reset
3:0	Reserved	R/W		
4	rtc_apb_32k_busy_sel	R/W	Select the source of the RTC PCLK busy signal (keep PCLK at full speed when busy): 0 = The PCLK busy signal is generated by the hardware circuit. 1 = The PCLK busy signal is controlled by the register rtc_apb_32k_force_busy.	0x0
7:5	Reserved			
8	rtc_apb_32k_force_busy	R/W	1 = PCLK runs at full speed always. 0 = PCLK only returns to full speed when PSel is asserted.	0x0
31:9	Reserved			

RTC_DB_PWR_VBAT_DET

Offset Address: 0x040

Bits	Name	Access	Description	Reset
15:0	RTC_DB_PWR_VBAT_DET	R/W	PWR_VBAT_DET debounce time (unit: 32K clocks)	0x2
31:16	Reserved			

RTC_DB_BUTTON1

Offset Address: 0x048

Bits	Name	Access	Description	Reset
15:0	RTC_DB_BUTTON1	R/W	PWR_BUTTON1 debounce time (unit: 32K clocks) The default value 0x100 is about 8ms.	0x100
31:16	Reserved			

RTC_DB_PWR_ON

Offset Address: 0x04c

Bits	Name	Access	Description	Reset

Bits	Name	Access	Description	Reset
15:0	RTC_DB_PWR_ON	R/W	PWR_ON debounce time (unit: 32K clocks)	0x100
31:16	Reserved			

RTC_7SEC_RESET

Offset Address: 0x050

Bits	Name	Access	Description	Reset
7:0	RTC_7SEC_RESET	R/W	Long press PWR_BUTTON1 reset debounce time (unit: seconds). This register will only be cleared by POR.	0x7
15:8	Reserved			
31:16	RTC_7SEC_UNLOCK_KEY	WO	Writing 0xDC78 at the same time can remove the write protection of RTC_7SEC_RESET.	0x0

RTC_THM_SHDN_AUTO_REBOOT

Offset Address: 0x064

Bits	Name	Access	Description	Reset
0	RTC_THM_SHDN_AUTO_REBOOT	R/W	Select the behavior when receiving REQ_THM_SHDN: 0 = start power-off process 1 = start power-cycle process (power off and then power on again)	0x0
31:1	Reserved			

RTC_POR_DB_MAGIC_KEY

Offset Address: 0x068

Bits	Name	Access	Description	Reset
15:0	RTC_POR_DB_MAGIC_KEY	R/W	Writing 0x5AF0 will cause a POR debounce (approximately 1ms).	Random
31:16	Reserved			

RTC_DB_SEL_PWR

Offset Address: 0x06c

Bits	Name	Access	Description	Reset
0	Reserved			
1	DB_SEL_PWR_BUTTON1	R/W	Select the debounce mode for PWR_BUTTON1: 0 = The state machine uses the falling edge of the PWR_BUTTON1 debounce signal as trigger. 1 = The state machine uses the low level of the PWR_BUTTON1 debounce signal as trigger.	0x1
2	DB_SEL_PWR_ON	R/W	Select PWR_ON debounce mode: 0 = The state machine is triggered by the rising edge of the PWR_ON debounce signal. 1 = The state machine is triggered by the high level of the PWR_ON debounce signal.	0x1
3	DB_SEL_PWR_WAKEUP0	R/W	Select PWR_WAKEUP0 debounce mode 0 = The state machine triggers on the	0x1

Bits	Name	Access	Description	Reset
			rising edge of PWR_WAKEUP0 debounce signal 1 = The state machine triggers on the high level of PWR_WAKEUP0 debounce signal.	
4	DB_SEL_PWR_WAKEUP1	R/W	Select PWR_WAKEUP1 debounce mode: 0 = State machine triggers on rising edge of PWR_WAKEUP1 debounce signal.	0x1
31:5	Reserved			

RTC_UP_SEQ0

Offset Address: 0x070

Bits	Name	Access	Description	Reset
15:0	RTC_UP_SEQ0	R/W	Power-on sequence PWR_SEQ0 rising time from 0 to 1 (unit: 32K clocks)	0x0
31:16	Reserved			

RTC_UP_SEQ1

Offset Address: 0x074

Bits	Name	Access	Description	Reset
15:0	RTC_UP_SEQ1	R/W	Power-on sequence PWR_SEQ1 rising time from 0 to 1 (unit: 32K clocks)	0x40
31:16	Reserved			

RTC_UP_SEQ2

Offset Address: 0x078

Bits	Name	Access	Description	Reset
15:0	RTC_UP_SEQ2	R/W	Power-on sequence PWR_SEQ2 rising time from 0 to 1 (unit: 32K clocks)	0x80
31:16	Reserved			

RTC_UP_SEQ3

Offset Address: 0x07c

Bits	Name	Access	Description	Reset
15:0	RTC_UP_SEQ3	R/W	Power-on sequence PWR_SEQ3 rising time from 0 to 1 (unit: 32K clocks)	0xc0
31:16	Reserved			

RTC_UP_IF_EN

Offset Address: 0x080

Bits	Name	Access	Description	Reset
15:0	RTC_UP_IF_EN	R/W	Time for releasing the isolation of the power-off area during the power-on sequence (unit: 32K clock).	0x100
31:16	Reserved			

RTC_UP_RSTN

Offset Address: 0x084

Bits	Name	Access	Description	Reset
15:0	RTC_UP_RSTN	R/W	Power-up process system reset release time (unit: 32K clock)	0x140

Bits	Name	Access	Description	Reset
31:16	Reserved			

RTC_UP_MAX

Offset Address: 0x088

Bits	Name	Access	Description	Reset
15:0	RTC_UP_MAX	R/W	Complete power-up sequence completion time (unit: 32K clocks). RTC_UP_SEQ0~RTC_UP_MAX are the absolute timing of each stage of the power-up sequence. It is recommended to use the default values.	0x180
31:16	Reserved			

RTC_DN_SEQ0

Offset Address: 0x090

Bits	Name	Access	Description	Reset
15:0	RTC_DN_SEQ0	R/W	The time for PWR_SEQ0 output to transition from 1 to 0 during the power-down process (in units of 32K clock).	0x140
31:16	Reserved			

RTC_DN_SEQ1

Offset Address: 0x094

Bits	Name	Access	Description	Reset
15:0	RTC_DN_SEQ1	R/W	The time for PWR_SEQ1 output to transition from 1 to 0 during the power-down process (in units of 32K clock).	0x100
31:16	Reserved			

RTC_DN_SEQ2

Offset Address: 0x098

Bits	Name	Access	Description	Reset
15:0	RTC_DN_SEQ2	R/W	The time for PWR_SEQ2 output to transition from 1 to 0 during the power-down process (in units of 32K clock).	0xc0
31:16	Reserved			

RTC_DN_SEQ3

Offset Address: 0x09c

Bits	Name	Access	Description	Reset
15:0	RTC_DN_SEQ3	R/W	The time for PWR_SEQ3 output to transition from 1 to 0 during the power-down process (in units of 32K clock).	0x80
31:16	Reserved			

RTC_DN_IF_EN

Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
15:0	RTC_DN_IF_EN	R/W	The duration for opening the isolation signal of the power-down area during the power-down process (unit: 32K clock).	0x40

Bits	Name	Access	Description	Reset
31:16	Reserved			

RTC_DN_RSTN

Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
15:0	RTC_DN_RSTN	R/W	Time for the power-down process to issue a system reset (unit: 32K clock).	0x0
31:16	Reserved			

RTC_DN_MAX

Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
15:0	RTC_DN_MAX	R/W	Complete shutdown or sleep process completion time (unit: 32K clock cycles). RTC_DN_SEQ0~RTC_DN_MAX are the absolute timing of each stage of the shutdown process, and it is recommended to use the default values.	0x180
31:16	Reserved			

RTC_PWR_CYC_MAX

Offset Address: 0x0b0

Bits	Name	Access	Description	Reset
15:0	RTC_PWR_CYC_MAX	R/W	Complete Power-cycle process completion time (Unit: 32K clock) The Power-cycle time includes the complete down sequence process and up sequence process.	0x4000
31:16	Reserved			

RTC_WARM_RST_MAX

Offset Address: 0x0b4

Bits	Name	Access	Description	Reset
15:0	RTC_WARM_RST_MAX	R/W	Complete WARM_RESET process completion time (unit: 32K clocks) Equivalent to the low-level time of system reset.	0x40
31:16	Reserved			

RTC_EN_7SEC_RST

Offset Address: 0x0b8

Bits	Name	Access	Description	Reset
0	RTC_EN_7SEC_RST	R/W	Enable long-pressing PWR_BUTTON1 for 7 seconds to trigger RTC forced reset.	0x0
1	RTC_7SEC_RST_MODE	R/W	7-second forced reset mode 0 = Low-level mode, 1 = Short-pulse mode When a 7-second forced reset occurs, select whether to generate a short pulse reset signal or keep the reset until the PWR_BUTTON1 button is released.	0x0
2	DB_SEL_PWR_BUTTON1_7SEC	R/W	0 = Reset 7-second reset counter if	0x0

Bits	Name	Access	Description	Reset
			PWR_BUTTON1 is pressed after power-up sequence is triggered. 1 = NOP.	
3	SEL_7SEC_RST_RTCSYS	R/W	0 = The 7-second forced reset signal will reset the RTC subsystem (sec reset will reset rtcsys). 1 = The RTC subsystem will not be reset.	0x1
15:4	Reserved			
31:16	RTC_EN_7SEC_UNLOCK_KEY	WO	Simultaneously writing 0xDC78 can remove the write protection for [3:0].	0x0

RTC_EN_PWR_WAKEUP

Offset Address: 0x0bc

Bits	Name	Access	Description	Reset
6:0	RTC_EN_PWR_WAKEUP	R/W	Set the source that can trigger wake-up from self-sleep mode: 0 = Cannot trigger wake-up 1 = Can trigger wake-up. [0] = PWR_WAKEUP0 [1] = PWR_WAKEUP1 [2] = PWR_ON [3] = REQ_POWERUP [4] = PWR_BUTTON1 [5] = Alarm [6] = REQ_WAKEUP	0x0
7	Reserved			
14:8	RTC_EN_PWR_UP	R/W	Set the sources that can trigger power-on. 0 = Cannot trigger power-on. 1 = Can trigger power-on. [8] = PWR_WAKEUP0 [9] = PWR_WAKEUP1 [10] = PWR_ON [11] = REQ_POWERUP [12] = PWR_BUTTON1 [13] = Alarm [14] = REQ_WAKEUP	0x14
31:15	Reserved			

RTC_EN_SHDN_REQ

Offset Address: 0x0c0

Bits	Name	Access	Description	Reset
0	RTC_EN_SHDN_REQ	R/W	Enable software to request power down. (REQ_SHDN) 0 = disable, 1 = enable	0x0
31:1	Reserved			

RTC_EN_THM_SHDN

Offset Address: 0x0c4

Bits	Name	Access	Description	Reset
0	RTC_EN_THM_SHDN	R/W	Enable request for thermal shutdown or reboot.(REQ_THM_SHDN) 0 = disable, 1 = enable	0x0
31:1	Reserved			

RTC_EN_PWR_CYC_REQ

Offset Address: 0x0c8

Bits	Name	Access	Description	Reset
0	RTC_EN_PWR_CYC_REQ	R/W	Enable request for Power-cycle (REQ_PWR_CYC) 0 = disable, 1 = enable	0x0
31:1	Reserved			

RTC_EN_WARM_RST_REQ

Offset Address: 0x0cc

Bits	Name	Access	Description	Reset
0	RTC_EN_WARM_RST_REQ	R/W	Enable request for a system soft restart. (REQ_WARM_RST) 0 = disable, 1 = enable	0x0
31:1	Reserved			

RTC_EN_PWR_VBAT_DET

Offset Address: 0x0d0

Bits	Name	Access	Description	Reset
0	RTC_EN_PWR_VBAT_DET_UP	R/W	Enable State Machine to Reference Battery Low Voltage Detection State (PWR_VBAT_DET) 0 = disable, 1 = enable If this value is set to 1, when any button attempts to trigger power-on or wake-up, the state machine will check the low voltage detection output value. If the low voltage detection output is low (indicating low battery voltage or no power supply), the RTC state machine will maintain the current state without changing.	0x1
1	RTC_EN_PWR_VBAT_DET_DN	R/W	Enable power off on battery low voltage status: 0 = disable, 1 = enable If this value is set to 1, when the chip is powered on or in sleep mode, the RTC state machine will check the low voltage detection output. If the low voltage detection output transitions from high to low (indicating low battery voltage or power loss), the state machine will trigger the power-off process.	0x1
2	RTC_EN_AUTO_POWER_UP	R/W	Enable RTC state machine to automatically enter power-up state: 1 = Automatically power-up when entering Power-down and PWR_VBAT_DET is at high level. 0 = Stay in this state when entering Power-down, until any power-up source is triggered.	0x1
31:3	Reserved			

FSM_STATE

Offset Address: 0x0d4

Bits	Name	Access	Description	Reset
3:0	FSM_STATE	RO	RTC state machine values: 4'h0 = ST_OFF (Power-off completed) 4'h1 = ST_UP (Power-up process in progress) 4'h2 = ST_DN (Power-down process in progress) 4'h3 = ST_ON (Power-on completed) 4'h4 = ST_PWR_CYC2 (Power-cycle power-down completed) 4'h6 = ST_PWR_CYC (Power-cycle power-down in progress) 4'h7 = ST_WARM_RESET (System reset in progress) 4'h9 = ST_SUSP (System suspended) 4'hB = ST_PRE_SUSP (Suspend power-down process in progress)	
31:4	Reserved			

RTC_EN_WDG_RST_REQ

Offset Address: 0x0e0

Bits	Name	Access	Description	Reset
0	RTC_EN_WDG_RST_REQ	R/W	Enable Watchdog to request reset of system(REQ_WDG_RST) 0 = disable, 1 = enable	0x0
1	RTC_EN_SUS_WDG_RST_REQ	R/W	Enable Watchdog to request system reset when in Sleep Mode. 0 = disable, 1 = enable	0x1
31:2	Reserved			

RTC_EN_SUSPEND_REQ

Offset Address: 0x0e4

Bits	Name	Access	Description	Reset
0	RTC_EN_SUSPEND_REQ	R/W	Enable request for sleep(REQ_SUSPEND) 0 = disable, 1 = enable	0x0
31:1	Reserved			

RTC_PG_REG

Offset Address: 0x0f0

Bits	Name	Access	Description	Reset
3:0	RTC_PG_REG	R/W	Chip Power Good Status 1 = Chip powered on (Power Good), IO signals can pass through 0 = Chip powered off, IO signals are in retentive state [0] = Controls DDR IO [3:1] = Reserved This register signal is used to control whether the chip and DDR IO interface are normally passed through or in retentive state. Before the system enters sleep, the software must first set the value of this register to 0 to keep the DDR IO in a fixed state. After the system is awakened from sleep, the	0xF

Bits	Name	Access	Description	Reset
			value of this register must be set to 1 to restore normal operation of the DDR. When entering power-down state, the value of this register will be automatically cleared to all 1.	
31:4	Reserved			

RTC_ST_ON_REASON

Offset Address: 0x0f8

Bits	Name	Access	Description	Reset
3:0	ST_ON_REASON_LAST_STATE	RO	RTC state machine returns to power-on completion (ST_ON) state from the following states: 4'h0 = Return from power-off (ST_OFF) to power-on state 4'h3 = Return from Power-cycle or Warm-reset to power-on state 4'h9 = Return from sleep to power-on state After system reboot, software can read this register to determine the cause of the chip's power-on.	
15:4	Reserved			
31:16	ST_ON_REASON_LAST_INPUT	RO	Trigger reasons for the state machine to return to the power-on state (record the status value of each signal): [0] = PWR_VBAT_DET (0: power-off triggered) [1] = PWR_ON (1: power-on triggered by button) [2] = RTC_EN_AUTO_POWER_UP [3] = PWR_BUTTON1 (0: power-on triggered by button) [4] = PWR_BUTTON1_7SEC [5] = PWR_WAKEUP0 (1: wakeup triggered by button) [6] = PWR_WAKEUP1 (1: wakeup triggered by button) [7] = Alarm (1: timed alarm occurred) [8] = REQ_PWR_CYC (1: software-triggered Power-cycle) [9] = REQ_THM_SHDN (1: software-triggered power-off/power-cycle) [10] = REQ_WARM_RST (1: software-triggered reset) [11] = REQ_WDG_RST (1: Watchdog-triggerd reset) [12] = REQ_SHDN (1: software-triggered power-off) [13] = REQ_SUSPEND (1: software-triggered suspend) [14] = REQ_WAKEUP (1: event-triggered wakeup) [15] = REQ_POWERUP	

RTC_ST_OFF_REASON

Offset Address: 0x0fc

Bits	Name	Access	Description	Reset
3:0	ST_OFF_REASON_LAST_STATE	RO	RTC state machine has previously entered power-down (ST_OFF) from the following states: 4'h3 = Entered power-down from power-on (ST_ON) 4'h9 = Entered power-down from suspend (ST_SUSP) Others = 7-second forced reset occurred After system restart, software can determine the reason for the previous power-down by reading this register.	
15:4	Reserved			
31:16	ST_OFF_REASON_LAST_INPUT	RO	Reasons for the state machine entering power-down (record signal state value): [14:0] same as ST_ON_REASON_LAST_INPUT [15] = 0: Forced reset occurred in 7 seconds.	

RTC_EN_WAKEUP_REQ

Offset Address: 0x120

Bits	Name	Access	Description	Reset
0	RTC_EN_WAKEUP_REQ	R/W	Enable event request to wake up from suspend state: 0 = disable, 1 = enable	0x0
1	RTC_EN_POWERUP_REQ	R/W	Enabled event request power-on: 0 = disable, 1 = enable	0x0
31:2	Reserved			

RTC_PWR_WAKEUP_POLARITY

Offset Address: 0x128

Bits	Name	Access	Description	Reset
0	PWR_WAKEUP0_POLARITY	R/W	Select PWR_WAKEUP0 active polarity: 1 = High level active 0 = Low level active	0x1
1	PWR_WAKEUP1_POLARITY	R/W	Select PWR_WAKEUP1 active polarity: 1 = High level active 0 = Low level active	0x1
31:2	Reserved			

RTC_DB_SEL_REQ

Offset Address: 0x130

Bits	Name	Access	Description	Reset
0	DB_SEL_REQ_SHDN	R/W	Select debounce mode for software signal REQ_SHDN: 0 = Triggered by rising edge of register value 1 = Triggered by pulse signal of register	0x1
1	DB_SEL_REQ_THM_SHDN	R/W	Select debounce mode for signal REQ_THM_SHDN: 0 = triggered by high level of signal 1 = triggered by rising edge of signal	0x1
2	DB_SEL_REQ_PWR_CYC	R/W	Select debounce mode for software	0x1

Bits	Name	Access	Description	Reset
			signal REQ_PWR_CYC: 0 = Triggered by rising edge of register value 1 = Triggered by pulse signal of register	
3	DB_SEL_REQ_WARM_RST	R/W	Select debounce mode for software signal REQ_WARM_RST: 0 = Triggered by rising edge of register value 1 = Triggered by pulse signal of register	0x1
4	DB_SEL_REQ_WDG_RST	R/W	Select debounce mode for signal REQ_WDG_RST: 0 = Triggered by signal high level 1 = Triggered by rising edge of signal.	0x1
5	DB_SEL_REQ_SUSPEND	R/W	Select debounce mode for software signal REQ_SUSPEND: 0 = Triggered by rising edge of register value 1 = Triggered by pulse signal of register	0x1
6	DB_SEL_REQ_WAKEUP	R/W	Select debounce mode for signal REQ_WAKEUP: 0 = Triggered by signal high level 1 = Triggered by rising edge of signal.	0x1
7	DB_SEL_REQ_POWERUP	R/W	Select debounce mode for signal REQ_POWERUP: 0 = Triggered by signal high level 1 = Triggered by rising edge of signal.	0x1
31:8	Reserved			

RTC_PWR_DET_SEL

Offset Address: 0x140

Bits	Name	Access	Description	Reset
0	pwr_det_o_sel_comp	R/W	Select the source of the status signal output for low-voltage detection: 0 = Directly from IO PWR_VBAT_DET 1 = From the analog low-voltage detection circuit output The low-voltage detection status value can be read from register RTC_CTRL_STATUS0[0].	0x0
1	pwr_det_i_sel_comp	R/W	Select the source of low-voltage trigger power-off signal for the RTC state machine: 0 = Directly from IO PWR_VBAT_DET 1 = From analog low-voltage detection circuit output	0x0
31:2	Reserved			

3.9.6.2 RTC_MACRO_CTRL

RTC_PWR_DET_COMP

Offset Address: 0x44

Bits	Name	Access	Description	Reset
0	pwr_det_comp_enable	R/W	Enable analog module low voltage detection: 1 = enable 0 = disable	0x0
7:1	Reserved			
12:8	pwr_det_comp_sel	R/W	Setting low voltage detection voltage comparison threshold. Threshold = 1.20V + (pwr_det_comp_sel * 12.5mV)	0xf
31:13	Reserved			

RTC_MACRO_DA_CLEAR_ALL

Offset Address: 0x080

Bits	Name	Access	Description	Reset
0	DA_CLEAR_ALL	R/W		0x0
31:1	Reserved			

RTC_MACRO_DA_SET_ALL

Offset Address: 0x084

Bits	Name	Access	Description	Reset
0	DA_SEL_ALL	R/W		0x0
31:1	Reserved			

RTC_MACRO_DA_LATCH_PASS

Offset Address: 0x088

Bits	Name	Access	Description	Reset
0	DA_LATCH_PASS	R/W		0x0
31:1	Reserved			

RTC_MACRO_DA_SOC_READY

Offset Address: 0x08c

Bits	Name	Access	Description	Reset
0	DA_SOC_READY	R/W		0x0
31:1	Reserved			

RTC_MACRO_PD_SLDO

Offset Address: 0x090

Bits	Name	Access	Description	Reset
0	PD_SLDO	R/W		0x0

Bits	Name	Access	Description	Reset
31:1	Reserved			

RTC_MACRO_RG_DEFD

Offset Address: 0x094

Bits	Name	Access	Description	Reset
15:0	RG_DEFD	R/W		0x7fff
31:16	Reserved			

RTC_MACRO_RG_SET_T

Offset Address: 0x098

Bits	Name	Access	Description	Reset
31:0	RG_SET_T	R/W		0x0

RTC_MACRO_RO_CLK_STOP

Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
0	RO_CLK_STOP	RO		
31:1	Reserved			

RTC_MACRO_RO_DEFQ

Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
15:0	RO_DEFQ	RO		
31:16	Reserved			

RTC_MACRO_RO_T

Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
31:0	RO_T	RO		

3.9.6.3 RTC_CTRL

RTC_CTRL0_UNLOCKKEY

Offset Address: 0x004

Bits	Name	Access	Description	Reset
15:0	rtc_ctrl0_unlockkey	R/W	The value 0xab18 must be written to this register to unlock the write permission of register RTC_CTRL0. If unlockkey_clear is set to 1, the register value will be automatically cleared to 0 after a write operation to RTC_CTRL0, and RTC_CTRL0 will return	0x0000

Bits	Name	Access	Description	Reset
			to write lock.	
31:16	Reserved			

RTC_CTRL0

Offset Address: 0x008

Bits	Name	Access	Description	Reset
0	req_shdn	R/W	request power down 0 = no action, 1 = request to RTC Register RTC_ EN_ SHDN_ REQ must be set to 1.	0x0
1	req_sw_thm_shdn	R/W	Software mode request thermal shutdown 0 = no action, 1 = request to RTC Register RTC_ EN_ THM_ SHDN must be set to 1.	0x0
2	hw_thm_shdn_en	R/W	Enable hardware thermal shutdown 0 = diable, 1 = enable Register RTC_ EN_ THM_ SHDN must be set to 1.	0x0
3	req_pwr_cyc	R/W	Request power cycle 0 = no action, 1 = request to RTC Register RTC_ EN_ PWR_ CYC_ REQ must be set to 1.	0x0
4	req_warm_rst	R/W	Request Warm-reset 0 = no action, 1 = request to RTC Register RTC_EN_WARM_RST_REQ must be set to 1.	0x0
5	req_sw_wdg_rst	R/W	Software mode request Watchdog reset 0 = no action, 1 = request to RTC Register RTC_EN_WDG_RST_REQmust be set to 1 to be valid.	0x0
6	hw_wdg_rst_en	R/W	Enable hardware mode Watchdog reset 0 = diable, 1 = enable	0x0
7	req_suspend	R/W	Request suspend 0 = no action, 1 = request to RTC Register RTC_ EN_ SUSPEND_ REQ must be set to 1.	0x0
8	unlockkey_clear	R/W	Enable auto clear register unlock	0x0
9	Reserved			
10	reg_rtc_mode	R/W	The source of 32K clock 0 = OSC32K, 1 = XTAL32K	0x0
11	reg_clk32k_cg_en	R/W	The switch of 32K clock 0 = close, 1 = open	0x1
31:12	Reserved			

RTC_CTRL_STATUS0

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
0	rtc_pwr_vbat_det_o	RO	Low voltage detection status signal output	
1	rtc_pwr_button0_o	RO	PWR_BUTTON0 IO signal output	
2	rtc_pwr_button1_o	RO	PWR_BUTTON1 IO signal output	
3	Reserved			

Bits	Name	Access	Description	Reset
4	rtc_pwr_on_o	RO	PWR_ON IO signal output	
5	rtc_pwr_wakeup0_o	RO	PWR_WAKEUP0 IO signal output	
6	rtc_pwr_wakeup1_o	RO	PWR_WAKEUP1 IO signal output	
7	rtc_mode_o	RO	RTC_MODE IO signal output	
20:8	Reserved			
21	rtc_alarm_o	RO	Alarm status	
22	hw_thm_shdn_sta_i	RO	Thermal shutdown status signal	
23	hw_wdg_rst_sta_i	RO	Watchdog reset status signal	
24	sys_reset_x_i	RO		
25	cg_en_out_clk_32k	RO		
29:26	rtc_fsm_st	RO	Value of the RTC state machine.	
31:30	Reserved			

RTC_CTRL_STATUS1

Offset Address: 0x010

Bits	Name	Access	Description	Reset
31:0	rtc_sec_value_o	RO	RTC second counter value	

rtc_ctrl_status2gpio

Offset Address: 0x014

Bits	Name	Access	Description	Reset
7:0	status2gpio_en	R/W		0x0
31:8	Reserved			

rtcsys_rst_ctrl

Offset Address: 0x018

Bits	Name	Access	Description	Reset
0	Reserved			
1	reg_soft_rstn_mcu	R/W	0 : reest MCU	0x0
2	reg_soft_rstn_sdio	R/W	0 : reset SD1	0x1
3	reg_soft_rstn_uart	R/W	0 : reset Uart	0x1
4	reg_soft_rstn_spinor	R/W	0 : reset spinor1	0x1
5	reg_soft_rstn_ictl	R/W	0 : reset dw_ictl	0x1
6	reg_soft_rstn_mbox	R/W	0 : reset mbox	0x1
7	reg_soft_rstn_fab_hs2rtc	R/W	0 : reset hs2rtc	0x1
8	reg_soft_rstn_fab_rtc2ap	R/W	0 : reset rtc2ap	0x1
9	reg_soft_rstn_fab_sram	R/W	0 : reset ahb sram logic	0x1
10	reg_soft_rstn_apb	R/W	no load	0x1
11	reg_soft_rstn_apb_timer	R/W	0 : reset dw timer apb logic	0x1
12	reg_soft_rstn_timer0	R/W	0 : reset dw timer0	0x1
13	reg_soft_rstn_timer1	R/W	0 : reset dw timer1	0x1
14	reg_soft_rstn_osc	R/W	0 : reset osc	0x1
15	reg_soft_rstn_gpio	R/W	0 : reset gpio	0x1
16	reg_soft_rstn_i2c	R/W	0 : reset i2c	0x1
17	reg_soft_rstn_saradc	R/W	0 : reset saradc	0x1
18	reg_soft_rstn_wdt	R/W	0 : reset wdt	0x1
19	reg_soft_rstn_irrx	R/W	0 : reset irrx	0x1
20	reg_soft_rstn_f32kless	R/W	0: reset f32kless	0x1
31:21	Reserved			

rtcsys_clkmux

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
3:0	reg_sdio_clk_mux	R/W	clk_sd1_pre 0 : fpll/4 1: osc_div	0x0
7:4	reg_fab_clk_mux	R/W	clk_fab_pre 0 : 32K, 1: fpll/5, 2: osc_div	0x0
9:8	reg_timer0_clk_mux	R/W	0: xtal 1: 32K	0x0
11:10	reg_timer1_clk_mux	R/W	0: xtal 1: 32K	0x0
13:12	reg_apb_clk_mux	R/W	00 : cgdiv and refer to apbactive 01 : force clk_apb, clk_fab 1:1 (default) 10 : force clk_apb, clk_fab 1:2 11 : force clk_apb, clk_fab 1:4	0x1
15:14	Reserved			
17:16	reg_i2c_clk_mux	R/W	0: xtal 1: osc div	0x0
19:18	reg_sd_mclk_clk_mux	R/W	0: 100Khz from OSC, 1: 32K	0x0
20	reg_saradc_clk_mux	R/W	0 : XTAL, 1: OSC DIV	0x0
21	reg_irrx_clk_mux	R/W	0 : XTAL, 1: OSC DIV	0x0
31:22	Reserved			

rtcsys_mcu51_ctrl0

Offset Address: 0x020

Bits	Name	Access	Description	Reset
4:0	reg_51_rom_addr_size	R/W	Determines how many of the sixteen internal ROM address bits (irom_addr) are used (0 = no internal ROM present);	0xc
5	reg_51_mem_ea_n	R/W	0 : external rom exist, 1: external rom not exist	0x0
6	reg_51_xdata_mode	R/W	0 : fetch xdata with clock gating 1 : fetch xdata wo clock gating (to support 51 timer and 51 uart)	0x0
7	reg_51_rom_addr_def	R/W	0: mercury define , max internal rom = 2^reg_51_rom_addr_size -1 internal rom offset = 4K*reg_51irom_ioffset 1: mars define , max internal rom = 2K*reg_51_rom_addr_size -1 internal rom offset = 2K*reg_51irom_ioffset	0x0
10:8	Reserved			
31:11	reg_51xdata_ioffset0	R/W	Set offset address[31:12] to select mcu8051 boot device	0x05200

rtcsys_mcu51_ctrl1

Offset Address: 0x024

Bits	Name	Access	Description	Reset
4:0	reg_51irom_ioffset	R/W	boot rom offset to rtcsys_sram	0x0
5	Reserved			
9:6	reg_51_pf_mode	R/W	reg_51_pf_mode	0x0
10	Reserved			
31:11	reg_51xdata_doffset0	R/W	Set offset address[31:12] to select mcu8051 xdata	0x05200

rtcsys_pmu

Offset Address: 0x028

Bits	Name	Access	Description	Reset
3:0	Reserved			
4	reg_dis_pmu_ldo_ctrl	R/W	disable pmu ldo ctrl 0: enable pmu to ctrl RTC_LDO sleep mode 1: disable pmu to ctrl RTC_LDO sleep mode	0x0
5	reg_wdt_clkoff_by_pmu	R/W	wdt_clk gate by pmu when mcu into idle mode 1. wdt clock gate by pmu	0x0
6	reg_force_osc_off	R/W	1 : force osc off	0x0
7	reg_force_osc_on	R/W	1 : force osc on	0x0
8	reg_pmu_sleep_mode	R/W	pmu enter light sleep mode when mcu idle 1 : enable pmu light sleep mode when mcu idle (pmu control osc_req/ sram slp) 0 disable pmu light sleep mode	0x0
9	reg_pmu_lowpwr_mode	R/W	mcu_pmu into sleep state when rtc at suspend state & mcu idle & reg_pmu_sleep_mode enable 1 : enable mcu_pmu into sleep mode (trigger rtc ldo step down power) 0 disable mcu_pmu sleep mode	0x0
13:10	reg_pmu_stable_cnt	R/W	Stable timer when mcu_pmu leave sleep state, clock unit : 31.25us (32khz), wait for 1~16 tick cycle	0x3
14	reg_xtal_off_by_pmu	R/W	pmu control xtal request 1: xtal request disable by pmu sleep mode	0x0
15	reg_rtcsys_clk25m_req	R/W	xtal request1 for rtcsys 0: disable 25m xtal request1(rtcsys) 1: enable 25m xtal request1 (rtcsys)	0x1
19:16	reg_rtc_vbat_det_db_cnt	R/W	vbat det int debounce time (cycle unit : 32K)	0x2
20	reg_rtc_vbat_det_db_en	R/W	0: disable vbat det int debounce 1: enable vbat det int debounce	0x1
21	reg_ahb_sram_auto_slp_en	R/W	1: enable ahb sram into slp md when bus idle	0x0
23:22	reg_ahb_sram_busy_sel	R/W	2'd0: cs | cs_d1 2'd1: cs | cs_d1 | cs_d2 2'd2: cs | cs_d1 | cs_d2 | cs_d3 3'd3: cs | cs_d1 | cs_d2 | cs_d3 | cs_d4	0x0
24	reg_rtc_stint_clr	W1P	clear rtc state change interrupt	
25	reg_vbat_det_int_clr	W1P	clear vbet det interrupt	
26	reg_rtcsys_clk25m_hw_req	R/W	xtal request1 for rtcsys from hw ip 0: disable 25m xtal request1 from hw ip(rtcsys) 1: enable 25m xtal request1 from hw ip(rtcsys)	0x0
27	Reserved			
28	reg_vbat_det_force_clk	R/W	1: when vbat det happen, change rtcsys bus clock to OSC	0x0

Bits	Name	Access	Description	Reset
29	reg_mcu_clkoff_by_pmu	R/W	mcu_clk gate by pmu when into idle mode 1. mcu clock gate by pmu	0x1
30	reg_xtal_off_by_susp	R/W	ISO off control xtal request 1: xtal request disable by ISO_OFF	0x0
31	reg_osc_off_by_susp	R/W	ISO off control osc request 1: osc request disable by ISO_OFF	0x0

rtcsys_status

Offset Address: 0x02c

Bits	Name	Access	Description	Reset
31:0	reg_rtcsys_status	RO	[0] enable rtc2apb ahb path 0: rtcsys ip can only access 0x05000000+16MB 1: rtcsys ip can access full range address [1] flag of vbat_det_force_clk	

rtcsys_clkbyp

Offset Address: 0x030

Bits	Name	Access	Description	Reset
31:0	reg_clk_byp	R/W	[0] : clk_fab , 0: clk_fab_pre, 1: xtal (default) [1] : clk_sdio, 1: clk_sd1_pre, 1: xtal (default) [31:2]: NA	0xffffffff

rtcsys_clk_en

Offset Address: 0x034

Bits	Name	Access	Description	Reset
31:0	reg_clk_en	R/W	[0]: NA [1]: clk_sd1 (sd1 card clock) [2]: clk_fab_sd1 (sd1 core clock) [3]: clk_mcu [4]: clk_hs2rtc_mst [5]: clk_rtc2ap_slv [6]: clk_spinor1 [7]: clk_fab_sram (AHB sram) [8]: NA [9]: clk_apb_timer [10]: clk_timer0 [11]: clk_timer1 [12]: clk_apb_uart [13]: clk_uart [14]: clk_apb_ictrl [15]: clk_apb_mbox [16]: clk_apb_gpio [17]: clk_apb_osc [18]: clk_gpio_db [19]: clk_apb_i2c [20]: clk_i2c [21]: NA [22]: clk_sd1_tmclk [23]: clk_apb_saradc [24]: clk_saradc [25]: clk_apb_wdt	0xffffffff

Bits	Name	Access	Description	Reset
			[26]: clk_wdt [27]: clk_irrx [31:28]: NA	

rtcsys_wkup_ctrl

Offset Address: 0x038

Bits	Name	Access	Description	Reset
14:0	reg_rtcsys_wkint_mask	R/W	mask int to RTC_CORE.REQ_WAKEUP/ MCU_PMU [0]: irrrx_intr [1]: gpio_int [2]: timer0_int [3]: timer1_int [4]: saradc_int [5]: rtcsys_ictrl_int [6]: wdt_int [7]: irrx_wakeup	0xff
15	reg_vbat_det_wkup_mask	R/W	1: mask vbat det int	0x1
16	reg_sw_wkint_req	R/W	mcu sw wakeup interrupt to RTC_CORE 1: interrupt active	0x0
23:17	Reserved			
24	reg_wkint2rtc_mask	R/W	1: mask wakeup int (rtcsys int) to RTC core	0x1
31:25	Reserved			

rtcsys_clkdiv

Offset Address: 0x03c

Bits	Name	Access	Description	Reset
3:0	reg_div_clk_osc_fab_div_val	R/W	Clock Divider Factor	0x1
4	reg_div_clk_osc_fab_dis	R/W	Clock gate	0x0
5	reg_div_clk_osc_fab_hwide	R/W	Select High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider	0x0
15:6	Reserved			
19:16	reg_div_clk_osc_i2c_div_val	R/W	Clock Divider Factor	0x1
20	reg_div_clk_osc_i2c_dis	R/W	Clock gate	0x0
21	reg_div_clk_osc_i2c_hwide	R/W	Select High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider	0x0
23:22	Reserved			
29:24	reg_div_clk_osc_saradc_div_val	R/W	Clock Divider Factor	0x1
30	reg_div_clk_osc_saradc_dis	R/W	Clock gate	0x0
31	reg_div_clk_osc_saradc_hwide	R/W	Select High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider	0x0

fc_coarse_en

Offset Address: 0x040

Bits	Name	Access	Description	Reset

Bits	Name	Access	Description	Reset
0	fc_coarse_en	R/W	Enable 32K coarse tuning. 0 = disable 1 = enable	0x0
31:1	Reserved			

fc_coarse_cal

Offset Address: 0x044

Bits	Name	Access	Description	Reset
15:0	fc_coarse_value	RO	32K coarse tuning counter value (unit: 25MHz clock). One 32K clock period counts as one unit in a 25MHz clock.	
31:16	fc_coarse_time	RO	32K coarse adjustment completion count.	

fc_fine_en

Offset Address: 0x048

Bits	Name	Access	Description	Reset
0	fc_fine_en	R/W	Enable 32K fine tuning. 0 = disable, 1 = enable	0x0
31:1	Reserved			

fc_fine_period

Offset Address: 0x04c

Bits	Name	Access	Description	Reset
15:0	fc_fine_period	R/W	32K fine tuning counting period (unit: 32K clock) Set how many 32K clock cycles to count each time using the 25MHz clock.	0x0100
31:16	Reserved			

fc_fine_cal

Offset Address: 0x050

Bits	Name	Access	Description	Reset
23:0	fc_fine_value	RO	32K Fine-Tuning Counter Value (in units of 25MHz clock) - The 25MHz clock counts one fc_fine_period period.	
31:24	fc_fine_time	RO	32K fine adjustment completion count.	

rtcsys_pmu2

Offset Address: 0x054

Bits	Name	Access	Description	Reset
0	reg_rtc_sys_wkint_db_en	R/W	PMU wakeup int debounce enable	0x1
4:1	reg_rtc_sys_wkint_db_cnt	R/W	PMU wakeup int debounce cycle (32K)	0x2
31:5	Reserved			

rtcsys_clkdiv1

Offset Address: 0x058

Bits	Name	Access	Description	Reset
15:0	Reserved			
21:16	reg_div_clk_osc_irrx_div_val	R/W	Clock Divider Factor	0x0

Bits	Name	Access	Description	Reset
22	Reserved			
23	reg_div_clk_osc_irrx_dis	R/W	Clock gate	0x1
31:24	Reserved			

rtcsys_mcu51_dbg

Offset Address: 0x05c

Bits	Name	Access	Description	Reset
3:0	reg_51_dbg_sel	R/W	select mcu51 debug bus (check mcu design review ppt)	0x0
4	reg_51_dbg_snap_shot	W1P	snap shot mcu51 internal register to dbg register (reg_rtcsys_dbg)	
5	reg_51_dbg_step_en	R/W	0: disable mcu debug function 1: enable mcu debug function, and mcu stop at current PC	0x0
6	reg_51_dbg_step	W1P	1: mcu jump to next PC	
7	reg_51_dbg_jump	W1P	1: mcu jump to target pc value (reg_51_dbg_jump2pc)	
15:8	Reserved			
31:16	reg_51_dbg_jump2pc	R/W	16 bit mcu target pc value	0x0

sw_reg0

Offset Address: 0x060

Bits	Name	Access	Description	Reset
7:0	sw_reg0	R/W	reg for SW	0x0
31:8	Reserved			

sw_reg1_por

Offset Address: 0x064

Bits	Name	Access	Description	Reset
7:0	sw_reg1_por	R/W	reg for SW could only be reset by power reset	0x0
31:8	Reserved			

fab_lp_ctrl

Offset Address: 0x068

Bits	Name	Access	Description	Reset
7:0	rtcsys_fab_busy_sel	R/W	select signal to request sys_ctrl to speed up fab clock	0xDF
9:8	rtcsys_fab_busy_ctrl	R/W	rtcsys_fab_busy signal is combi or register out	0x0
11:10	apdbg_busy_ctrl	R/W	apdbg_busy signal is combi or register out	0x0
13:12	reg_apb_busy_ctrl	R/W	apb bridge_busy signal is combi or register out	0x3
15:14	reg_mcu_busy_ctrl	R/W	mcu_busy signal is combi or register out	0x3
31:16	Reserved			

rtcsys_mcu51_ictrl1

Offset Address: 0x07c

Bits	Name	Access	Description	Reset
15:0	reg_51_int1_src_mask	R/W	select rtcsys_int src to mcu int1_n 1: mask, 0: un-mask [0]: vbat_det [1]: mbox0_int [2]: NA [3]: irrx_int [4]: gpio_int [5]: uart_int [6]: spinor1_int [7]: timer0_int [8]: timer1_int [9]: irq_ap2rtc[0] [10]: irq_ap2rtc[1] [11]: i2c_int [12]: rtc_state_change_int [13]: hw_thm_shdn [14]: saradc [15]: wdt_int	0xffff
31:16	reg_51_int1_final_status	R0	mcu int1_n status [0]: vbat_det [1]: mbox0_int [2]: NA [3]: irrx_int [4]: gpio_int [5]: uart_int [6]: spinor1_int [7]: timer0_int [8]: timer1_int [9]: irq_ap2rtc[0] [10]: irq_ap2rtc[1] [11]: i2c_int [12]: rtc_state_change_int [13]: hw_thm_shdn [14]: saradc [15]: wdt_int	

rtc_ip_pwr_req

Offset Address: 0x080

Bits	Name	Access	Description	Reset
0	reg_sd1_pwr_req	R/W	power fence control 1: power on, 0: power off [0]: sd1	0x1
1	reg_sd1_pwr_req_2nd	R/W	power fence control 1: power on, 0: power off [0]: sd1	0x1
2	reg_mcu_pwr_req	R/W	power fence control 1: power on, 0: power off [1]: mcu subsys	0x1
3	reg_mcu_pwr_req_2nd	R/W	power fence control 1: power on, 0: power off [1]: mcu subsys	0x1
15:4	Reserved			
16	reg_sd1_pwr_ack	R0	power fence power status 1: power on, 0: power off [0]: sd1	
17	reg_sd1_pwr_ack_2nd	R0	power fence power status	

Bits	Name	Access	Description	Reset
			1: power on, 0: power off [0]: sd1	
18	reg_mcu_pwr_ack	R0	power fence power status 1: power on, 0: power off [1]: mcu subsys	
19	reg_mcu_pwr_ack_2nd	R0	power fence power status 1: power on, 0: power off [1]: mcu subsys	
31:20	Reserved			

rtc_ip_iso_ctrl

Offset Address: 0x084

Bits	Name	Access	Description	Reset
0	reg_sd1_iso_en	R/W	sd1 iso enablel 1: iso enable, 0: iso disable	0x0
1	reg_mcu_iso_en	R/W	mcu iso enablel 1: iso enable, 0: iso disable	0x0
15:2	Reserved			
17:16	reg_ip_por_en	R/W	1: pwr_island reset assert when power ack is 0	0x3
31:18	Reserved			

rtcsys_wkup_ctrl1

Offset Address: 0x094

Bits	Name	Access	Description	Reset
7:0	reg_rtcsys_wkint_final_status	RO	wkint final status [0]: sd1_wakeup_intr [1]: gpio_int [2]: timer0_int [3]: timer1_int [4]: saradc_int [5]: rtcsys_ictrl_int [6]: NA [7]: NA	
31:8	Reserved			

rtcsys_sram_ctrl

Offset Address: 0x098

Bits	Name	Access	Description	Reset
0	reg_ahb_sram_slp	R/W	1 : ahb sram into sleep mode	0x0
1	reg_ahb_sram_sd	R/W	1 : ahb sram into shut down mode	0x0
2	reg_ahb_sram_ctrl_ov	R/W	0 : ahb sram ctrl by PMU FSM and ahb sram busy 1: sram ctrol by register reg_ahb_sram_slp/reg_ahb_sram_sd	0x1
3	reg_sdio_sram_slp	R/W	1 : sdio sram into sleep mode	0x0
4	reg_sdio_sram_sd	R/W	1 : sdio sram into shut down mode	0x0
5	reg_sdio_sram_ctrl_ov	R/W	0 : sram's sd pin = 1'b0 1: sram ctrol by register reg_sdio_sram_sd	0x1
6	reg_mcu_sram_slp	R/W	1 : mcu iram sram into sleep mode	0x0
7	reg_mcu_sram_sd	R/W	1 : mcu iram sram into shut down mode	0x0
8	reg_mcu_sram_ctrl_ov	R/W	0 : mcu iram sram ctrl by PMU FSM 1: sram ctrol by register	0x1

Bits	Name	Access	Description	Reset
			reg_ahb_sram_slp/reg_ahb_sram_sd	
9	reg_rtc_sram_slp	R/W	1 : mcu iram sram into sleep mode	0x0
10	reg_rtc_sram_sd	R/W	1 : mcu iram sram into shut down mode	0x0
11	reg_rtc_sram_ctrl_ov	R/W	0 : mcu iram sram ctrl by PMU FSM 1: sram ctrol by register reg_ahb_sram_slp/reg_ahb_sram_sd	0x1
27:12	Reserved			
28	reg_mcu_sram_force_ce	R/W	1: force mcu_iram cs = 1	0x1
31:29	Reserved			

rtcsys_io_ctrl

Offset Address: 0x09c

Bits	Name	Access	Description	Reset
0	reg_i2c_mux_opt0	R/W	0: pwr_gpio6/8 control by dw_gpio 1: pwr_gpio6 is PWR_IIC_SDA pwr_gpio8 is PWR_IIC_SCL	0x0
31:1	Reserved			

rtcsys_wdt_ctrl

Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
0	reg_rtc_hw_wdg_rst_en	R/W	0: disable rtc wdt trigger warm reset or pwrcyc reset 1: enable rtc wdt trigger warm reset or pwrcyc reset	0x0
1	reg_rtc_wdt_ctrl_mask_en	R/W	no load	0x1
31:2	Reserved			

rtcsys_irrx_clk_ctrl

Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
0	reg_irrx_clk_sw_force_on	R/W	force on clk ctrl of irrx	0x1
1	reg_irrx_xtal_req_en	R/W	enable irrx clk ctrl requet XTAL	0x0
2	reg_irrx_osc_req_en	R/W	enable irrx clk ctrl requet OSC	0x0
3	reg_irrx_ldo_req_en	R/W	enable irrx clk ctrl requet LDO	0x0
7:4	Reserved			
15:8	reg_irrx_xtal_filter_cyc	R/W	irrx xtal filter cycle (default 2ms)	0x40
19:16	reg_irrx_clk_ctrl_st	RO	irrx clock ctrol state	
31:20	Reserved			

rtcsys_rtc_wkup_ctrl

Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
7:0	reg_rtc_wkint_mask	R/W	wakeup source mask int to RTC_CORE [0]: irrrx_intr [1]: gpio_int [2]: timer0_int [3]: timer1_int [4]: saradc_int [5]: rtcsys_ictrl_int [6]: wdt_int [7]: irrx_wakeup	0xff

Bits	Name	Access	Description	Reset
15:8	Reserved			
23:16	reg_rtc_puint_mask	R/W	power-up source mask int to RTC_CORE [0]: irrrx_intr [1]: gpio_int [2]: timer0_int [3]: timer1_int [4]: saradc_int [5]: rtcsys_ictrl_int [6]: wdt_int [7]: irrx_wakeup	0xff
31:24	Reserved			

rtcsys_por_rst_ctrl

Offset Address: 0x0ac

Bits	Name	Access	Description	Reset
0	reg_rtcsys_reset_en	R/W	0: not allow rtcsys reset by pwr cyc/ wdt warm reset 1 : allow rtcsys reset by pwr cyc/ wdt warm reset	0x0
1	reg_rtcsys_rstn_src_sel	R/W	select rtcsys rstn src 0: rtc_core fsm (reset with die domain) 1: por_pwr_rstn	0x0
31:2	Reserved			

3.10 Power management and low power consumption mode

3.10.1 Overview

The CV180ZB/CV1800B/CV1801B chip supports two main power modes.

a. Active

3.10.1.1 Active Mode

Active mode is the state where the chip is fully awake and operational. However, there are still power-saving techniques such as dynamic frequency scaling or dynamic clock gating.

3.10.2 Clock control

3.10.2.1 Turn off unwanted clock dividers

Refer to the clock configuration section, turn off the unused clock divider according to the clock source required by each module. To achieve the purpose of saving power consumption.

3.10.2.2 Adjust the working frequency of the module

According to the required clock specifications of each module, choose a lower clock source. The frequency division configuration is more to reduce the working frequency of the module. It is the first mock exam that the frequency of a single module is not necessarily reduced.

3.10.2.3 Module level low power control

Analog module: Mipi / USB / eth / aud related register settings, will not use the module off or into low power consumption mode.
Digital module: according to the hardware and specifications, turn off the clock of unnecessary digital module

3.10.2.4 Turn off unused PLL

Referring to PLL configuration, you can powerdown the PLL you don't need to use to save power.

3.10.3 DDR low power consumption control

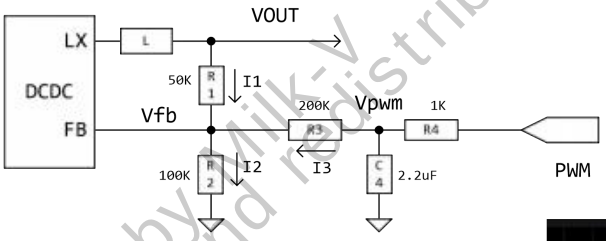
After the bus has not been accessed for a period of time, the DDR controller will automatically enter the state of self refresh and power down to reduce the system power consumption.
In some scenarios, due to intermittent access, it is impossible to find enough space to enter self refresh. In this case, we can also consider building a statistics register of the amount of data accessed to confirm whether the band is excessive. We can consider direct frequency reduction.

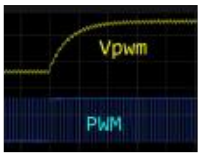
DDR controller supports dynamic frequency adjustment. However, since adjusting the frequency will temporarily stop DDR access for a period of time, in order to reduce the interruption time, it may cause real-time application buffer underflow / overflow. It is limited to 50% and 100%.

3.10.4 Voltage regulation

CV180ZB/CV1800B/CV1801B uses PWM0 to control VDDC voltage regulation by default.

The following is an example of using PWM to control the output voltage of DCDC






```
VOUT = Vfb + R1(I2-I3)
     = Vfb + R1(Vfb/R2-(Vpwm-Vfb)/R3)
     = Vfb + R1(Vfb/R2-(VDDIOxduty-Vfb)/R3)
```

Vfb	R1	R2	R3	VDDIO
0.6	50,000	100,000	200,000	1.8

Duty	Vpwm	Vout
0.00%	0.00	1.05
5.00%	0.09	1.03
10.00%	0.18	1.01
15.00%	0.27	0.98
20.00%	0.36	0.96
25.00%	0.45	0.94
30.00%	0.54	0.92
35.00%	0.63	0.89
40.00%	0.72	0.87
45.00%	0.81	0.85
50.00%	0.90	0.83
55.00%	0.99	0.80
60.00%	1.08	0.78
65.00%	1.17	0.76
70.00%	1.26	0.74
75.00%	1.35	0.71
80.00%	1.44	0.69
85.00%	1.53	0.67
90.00%	1.62	0.65
95.00%	1.71	0.62
100.00%	1.80	0.60

```
5 RC = 5x1Kx2.2uF = 11ms
(Vpwm settle time)

Vpwm = VDDIO x PWM duty cycle
```

Chart 3- 9 Example of using PWM to control DCDC voltage.

3.11 Chip internal temperature detection

Temperature sensor is built in the chip. Please refer to 12.8 for details
High junction temperature may cause thermal run away and cause permanent damage. So the chip needs to control the temperature

The first stage is software behavior
The temperature sensor can automatically detect whether the temperature exceeds a specific temperature and send out an overheat interrupt. After receiving the overheat interrupt, the software can reduce the power consumption and temperature by limiting the frequency or voltage of the high-power module and starting the fan. If the temperature returns to the safe range, the limit is removed

The second stage is hardware behavior
If the temperature continues to rise after the software is started, the hardware will intervene in the emergency of thermal shut-down. However, this function is turned off by default. After the software is started, it needs to set the relevant settings and then enable

3.12 8051 subsystem

3.12.1 Overview

The 8051 subsystem is located in a module that is independently powered by the RTC. The subsystem is configured with an 8051, an I2C/UART/SPI NOR/SD controller, a Timer/WDT, interrupt management, and a Mailbox IP. The system software can use the 8051 to manage wake-up conditions and wake up the system while it is in sleep mode, and communicate with external devices through peripheral controllers.

3.12.2 Characteristics

Configuration of the subsystem:

- The 8051 microprocessor has the following features:
  - Supports standard 8051 instruction set
  - Frequency range of 25MHz ~ 300 MHz
  - Debugging functions: single-step execution / jump2pc / snapshot PSW, DPTR, PC
  - Supports 32-bit data access
  - Reset vector can be configured to system AHB SRAM / DRAM / SPINOR
  - Supports WFI (Clock gating)
  - Supports code banking (maximum 64x64 KB)
- Provides 8KB of AHB SRAM space, which can be used by 8051 as instruction TCM or temporary storage for data
- Provides 2 sets of 32-bit counters for timing and counting functions, which can be used by applications to implement timing and counting, or by the operating system to implement system clocks.
- Provides 1 set of WDT for system interrupt or reset signal after a certain period of time in case of system exception.
- Provides 1 set of external SPINOR control.
- Provides interrupt controller for managing interrupt sources.
- Provides 2 sets of Mailbox for communication between ACPU and 8051.
- Provides 1 set of I2C.
- Provides 1 set of UART.

3.12.3 Function Description

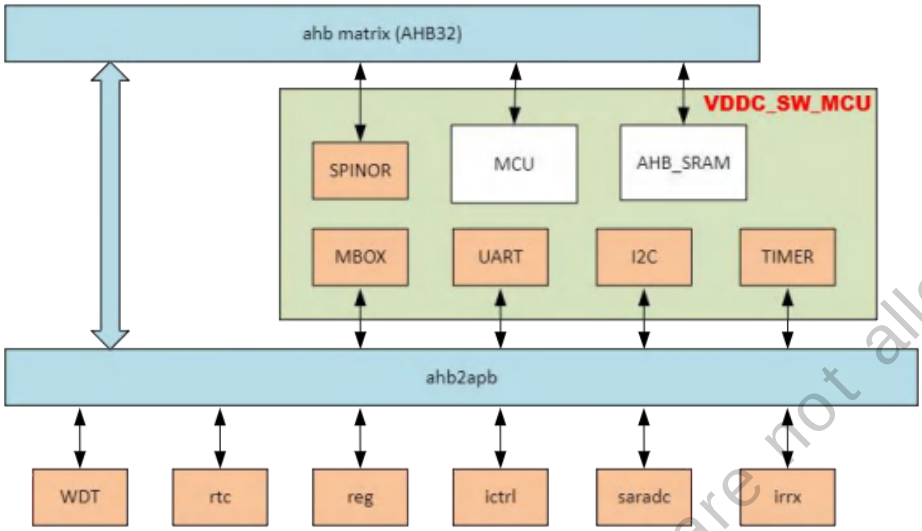


Figure 3-10 the architecture of 8051 subsystem

The subsystem is divided into two power domains: the AO domain and the MCU domain (green area). The system can use registers to select the power domain of the MCU to achieve power-saving requirements.

In sleep mode, the system can handle interrupts through the MCU and wake up the system by configuring registers. It can also communicate with external devices through I2C/UART.

3.12.4 Working Mode

3.12.4.1 Power Domain Control Flow

The subsystem is divided into three power domains: RTC domain (Always on), MCU domain. The power on/off process of the MCU domain can be controlled by configuring the registers, which follow the steps below:

MCU power_off>

1. Set the software reset register to 0.
2. Set the isolation enable register reg_mcu_iso_en to 1.
3. Set the power request register reg_mcu_pwr_req to 0.

MCU power_on=>

1. Set the power request register reg_mcu_pwr_req to 1.
2. Poll the power acknowledge register reg_mcu_pwr_ack until it is 1.

3. Set the isolation enable register reg_mcu_iso_en to 0.
4. Set the software reset register to 1.

3.12.4.2 8051 Initailization

The 8051 is in a reset state at the system initialization, and it can complete the following software flow through the ACPU using the 8051:

1. The 8051 is in the reset state (register reg_soft_rstn_mcu = 0).
2. Configure the register reg_mcu_rom_addr_size to determine the instruction TCM size.
3. Configure the register reg_51irom_ioffset to determine the location of TCM execution on AHB SRAM.
4. Release the 8051 reset state by setting the register reg_soft_rstn_mcu = 1.

3.12.4.3 Interrupt Handling

The 8051 can receive external level-triggered interrupts through the int0_n and int1_n interfaces. int0_n/int1_n can be selected to output interrupt signals to the 8051 from ictl (interrupt controller) and configuration register reg_51_int1_src_mask, respectively.

Interrupt Number	Interrupt Name	Interrupt Description
0	Vbat_det	System power-off interrupt
1	mbox_int0	Mailbox interrupt
2	NA	Reserved
3	irrx	Remote control reception interrupt
4	gpio_int	PWR GPIO interrupt
5	uart_int	PWR UART interrupt
6	spinor1_int	SPINOR1 interrupt
7	timer_int0	TIMER0 interrupt
8	timer_int1	TIMER1 interrupt
9	Irq_ap2rtc[0]	System interrupt
10	Irq_ap2rtc[1]	System interrupt

11	I2c_int	PWR I2C interrupt
12	st_change_int	RTC state change interrupt
13	hw_thm_shdn	System overheat interrupt
14	saradc	SARADC interrupt
15	wdt_int	Watchdog interrupt

Table 3-12 the interrupt list of 8051 subsystem

3.12.4.4 MAILBOX

Mailbox provides 2 sets of spinlock fields and 4 sets of 32-bit message fields, which allow ACPU/8051 to transfer information between each other.

3.12.5 8051 Subsystem Registers Overview

Table 3-12 RTC_CTRL registers overview(base address 0x05025000)

Name	Address Offset	Description
rtc_ctrl_version	0x000	rtc_ctrl_version
rtc_ctrl_unlockkey	0x004	rtc_ctrl_unlockkey
rtc_ctrl0	0x008	rtc_ctrl0
rtc_ctrl_status0	0x00c	rtc_ctrl_status0
rtc_ctrl_status1	0x010	rtc_ctrl_status1
rtc_ctrl_status2gpio	0x014	rtc_ctrl_status2gpio
rtcsys_rst_ctrl	0x018	rtcsys_rst_ctrl
rtcsys_clkmux	0x01c	rtcsys_clkmux
rtcsys_mcu51_ctrl0	0x020	rtcsys_mcu51_ctrl0
rtcsys_mcu51_ctrl1	0x024	rtcsys_mcu51_ctrl1
rtcsys_pmu	0x028	rtcsys_pmu
rtcsys_status	0x02c	rtcsys_status
rtcsys_clkbyp	0x030	rtcsys_clkbyp
rtcsys_clk_en	0x034	rtcsys_clk_en
rtcsys_wkup_ctrl	0x038	rtcsys_wkup_ctrl
rtcsys_clkdiv	0x03c	rtcsys_clkdiv
fc_coarse_en	0x040	fc_coarse_en
fc_coarse_cal	0x044	fc_coarse_cal
fc_fine_en	0x048	fc_fine_en
fc_fine_period	0x04c	fc_fine_period
fc_fine_cal	0x050	fc_fine_cal
rtcsys_pmu2	0x054	rtcsys_pmu2
rtcsys_clkdiv1	0x058	rtcsys_clkdiv
rtcsys_mcu51_dbg	0x05c	rtcsys_mcu51_dbg
sw_reg0	0x060	sw_reg0
sw_reg1_por	0x064	sw_reg1_por

Name	Address Offset	Description
fab_lp_ctrl	0x068	fab_lp_ctrl
fab_option	0x06c	fab_option
rtcsys_mcu51_ictrl1	0x07c	rtcsys_mcu51_ictrl1
rtc_ip_pwr_req	0x080	rtc_ip_pwr_req
rtc_ip_iso_ctrl	0x084	rtc_ip_iso_ctrl
rtcsys_wkup_ctrl1	0x094	rtcsys_wkup_ctrl1
rtcsys_sram_ctrl	0x098	rtcsys_sram_ctrl
rtcsys_io_ctrl	0x09c	rtcsys_io_ctrl
rtcsys_wdt_ctrl	0x0a0	rtcsys_wdt_ctrl
rtcsys_irrx_clk_ctrl	0x0a4	rtcsys_irrx_clk_ctrl
rtcsys_rtc_wkup_ctrl	0x0a8	rtcsys_rtc_wkup_ctrl
rtcsys_por_rst_ctrl	0x0ac	rtcsys_por_rst_ctrl

3.12.6 8051 Subsystem Registers

rtc_ctrl_unlockkey

Offset Address: 0x004

Bits	Name	Access	Description	Reset
15:0	rtc_ctrl_unlockkey	R/W	"rtc_ctrl0" could be write when unlockkey is set to be 0xAB18. "ptest_adc2ram_ctrl" could be write when unlockkey is set to be 0x0423. If unlockkey_clear is set to 1, the rtc_ctrl0_unlockkey will be clear after a apb write to rtc_ctrl0 or ptest_adc2ram_ctrl	0x0000
31:16	Reserved			

rtc_ctrl0

Offset Address: 0x008

Write Lock: wr_lock_rtc_ctrl0

Bits	Name	Access	Description	Reset
0	req_shdn	W1P	Mask: Enabled	
1	req_sw_thm_shdn	R/W	Mask: Enabled	0x0
2	hw_thm_shdn_en	R/W	Mask: Enabled	0x0
3	req_pwr_cyc	W1P	Mask: Enabled	
4	req_warm_rst	W1P	Mask: Enabled	
5	req_sw_wdg_rst	R/W	Mask: Enabled	0x0
6	hw_wdg_rst_en	R/W	Mask: Enabled	0x0

Bits	Name	Access	Description	Reset
7	req_suspend	W1P	Mask: Enabled	
8	unlockkey_clear	R/W	Mask: Enabled	0x0
9	Reserved			
10	reg_rtc_mode	R/W	Mask: Enabled	0x0
11	reg_clk32k_cg_en	R/W	Mask: Enabled	0x1
31:12	Reserved			

rtc_ctrl_status0

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
0	rtc_pwr_vbat_det_o	RO		
1	rtc_pwr_button0_o	RO		
2	rtc_pwr_button1_o	RO		
3	rtc_pwr_button1_7sec_o	RO		
4	rtc_pwr_on_o	RO		
5	rtc_pwr_wakeup0_o	RO		
6	rtc_pwr_wakeup1_o	RO		
7	rtc_mode_o	RO		
19:8	Reserved			
20	rtc_rstn_o	RO		
21	rtc_alarm_o	RO		
22	hw_thm_shdn_sta_i	RO		
23	hw_wdg_rst_sta_i	RO		
24	sys_reset_x_i	RO		
25	cg_en_out_clk_32k	RO		
29:26	rtc_fsm_st	RO		
31:30	Reserved			

rtc_ctrl_status1

Offset Address: 0x010

Bits	Name	Access	Description	Reset
31:0	rtc_sec_value_o	RO		

rtc_ctrl_status2gpio

Offset Address: 0x014

Bits	Name	Access	Description	Reset
7:0	status2gpio_en	R/W		0x0
31:8	Reserved			

rtcsys_rst_ctrl

Offset Address: 0x018

Bits	Name	Access	Description	Reset

Bits	Name	Access	Description	Reset
0	Reserved			
1	reg_soft_rstn_mcu	R/W	0 : reest MCU	0x0
2	reg_soft_rstn_sdio	R/W	0 : reset SD1	0x1
3	reg_soft_rstn_uart	R/W	0 : reset Uart	0x1
4	reg_soft_rstn_spinor	R/W	0 : reset spinor1	0x1
5	reg_soft_rstn_ictl	R/W	0 : reset dw_ictl	0x1
6	reg_soft_rstn_mbox	R/W	0 : reset mbox	0x1
7	reg_soft_rstn_fab_hs2rtc	R/W	0 : reset hs2rtc	0x1
8	reg_soft_rstn_fab_rtc2ap	R/W	0 : reset rtc2ap	0x1
9	reg_soft_rstn_fab_sram	R/W	0 : reset ahb sram logic	0x1
10	reg_soft_rstn_apb	R/W	no load	0x1
11	reg_soft_rstn_apb_timer	R/W	0 : reset dw timer apb logic	0x1
12	reg_soft_rstn_timer0	R/W	0 : reset dw timer0	0x1
13	reg_soft_rstn_timer1	R/W	0 : reset dw timer1	0x1
14	reg_soft_rstn_osc	R/W	0 : reset osc	0x1
15	reg_soft_rstn_gpio	R/W	0 : reset gpio	0x1
16	reg_soft_rstn_i2c	R/W	0 : reset i2c	0x1
17	reg_soft_rstn_saradc	R/W	0 : reset saradc	0x1
18	reg_soft_rstn_wdt	R/W	0 : reset wdt	0x1
19	reg_soft_rstn_irrx	R/W	0 : reset irrx	0x1
20	reg_soft_rstn_f32kless	R/W	0 : reset f32kless	0x1
31:21	Reserved			

rtcsys_clkmux

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
3:0	reg_sdio_clk_mux	R/W	clk_sd1_pre 0 : fpll/4 1: osc_div	0x0
7:4	reg_fab_clk_mux	R/W	clk_fab_pre 0 : 32K, 1: fpll/5, 2: osc_div	0x0
9:8	reg_timer0_clk_mux	R/W	0: xtal 1: 32K	0x0
11:10	reg_timer1_clk_mux	R/W	0: xtal 1: 32K	0x0
13:12	reg_apb_clk_mux	R/W	00 : cgdiv and refer to apbactive 01 : force clk_apb, clk_fab 1:1 (default) 10 : force clk_apb, clk_fab 1:2 11 : force clk_apb, clk_fab 1:4	0x1
15:14	Reserved			
17:16	reg_i2c_clk_mux	R/W	0: xtal 1: osc div	0x0
19:18	reg_sd_mclk_clk_mux	R/W	0: 100Khz from OSC, 1: 32K	0x0
20	reg_saradc_clk_mux	R/W	0 : XTAL, 1: OSC DIV	0x0
21	reg_irrx_clk_mux	R/W	0 : XTAL, 1: OSC DIV	0x0
31:22	Reserved			

rtcsys_mcu51_ctrl0

Offset Address: 0x020

Bits	Name	Access	Description	Reset
4:0	reg_51_rom_addr_size	R/W	Determines how many of the sixteen internal ROM address bits (irom_addr) are used (0 = no internal ROM present);	0xc

Bits	Name	Access	Description	Reset
5	reg_51_mem_ea_n	R/W	0 : external rom exist, 1: external rom not exist	0x0
6	reg_51_xdata_mode	R/W	0 : fetch xdata with clock gating 1 : fetch xdata wo clock gating (to support 51 timer and 51 uart)	0x0
7	reg_51_rom_addr_def	R/W	0: mercury define , max internal rom = 2^reg_51_rom_addr_size -1 internal rom offset = 4K*reg_51irom_ioffset 1: mars define , max internal rom = 2K*reg_51_rom_addr_size -1 internal rom offset = 2K*reg_51irom_ioffset	0x0
10:8	Reserved			
31:11	reg_51xdata_ioffset0	R/W	Set offset address[31:11] to select mcu8051 boot device	0x0A400

rtcsys_mcu51_ctrl1

Offset Address: 0x024

Bits	Name	Access	Description	Reset
4:0	reg_51irom_ioffset	R/W	boot rom offset to rtcsys_sram	0x0
5	Reserved			
9:6	reg_51_pf_mode	R/W	reg_51_pf_mode	0x0
10	Reserved			
31:11	reg_51xdata_doffset0	R/W	Set offset address[31:11] to select mcu8051 xdata	0x0A400

rtcsys_pmu

Offset Address: 0x028

Bits	Name	Access	Description	Reset
3:0	Reserved			
4	reg_dis_pmu_ldo_ctrl	R/W	disable pmu ldo ctrl 0: enable pmu to ctrl RTC_LDO sleep mode 1: disable pmu to ctrl RTC_LDO sleep mode	0x0
5	reg_wdt_clkoff_by_pmu	R/W	wdt_clk gate by pmu when mcu into idle mode 1. wdt clock gate by pmu	0x0
6	reg_force_osc_off	R/W	1 : force osc off	0x0
7	reg_force_osc_on	R/W	1 : force osc on	0x0
8	reg_pmu_sleep_mode	R/W	pmu enter light sleep mode when mcu idle 1 : enable pmu light sleep mode when mcu idle (pmu control osc_req/ sram slp) 0 disable pmu light sleep mode	0x0
9	reg_pmu_lowpwr_mode	R/W	mcu_pmu into sleep state when rtc at suspend state & mcu idle & reg_pmu_sleep_mode enable 1 : enable mcu_pmu into sleep mode (trigger rtc ldo step down power) 0 disable mcu_pmu sleep mode	0x0

Bits	Name	Access	Description	Reset
13:10	reg_pmu_stable_cnt	R/W	Stable timer when mcu_pmu leave sleep state, clock unit : 31.25us (32khz), wait for 1~16 tick cycle	0x3
14	reg_xtal_off_by_pmu	R/W	pmu control xtal request 1: xtal request disable by pmu sleep mode	0x0
15	reg_rtcsys_clk25m_req	R/W	xtal request1 for rtcsys 0: disable 25m xtal request1(rtcsys) 1: enable 25m xtal request1 (rtcsys)	0x1
19:16	reg_rtc_vbat_det_db_cnt	R/W	vbat det int debounce time (cycle unit : 32K)	0x2
20	reg_rtc_vbat_det_db_en	R/W	0: disable vbat det int debounce 1: enable vbat det int debounce	0x1
21	reg_ahb_sram_auto_slp_en	R/W	1: enable ahb sram into slp md when bus idle	0x0
23:22	reg_ahb_sram_busy_sel	R/W	2'd0: cs | cs_d1 2'd1: cs | cs_d1 | cs_d2 2'd2: cs | cs_d1 | cs_d2 | cs_d3 3'd3: cs | cs_d1 | cs_d2 | cs_d3 | cs_d4	0x0
24	reg_rtc_stint_clr	W1P	clear rtc state change interrupt	
25	reg_vbat_det_int_clr	W1P	clear vbet det interrupt	
26	reg_rtcsys_clk25m_hw_req	R/W	xtal request1 for rtcsys from hw ip 0: disable 25m xtal request1 from hw ip(rtcsys) 1: enable 25m xtal request1 from hw ip(rtcsys)	0x0
27	Reserved			
28	reg_vbat_det_force_clk	R/W	1: when vbat det happen, change rtcsys bus clock to OSC	0x0
29	reg_mcu_clkoff_by_pmu	R/W	mcu_clk gate by pmu when into idle mode 1. mcu clock gate by pmu	0x1
30	reg_xtal_off_by_susp	R/W	ISO off control xtal request 1: xtal request disable by ISO_OFF	0x0
31	reg_osc_off_by_susp	R/W	ISO off control osc request 1: osc request disable by ISO_OFF	0x0

rtcsys_status

Offset Address: 0x02c

Bits	Name	Access	Description	Reset
31:0	reg_rtcsys_status	RO	[0] enable rtc2apb ahb path 0: rtcsys ip can only access 0x05000000+16MB 1: rtcsys ip can access full range address [1] flag of vbat_det_force_clk	

rtcsys_clkbyp

Offset Address: 0x030

Bits	Name	Access	Description	Reset
31:0	reg_clk_byp	R/W	[0] : clk_fab , 0: clk_fab_pre, 1: xtal (default)	0xffffffff

Bits	Name	Access	Description	Reset
			[1] : clk_sdio, 1: clk_sd1_pre, 1: xtal (default) [31:2]: NA	

rtcsys_clk_en

Offset Address: 0x034

Bits	Name	Access	Description	Reset
31:0	reg_clk_en	R/W	[0]: NA [1]: clk_sd1 (sd1 card clock) [2]: clk_fab_sd1 (sd1 core clock) [3]: clk_mcu [4]: clk_hs2rtc_mst [5]: clk_rtc2ap_slv [6]: clk_spinor1 [7]: clk_fab_sram (AHB sram) [8]: NA [9]: clk_apb_timer [10]: clk_timer0 [11]: clk_timer1 [12]: clk_apb_uart [13]: clk_uart [14]: clk_apb_ictrl [15]: clk_apb_mbox [16]: clk_apb_gpio [17]: clk_apb_osc [18]: clk_gpio_db [19]: clk_apb_i2c [20]: clk_i2c [21]: NA [22]: clk_sd1_tmclk [23]: clk_apb_saradc [24]: clk_saradc [25]: clk_apb_wdt [26]: clk_wdt [27]: clk_irrx [31:28]: NA	0xffffffff

rtcsys_wkup_ctrl

Offset Address: 0x038

Bits	Name	Access	Description	Reset
14:0	reg_rtcsys_wkint_mask	R/W	mask int to RTC_CORE.REQ_WAKEUP/ MCU_PMU [0]: irrrx_intr [1]: gpio_int [2]: timer0_int [3]: timer1_int [4]: saradc_int [5]: rtcsys_ictrl_int [6]: wdt_int [7]: irrx_wakeup	0xff
15	reg_vbat_det_wkup_mask	R/W	1: mask vbat det int	0x1
16	reg_sw_wkint_req	R/W	mcu sw wakeup interrupt to RTC_CORE 1: interrupt active	0x0
23:17	Reserved			

Bits	Name	Access	Description	Reset
24	reg_wkint2rtc_mask	R/W	1: mask wakeup int (rtcsys int) to RTC core	0x1
31:25	Reserved			

rtcsys_clkdiv

Offset Address: 0x03c

Bits	Name	Access	Description	Reset
3:0	reg_div_clk_osc_fab_div_val	R/W	Clock Divider Factor	0x1
4	reg_div_clk_osc_fab_dis	R/W	Clock gate	0x0
5	reg_div_clk_osc_fab_hwide	R/W	Select High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider	0x0
15:6	Reserved			
19:16	reg_div_clk_osc_i2c_div_val	R/W	Clock Divider Factor	0x1
20	reg_div_clk_osc_i2c_dis	R/W	Clock gate	0x0
21	reg_div_clk_osc_i2c_hwide	R/W	Select High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider	0x0
23:22	Reserved			
29:24	reg_div_clk_osc_saradc_div_val	R/W	Clock Divider Factor	0x1
30	reg_div_clk_osc_saradc_dis	R/W	Clock gate	0x0
31	reg_div_clk_osc_saradc_hwide	R/W	Select High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider	0x0

fc_coarse_en

Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	fc_coarse_en	R/W	Enable 32K course tuning: 0 = disable, 1 = enable	0x0
31:1	Reserved			

fc_coarse_cal

Offset Address: 0x044

Bits	Name	Access	Description	Reset
15:0	fc_coarse_value	RO	32K coarse counter value (unit: 25MHz clock) 25MHz clock counts one 32K clock cycle.	
31:16	fc_coarse_time	RO	32K coarse tune completion count.	

fc_fine_en

Offset Address: 0x048

Bits	Name	Access	Description	Reset
0	fc_fine_en	R/W	Enable 32K fine tuning: 0 = disable, 1 = enable	0x0
31:1	Reserved			

fc_fine_period

Offset Address: 0x04c

Bits	Name	Access	Description	Reset
15:0	fc_fine_period	R/W	32K fine adjustment counting period (unit: 32K clock) Set how many 32K clock periods are counted each time using the 25MHz clock.	0x0100
31:16	Reserved			

fc_fine_cal

Offset Address: 0x050

Bits	Name	Access	Description	Reset
23:0	fc_fine_value	RO	32K fine adjustment counter value (unit: 25MHz clock) One 25MHz clock counts one fc_fine_period cycle.	
31:24	fc_fine_time	RO	32K fine tune completion count.	

rtcsys_pmu2

Offset Address: 0x054

Bits	Name	Access	Description	Reset
0	reg_rtc_sys_wkint_db_en	R/W	PMU wakeup int debounce enable	0x1
4:1	reg_rtc_sys_wkint_db_cnt	R/W	PMU wakeup int debounce cycle (32K)	0x2
31:5	Reserved			

rtcsys_clkdiv1

Offset Address: 0x058

Bits	Name	Access	Description	Reset
15:0	Reserved			
21:16	reg_div_clk_osc_irrx_div_val	R/W	Clock Divider Factor	0x0
22	Reserved			
23	reg_div_clk_osc_irrx_dis	R/W	Clock gate	0x1
31:24	Reserved			

rtcsys_mcu51_dbg

Offset Address: 0x05c

Bits	Name	Access	Description	Reset
3:0	reg_51_dbg_sel	R/W	select mcu51 debug bus (check mcu design review ppt)	0x0
4	reg_51_dbg_snap_shot	W1P	snap shot mcu51 internal register to dbg register (reg_rtcsys_dbg)	
5	reg_51_dbg_step_en	R/W	0: disable mcu debug function 1: enable mcu debug function, and mcu stop at current PC	0x0
6	reg_51_dbg_step	W1P	1: mcu jump to next PC	
7	reg_51_dbg_jump	W1P	1: mcu jump to target pc value (reg_51_dbg_jump2pc)	
15:8	Reserved			
31:16	reg_51_dbg_jump2pc	R/W	16 bit mcu target pc value	0x0

sw_reg0

Offset Address: 0x060

Bits	Name	Access	Description	Reset
7:0	sw_reg0	R/W	reg for SW	0x0
31:8	Reserved			

sw_reg1_por

G

Offset Address: 0x064

Bits	Name	Access	Description	Reset
7:0	sw_reg1_por	R/W	reg for SW could only be reset by power reset	0x0
31:8	Reserved			

fab_lp_ctrl

Offset Address: 0x068

Bits	Name	Access	Description	Reset
7:0	rtcsys_fab_busy_sel	R/W	select signal to request sys_ctrl to speed up fab clock	0xDF
9:8	rtcsys_fab_busy_ctrl	R/W	rtcsys_fab_busy signal is combi or register out	0x0
11:10	apdbg_busy_ctrl	R/W	apdbg_busy signal is combi or register out	0x0
13:12	reg_apb_busy_ctrl	R/W	apb bridge_busy signal is combi or register out	0x3
15:14	reg_mcu_busy_ctrl	R/W	mcu_busy signal is combi or register out	0x3
31:16	Reserved			

fab_option

Offset Address: 0x06c

Bits	Name	Access	Description	Reset
1:0	rtcsys_fab_option	R/W	the ahb_h2h2 design option	0x0
31:2	Reserved			

rtcsys_mcu51_ictrl1

Offset Address: 0x07c

Bits	Name	Access	Description	Reset
15:0	reg_51_int1_src_mask	R/W	select rtcsys_int src to mcu int1_n 1: mask, 0: un-mask [0]: vbat_det [1]: mbox0_int [2]: NA [3]: irrx_int [4]: gpio_int [5]: uart_int [6]: spinor1_int [7]: timer0_int [8]: timer1_int [9]: irq_ap2rtc[0] [10]: irq_ap2rtc[1]	0xffff

Bits	Name	Access	Description	Reset
			[11]: i2c_int [12]: rtc_state_change_int [13]: hw_thm_shdn [14]: saradc [15]: wdt_int	
31:16	reg_51_int1_final_status	RO	mcu int1_n status [0]: vbat_det [1]: mbox0_int [2]: NA [3]: irrx_int [4]: gpio_int [5]: uart_int [6]: spinor1_int [7]: timer0_int [8]: timer1_int [9]: irq_ap2rtc[0] [10]: irq_ap2rtc[1] [11]: i2c_int [12]: rtc_state_change_int [13]: hw_thm_shdn [14]: saradc [15]: wdt_int	

rtc_ip_pwr_req

Offset Address: 0x080

Bits	Name	Access	Description	Reset
0	reg_sd1_pwr_req	R/W	power fence control 1: power on, 0: power off [0]: sd1	0x1
1	reg_sd1_pwr_req_2nd	R/W	power fence control 1: power on, 0: power off [0]: sd1	0x1
2	reg_mcu_pwr_req	R/W	power fence control 1: power on, 0: power off [1]: mcu subsys	0x1
3	reg_mcu_pwr_req_2nd	R/W	power fence control 1: power on, 0: power off [1]: mcu subsys	0x1
15:4	Reserved			
16	reg_sd1_pwr_ack	RO	power fence power status 1: power on, 0: power off [0]: sd1	
17	reg_sd1_pwr_ack_2nd	RO	power fence power status 1: power on, 0: power off [0]: sd1	
18	reg_mcu_pwr_ack	RO	power fence power status 1: power on, 0: power off [1]: mcu subsys	
19	reg_mcu_pwr_ack_2nd	RO	power fence power status 1: power on, 0: power off [1]: mcu subsys	
31:20	Reserved			

rtc_ip_iso_ctrl

Offset Address: 0x084

Bits	Name	Access	Description	Reset
0	reg_sd1_iso_en	R/W	sd1 iso enablel 1: iso enable, 0: iso disable	0x0
1	reg_mcu_iso_en	R/W	mcu iso enablel 1: iso enable, 0: iso disable	0x0
15:2	Reserved			
17:16	reg_ip_por_en	R/W	1: pwr_island reset assert when power ack is 0	0x3
31:18	Reserved			

rtcsys_wkup_ctrl1

Offset Address: 0x094

Bits	Name	Access	Description	Reset
7:0	reg_rtcsys_wkint_final_status	RO	wkint final status [0]: sd1_wakeup_intr [1]: gpio_int [2]: timer0_int [3]: timer1_int [4]: saradc_int [5]: rtcsys_ictrl_int [6]: NA [7]: NA	
31:8	Reserved			

rtcsys_sram_ctrl

Offset Address: 0x098

Bits	Name	Access	Description	Reset
0	reg_ahb_sram_slp	R/W	1 : ahb sram into sleep mode	0x0
1	reg_ahb_sram_sd	R/W	1 : ahb sram into shut down mode	0x0
2	reg_ahb_sram_ctrl_ov	R/W	0 : ahb sram ctrl by PMU FSM and ahb sram busy 1: sram ctrol by register reg_ahb_sram_slp/reg_ahb_sram_sd	0x1
3	reg_sdio_sram_slp	R/W	1 : sdio sram into sleep mode	0x0
4	reg_sdio_sram_sd	R/W	1 : sdio sram into shut down mode	0x0
5	reg_sdio_sram_ctrl_ov	R/W	0 : sram's sd pin = 1'b0 1: sram ctrol by register reg_sdio_sram_sd	0x1
6	reg_mcu_sram_slp	R/W	1 : mcu iram sram into sleep mode	0x0
7	reg_mcu_sram_sd	R/W	1 : mcu iram sram into shut down mode	0x0
8	reg_mcu_sram_ctrl_ov	R/W	0 : mcu iram sram ctrl by PMU FSM 1: sram ctrol by register reg_ahb_sram_slp/reg_ahb_sram_sd	0x1
9	reg_rtc_sram_slp	R/W	1 : mcu iram sram into sleep mode	0x0
10	reg_rtc_sram_sd	R/W	1 : mcu iram sram into shut down mode	0x0
11	reg_rtc_sram_ctrl_ov	R/W	0 : mcu iram sram ctrl by PMU FSM 1: sram ctrol by register reg_ahb_sram_slp/reg_ahb_sram_sd	0x1
12	reg_ahb_sram_1_slp	R/W	1 : ahb sram into sleep mode	0x0
13	reg_ahb_sram_1_sd	R/W	1 : ahb sram into shut down mode	0x0
14	reg_ahb_sram_1_ctrl_ov	R/W	0 : ahb sram ctrl by PMU FSM and ahb sram busy 1: sram ctrol by register	0x1

Bits	Name	Access	Description	Reset
			reg_ahb_sram_slp/reg_ahb_sram_sd	
15	reg_ahb_sram_2_slp	R/W	1 : ahb sram into sleep mode	0x0
16	reg_ahb_sram_2_sd	R/W	1 : ahb sram into shut down mode	0x0
17	reg_ahb_sram_2_ctrl_ov	R/W	0 : ahb sram ctrl by PMU FSM and ahb sram busy 1: sram ctrol by register reg_ahb_sram_slp/reg_ahb_sram_sd	0x1
27:18	Reserved			
28	reg_mcu_sram_force_ce	R/W	1: force mcu_iram cs = 1	0x1
31:29	Reserved			

rtcsys_io_ctrl

Offset Address: 0x09c

Bits	Name	Access	Description	Reset
0	reg_i2c_mux_opt0	R/W	0: pwr_gpio6/8 control by dw_gpio 1: pwr_gpio6 is PWR_IIC_SDA pwr_gpio8 is PWR_IIC_SCL	0x0
31:1	Reserved			

rtcsys_wdt_ctrl

Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
0	reg_rtc_hw_wdg_rst_en	R/W	0: disable rtc wdt trigger warm reset or pwrcyc reset 1: enable rtc wdt trigger warm reset or pwrcyc reset	0x0
1	reg_rtc_wdt_ctrl_mask_en	R/W	no load	0x1
31:2	Reserved			

rtcsys_irrx_clk_ctrl

Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
0	reg_irrx_clk_sw_force_on	R/W	force on clk ctrl of irrx	0x1
1	reg_irrx_xtal_req_en	R/W	enable irrx clk ctrl requet XTAL	0x0
2	reg_irrx_osc_req_en	R/W	enable irrx clk ctrl requet OSC	0x0
3	reg_irrx_ldo_req_en	R/W	enable irrx clk ctrl requet LDO	0x0
7:4	Reserved			
15:8	reg_irrx_xtal_filter_cyc	R/W	irrx xtal filter cycle (default 2ms)	0x40
19:16	reg_irrx_clk_ctrl_st	RO	irrx clock ctrol state	
31:20	Reserved			

rtcsys_rtc_wkup_ctrl

Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
7:0	reg_rtc_wkint_mask	R/W	wakeup source mask int to RTC_CORE [0]: irrrx_intr [1]: gpio_int [2]: timer0_int [3]: timer1_int [4]: saradc_int	0xff

Bits	Name	Access	Description	Reset
			[5]: rtcsys_ictrl_int [6]: wdt_int [7]: irrx_wakeup	
15:8	Reserved			
23:16	reg_rtc_puint_mask	R/W	power-up source mask int to RTC_CORE [0]: irrrx_intr [1]: gpio_int [2]: timer0_int [3]: timer1_int [4]: saradc_int [5]: rtcsys_ictrl_int [6]: wdt_int [7]: irrx_wakeup	0xff
31:24	Reserved			

rtcsys_por_rst_ctrl

Offset Address: 0x0ac

Bits	Name	Access	Description	Reset
0	reg_rtcsys_reset_en	R/W	0: not allow rtcsys reset by pwr cyc/ wdt warm reset 1 : allow rtcsys reset by pwr cyc/ wdt warm reset	0x0
1	reg_rtcsys_rstn_src_sel	R/W	select rtcsys rstn src 0: rtc_core fsm (reset with die domain) 1: por_pwr_rstn	0x0
31:2	Reserved			

4 Memory Interface

4.1 DDR Controller

4.1.1 Overview

DDR controller realizes the data access of dynamic random access memory (DRAM). It converts the data access command of each main device in SoC into the DRAM command conforming to JEDEC standard and schedules it properly, so as to improve the efficiency of dynamic memory.

4.1.2 Characteristics

- Features:
- Supports:
  - DDR2 with maximum data rate of 1333 Mbps.
  - DDR3 with maximum data rate of 1866 Mbps.
- Supports interface data width of 16-bit.
- Supports single channel, single rank.
- Supports automatic refresh control.
- Supports priority control.
- Supports data flow statistics.
- Supports low power mode.
- Supports address mapping.
- Supports pin multiplexing.

4.1.3 Function Description

4.1.3.1 Application Block Diagram

The DRAM interface supports a 16-bit data width. Figure 4-1 shows a schematic diagram of the interconnection between the main chip and a single DRAM device.

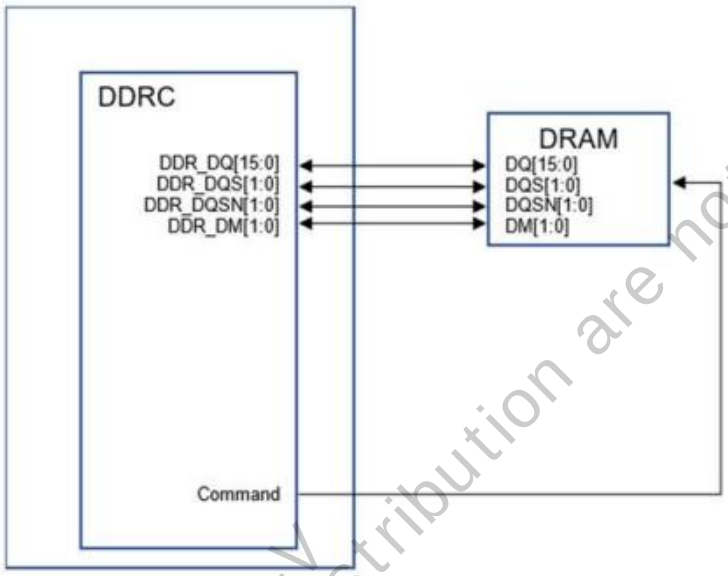


Figure 4- 1 SoC/DRAM Interconnection Diagram

Command consists of several signals, which vary according to DRAM type. Table 4-1 compares the DDR2 / DDR3 command signals.

Table 4-1 DDR2/DDR3 Command Difference Comparison

Function	DDR2	DDR3
DDR CKE	D2x_CKE	D3x_CKE
DDR CLKN	D2x_CK_N	D3x_CK_N
DDR CLKP	D2x_CK_P	D3x_CK_P
DDR CSB	D2x_CS_N	D3x_CS_N
DDR RESETN	D2x_RESET_N	D3x_RESET_N
DDR RASN	D2x_RAS_N	D3x_RAS_N
DDR CASN	D2x_CAS_N	D3x_CAS_N
DDR WEN	D2x_WE_N	D3x_WE_N
DDR ACTN	N/A	N/A
DDR BA	D2x_BAn	D3x_BAn
DDR BG	N/A	N/A
DDR MA	D2x_An (n = 0 – 15)	D3x_An (n = 0 – 15)
DDR ODT	D2x_ODT	D3x_ODT
Explanation: The pin names may have slight differences depending on the package.		

4.1.3.2 Function Principle

Based on the storage characteristics of DRAM, JEDEC formulates a set of standards, which regulate the command and sequence needed to access DRAM data and control DRAM status. With proper configuration of DDR register, DDR controller can send out the commands that meets JEDEC standard and complete the actions of reading, writing and power consumption control.

4.1.3.2.1 Command Truth Table

DDR interface meets JDEDC standard, which is shown in Table 4-1 and Table 4-2. They are the support commands truth tables of DDR2, DDR3 respectively for users' reference. Other information can refer to JEDEC standard.

Table 4- 1 DDR2 Command Truth Table

Function	CKE Pre	CKE Cur	CS#	RAS#	CAS#	WE#	BA0 - BA2	A11 - A15	A10 / AP	A0 - A9
Mode Register Set	H	H	L	L	L	L	BA	OP		
Refresh	H	H	L	L	L	H	V	V	V	V
Self Refresh Entry	H	L	L	L	L	H	V	V	V	V
Self Refresh Exit	L	H	H	X	X	X	X	X	X	X
			L	H	H	H	V	V	V	V
Single Bank Precharge	H	H	L	L	H	L	BA	V	L	V
Precharge all Banks	H	H	L	L	H	L	V	V	H	V
Bank Activate	H	H	L	L	H	H	BA	RA		
Write	H	H	L	H	L	L	BA	RFU	L	CA
Write with Auto Precharge	H	H	L	H	L	L	BA	RFU	H	CA
Read	H	H	L	H	L	H	BA	RFU	L	CA
Read with Auto Precharge	H	H	L	H	L	H	BA	RFU	H	CA
Read with Auto Precharge	H	H	L	H	L	H	BA	RFU	H	CA
No Operation	H	H	L	H	H	H	V	V	V	V
Device Deselected	H	H	H	X	X	X	X	X	X	X
Power Down Entry	H	L	L	H	H	H	V	V	V	V
			H	X	X	X	X	X	X	X
Power Down Exit	L	H	L	H	H	H	V	V	V	V
			H	X	X	X	X	X	X	X

H：High level；L：Low level；V：Effective；X：Does not matter
RFU：reserve for future using.

Table 4- 2 DDR3 Command Truth Table

Function	CKE Pre	CKE Cur	CS #	RAS#	CAS#	WE#	BA0 - BA2	A13 - A15	A12 / BC #	A10 / AP	A0 - A9, 11
Mode Register Set	H	H	L	L	L	L	BA	OP			
Refresh	H	H	L	L	L	H	V	V	V	V	V
Self Refresh Entry	H	L	L	L	L	H	V	V	V	V	V
Self Refresh Exit	L	H	H	X	X	X	X	X	X	X	X
			L	H	H	H	V	V	V	V	V
Single Bank Precharge	H	H	L	L	H	L	BA	V	V	L	V
Precharge all Banks	H	H	L	L	H	L	V	V	V	H	V
Bank Activate	H	H	L	L	H	H	BA	RA			
Write (Fixed BL8 or BC4)	H	H	L	H	L	L	BA	RFU	V	L	CA
Write (BC4, on the Fly)	H	H	L	H	L	L	BA	RFU	L	L	CA
Write (BL8, on the Fly)	H	H	L	H	L	L	BA	RFU	H	L	CA
Write with Auto Precharge (Fixed BL8 or BC4)	H	H	L	H	L	L	BA	RFU	V	H	CA
Write with Auto Precharge (BC4, on the Fly)	H	H	L	H	L	L	BA	RFU	L	H	CA
Write with Auto Precharge (BL8, on the Fly)	H	H	L	H	L	L	BA	RFU	H	H	CA
Read (Fixed BL8 or BC4)	H	H	L	H	L	H	BA	RFU	V	L	CA
Read (BC4, on the Fly)	H	H	L	H	L	H	BA	RFU	L	L	CA
Read (BL8, on the Fly)	H	H	L	H	L	H	BA	RFU	H	L	CA
Read with Auto Precharge (Fixed BL8 or BC4)	H	H	L	H	L	H	BA	RFU	V	H	CA
Read with Auto Precharge (BC4, on the Fly)	H	H	L	H	L	H	BA	RFU	L	H	CA
Read with Auto Precharge (BL8, on the Fly)	H	H	L	H	L	H	BA	RFU	H	H	CA
No Operation	H	H	L	H	H	H	V	V	V	V	V
Device Deselected	H	H	H	X	X	X	X	X	X	X	X
Power Down Entry	H	L	L	H	H	H	V	V	V	V	V
			H	X	X	X	X	X	X	X	X
Power Down Exit	L	H	L	H	H	H	V	V	V	V	V
			H	X	X	X	X	X	X	X	X
ZQ Calibration Long	H	H	L	H	H	L	X	X	X	H	X
ZQ Calibration Short	H	H	L	H	H	L	X	X	X	L	X

H：High level；L：Low level；V：Effective；X：Does not matter
RFU：reserve for future using.

4.1.3.2.2 Automatic Refresh

The DDR controller has the ability to refresh the DRAM content automatically. The purpose of controlling the automatic refresh is to reduce the delay of accessing data or the impact of refresh command on the DRAM bandwidth by trying to send refresh command when the DRAM is idle. The specific available means are as follows：

Equal Interval Refresh: issue refresh command ever tREFI time.

Smart Refresh: The internal DDR controller will count the number of expired tREFI, and then use the idle time to send data continuously.

4.1.3.2.3 Low Power Consumption Management

DDR controller supports low power consumption mode:

Normal Low Power Consumption Mode: Set an idle timer through register. When the normal low power consumption mode is enabled and the DDR controller was idle for a shorter period, the DRAM will be automatically put to the normal low power consumption mode until there is any access.

Self Refresh Mode: It is a mode that consumes much lower power. When the self refresh mode is enabled and the DDR controller was idle for a longer period, the DRAM will be automatically put to the self refresh mode until there is any access.

4.1.3.2.4 Arbitration mechanism

DDR controller optimizes the bandwidth utilization of the system based on the control timing of DRAM, and schedules the commands through priority scheduling algorithm. In addition, DDRC also implements two scheduling auxiliary means and real-time control (enabling these two control means according to business needs, which can be enabled at the same time or separately), to control command requests

Continuous Address Access Restrictions:

The limitation is 0 ~ 15 DRAM read / write instructions, and the configuration of each AXI port is independent. DDR controller has high priority for continuous address by default to optimize DRAM utilization. This mechanism limits the maximum length of continuous access DRAM for each AXI port.

Timeout Control

For each read/write transfer on the AXI port, a timeout register can be configured to avoid waiting for an excessively long time. Once the waiting time is reached, the AXI

port that has not yet reached the waiting time or has not been configured with timeout properties will be forcibly masked.

Priority Scheduling:

The priority level is 0-15. The higher the value is, the higher the priority is. The read / write priority configuration of each AXI port is independent.

Real-time Control:

For real-time function, the hardware buffer threshold can be configured. If the buffer is insufficient, the priority will be raised to the highest automatically, and other AXI ports can be restricted to generate new transmissions

4.1.3.2.5 Flow Statistics and Command Latency Statistics Function

DDR controller supports traffic statistics function: it can count the read and write traffic of each AXI port to collect the current traffic information and decide whether to control the traffic. It can be used to count the total read / write traffic of DRAM.
DDR controller supports AXI latency statistics function, which supports cumulative latency statistics for specified/unscheduled transfers.

4.1.3.2.6 Address Mapping Method

DDR controller converts the access address of system to that of DRAM. It realizes RBC (row_bank_column) , BRC and bank interleave in row / column bit.

4.1.4 Working Method

4.1.4.1 Soft Reset

Soft reset is not supported.

4.1.4.2 DDR Initialization Configuration Process

The initialization process of the controller is provided in the form of software package.

4.1.5 AXI Register

4.1.5.1 AXI RegisterOverview

Base Address: 0x0800_4000

Name	Address Offset	Description
AXI_CTRL0_1	0x4b4	AXI1 read timeout control
AXI_CTRL1_1	0x4b8	AXI1 write timeout control
AXI_CTRL0_2	0x564	AXI2 read timeout control
AXI_CTRL1_2	0x568	AXI2 write timeout control
AXI_CTRL0_3	0x614	AXI3 read timeout control
AXI_CTRL1_3	0x618	AXI3 write timeout control

Base Address: 0x0800_8000

Name	Address Offset	Description
AXI_MON0_CTRL	0x000	AXI monitor 0 control
AXI_MON0_INPUT	0x004	AXI monitor 0 input selection
AXI_MON0_FILTER0	0x010	AXI monitor 0 filter settings
AXI_MON0_FILTER1	0x014	AXI monitor 0 filter settings
AXI_MON0_FILTER2	0x018	AXI monitor 0 filter settings
AXI_MON0_FILTER3	0x01c	AXI monitor 0 filter settings
AXI_MON0_FILTER4	0x020	AXI monitor 0 filter settings
AXI_MON0_FILTER5	0x024	AXI monitor 0 filter settings
AXI_MON0_FILTER6	0x028	AXI monitor 0 filter settings
AXI_MON0_FILTER7	0x02c	AXI monitor 0 filter settings
AXI_MON0_FILTER8	0x030	AXI monitor 0 filter settings
AXI_MON0_RPT0	0x040	AXI monitor 0 cycle count
AXI_MON0_RPT1	0x044	AXI monitor 0 hit count
AXI_MON0_RPT2	0x048	AXI monitor 0 byte count
AXI_MON0_RPT3	0x04c	AXI monitor 0 latency count
AXI_MON1_CTRL	0x080	AXI monitor 1 control
AXI_MON1_INPUT	0x084	AXI monitor 1 input selection
AXI_MON1_FILTER0	0x090	AXI monitor 1 filter settings
AXI_MON1_FILTER1	0x094	AXI monitor 1 filter settings
AXI_MON1_FILTER2	0x098	AXI monitor 1 filter settings
AXI_MON1_FILTER3	0x09c	AXI monitor 1 filter settings
AXI_MON1_FILTER4	0x0a0	AXI monitor 1 filter settings
AXI_MON1_FILTER5	0x0a4	AXI monitor 1 filter settings
AXI_MON1_FILTER6	0x0a8	AXI monitor 1 filter settings
AXI_MON1_FILTER7	0x0ac	AXI monitor 1 filter settings
AXI_MON1_FILTER8	0x0b0	AXI monitor 1 filter settings
AXI_MON1_RPT0	0x0c0	AXI monitor 1 cycle count
AXI_MON1_RPT1	0x0c4	AXI monitor 1 hit count
AXI_MON1_RPT2	0x0c8	AXI monitor 1 byte count
AXI_MON1_RPT3	0x0cc	AXI monitor 1 latency count
AXI_MON2_CTRL	0x100	AXI monitor 2 control
AXI_MON2_INPUT	0x104	AXI monitor 2 input selection
AXI_MON2_FILTER0	0x110	AXI monitor 2 filter settings

Name	Address Offset	Description
AXI_MON2_FILTER1	0x114	AXI monitor 2 filter settings
AXI_MON2_FILTER2	0x118	AXI monitor 2 filter settings
AXI_MON2_FILTER3	0x11c	AXI monitor 2 filter settings
AXI_MON2_FILTER4	0x120	AXI monitor 2 filter settings
AXI_MON2_FILTER5	0x124	AXI monitor 2 filter settings
AXI_MON2_FILTER6	0x128	AXI monitor 2 filter settings
AXI_MON2_FILTER7	0x12c	AXI monitor 2 filter settings
AXI_MON2_FILTER8	0x130	AXI monitor 2 filter settings
AXI_MON2_RPT0	0x140	AXI monitor 2 cycle count
AXI_MON2_RPT1	0x144	AXI monitor 2 hit count
AXI_MON2_RPT2	0x148	AXI monitor 2 byte count
AXI_MON2_RPT3	0x14c	AXI monitor 2 latency count
AXI_MON3_CTRL	0x180	AXI monitor 3 control
AXI_MON3_INPUT	0x184	AXI monitor 3 input selection
AXI_MON3_FILTER0	0x190	AXI monitor 3 filter settings
AXI_MON3_FILTER1	0x194	AXI monitor 3 filter settings
AXI_MON3_FILTER2	0x198	AXI monitor 3 filter settings
AXI_MON3_FILTER3	0x19c	AXI monitor 3 filter settings
AXI_MON3_FILTER4	0x1a0	AXI monitor 3 filter settings
AXI_MON3_FILTER5	0x1a4	AXI monitor 3 filter settings
AXI_MON3_FILTER6	0x1a8	AXI monitor 3 filter settings
AXI_MON3_FILTER7	0x1ac	AXI monitor 3 filter settings
AXI_MON3_FILTER8	0x1b0	AXI monitor 3 filter settings
AXI_MON3_RPT0	0x1c0	AXI monitor 3 cycle count
AXI_MON3_RPT1	0x1c4	AXI monitor 3 hit count
AXI_MON3_RPT2	0x1c8	AXI monitor 3 byte count
AXI_MON3_RPT3	0x1cc	AXI monitor 3 latency count
AXI_MON4_CTRL	0x200	AXI monitor 4 control
AXI_MON4_INPUT	0x204	AXI monitor 4 input selection
AXI_MON4_FILTER0	0x210	AXI monitor 4 filter settings
AXI_MON4_FILTER1	0x214	AXI monitor 4 filter settings
AXI_MON4_FILTER2	0x218	AXI monitor 4 filter settings
AXI_MON4_FILTER3	0x21c	AXI monitor 4 filter settings
AXI_MON4_FILTER4	0x220	AXI monitor 4 filter settings
AXI_MON4_FILTER5	0x224	AXI monitor 4 filter settings
AXI_MON4_FILTER6	0x228	AXI monitor 4 filter settings
AXI_MON4_FILTER7	0x22c	AXI monitor 4 filter settings
AXI_MON4_FILTER8	0x230	AXI monitor 4 filter settings
AXI_MON4_RPT0	0x240	AXI monitor 4 cycle count
AXI_MON4_RPT1	0x244	AXI monitor 4 hit count
AXI_MON4_RPT2	0x248	AXI monitor 4 byte count
AXI_MON4_RPT3	0x24c	AXI monitor 4 latency count
AXI_MON5_CTRL	0x280	AXI monitor 5 control
AXI_MON5_INPUT	0x284	AXI monitor 5 input selection
AXI_MON5_FILTER0	0x290	AXI monitor 5 filter settings
AXI_MON5_FILTER1	0x294	AXI monitor 5 filter settings
AXI_MON5_FILTER2	0x298	AXI monitor 5 filter settings
AXI_MON5_FILTER3	0x29c	AXI monitor 5 filter settings
AXI_MON5_FILTER4	0x2a0	AXI monitor 5 filter settings
AXI_MON5_FILTER5	0x2a4	AXI monitor 5 filter settings
AXI_MON5_FILTER6	0x2a8	AXI monitor 5 filter settings
AXI_MON5_FILTER7	0x2ac	AXI monitor 5 filter settings
AXI_MON5_FILTER8	0x2b0	AXI monitor 5 filter settings
AXI_MON5_RPT0	0x2c0	AXI monitor 5 cycle count

Name	Address Offset	Description
AXI_MON5_RPT1	0x2c4	AXI monitor 5 hit count
AXI_MON5_RPT2	0x2c8	AXI monitor 5 byte count
AXI_MON5_RPT3	0x2cc	AXI monitor 5 latency count
AXI_MON6_CTRL	0x300	AXI monitor 6 control
AXI_MON6_INPUT	0x304	AXI monitor 6 input selection
AXI_MON6_FILTER0	0x310	AXI monitor 6 filter settings
AXI_MON6_FILTER1	0x314	AXI monitor 6 filter settings
AXI_MON6_FILTER2	0x318	AXI monitor 6 filter settings
AXI_MON6_FILTER3	0x31c	AXI monitor 6 filter settings
AXI_MON6_FILTER4	0x320	AXI monitor 6 filter settings
AXI_MON6_FILTER5	0x324	AXI monitor 6 filter settings
AXI_MON6_FILTER6	0x328	AXI monitor 6 filter settings
AXI_MON6_FILTER7	0x32c	AXI monitor 6 filter settings
AXI_MON6_FILTER8	0x330	AXI monitor 6 filter settings
AXI_MON6_RPT0	0x340	AXI monitor 6 cycle count
AXI_MON6_RPT1	0x344	AXI monitor 6 hit count
AXI_MON6_RPT2	0x348	AXI monitor 6 byte count
AXI_MON6_RPT3	0x34c	AXI monitor 6 latency count
AXI_MON7_CTRL	0x380	AXI monitor 7 control
AXI_MON7_INPUT	0x384	AXI monitor 7 input selection
AXI_MON7_FILTER0	0x390	AXI monitor 7 filter settings
AXI_MON7_FILTER1	0x394	AXI monitor 7 filter settings
AXI_MON7_FILTER2	0x398	AXI monitor 7 filter settings
AXI_MON7_FILTER3	0x39c	AXI monitor 7 filter settings
AXI_MON7_FILTER4	0x3a0	AXI monitor 7 filter settings
AXI_MON7_FILTER5	0x3a4	AXI monitor 7 filter settings
AXI_MON7_FILTER6	0x3a8	AXI monitor 7 filter settings
AXI_MON7_FILTER7	0x3ac	AXI monitor 7 filter settings
AXI_MON7_FILTER8	0x3b0	AXI monitor 7 filter settings
AXI_MON7_RPT0	0x3c0	AXI monitor 7 cycle count
AXI_MON7_RPT1	0x3c4	AXI monitor 7 hit count
AXI_MON7_RPT2	0x3c8	AXI monitor 7 byte count
AXI_MON7_RPT3	0x3cc	AXI monitor 7 latency count
AXI_MON8_CTRL	0x400	AXI monitor 8 control
AXI_MON8_INPUT	0x404	AXI monitor 8 input selection
AXI_MON8_FILTER0	0x410	AXI monitor 8 filter settings
AXI_MON8_FILTER1	0x414	AXI monitor 8 filter settings
AXI_MON8_FILTER2	0x418	AXI monitor 8 filter settings
AXI_MON8_FILTER3	0x41c	AXI monitor 8 filter settings
AXI_MON8_FILTER4	0x420	AXI monitor 8 filter settings
AXI_MON8_FILTER5	0x424	AXI monitor 8 filter settings
AXI_MON8_FILTER6	0x428	AXI monitor 8 filter settings
AXI_MON8_FILTER7	0x42c	AXI monitor 8 filter settings
AXI_MON8_FILTER8	0x430	AXI monitor 8 filter settings
AXI_MON8_RPT0	0x440	AXI monitor 8 cycle count
AXI_MON8_RPT1	0x444	AXI monitor 8 hit count
AXI_MON8_RPT2	0x448	AXI monitor 8 byte count
AXI_MON8_RPT3	0x44c	AXI monitor 8 latency count
AXI_MON9_CTRL	0x480	AXI monitor 9 control
AXI_MON9_INPUT	0x484	AXI monitor 9 input selection
AXI_MON9_FILTER0	0x490	AXI monitor 9 filter settings
AXI_MON9_FILTER1	0x494	AXI monitor 9 filter settings
AXI_MON9_FILTER2	0x498	AXI monitor 9 filter settings
AXI_MON9_FILTER3	0x49c	AXI monitor 9 filter settings

Name	Address Offset	Description
AXI_MON9_FILTER4	0x4a0	AXI monitor 9 filter settings
AXI_MON9_FILTER5	0x4a4	AXI monitor 9 filter settings
AXI_MON9_FILTER6	0x4a8	AXI monitor 9 filter settings
AXI_MON9_FILTER7	0x4ac	AXI monitor 9 filter settings
AXI_MON9_FILTER8	0x4b0	AXI monitor 9 filter settings
AXI_MON9_RPT0	0x4c0	AXI monitor 9 cycle count
AXI_MON9_RPT1	0x4c4	AXI monitor 9 hit count
AXI_MON9_RPT2	0x4c8	AXI monitor 9 byte count
AXI_MON9_RPT3	0x4cc	AXI monitor 9 latency count
AXI_MON10_CTRL	0x500	AXI monitor 10 control
AXI_MON10_INPUT	0x504	AXI monitor 10 input selection
AXI_MON10_FILTER0	0x510	AXI monitor 10 filter settings
AXI_MON10_FILTER1	0x514	AXI monitor 10 filter settings
AXI_MON10_FILTER2	0x518	AXI monitor 10 filter settings
AXI_MON10_FILTER3	0x51c	AXI monitor 10 filter settings
AXI_MON10_FILTER4	0x520	AXI monitor 10 filter settings
AXI_MON10_FILTER5	0x524	AXI monitor 10 filter settings
AXI_MON10_FILTER6	0x528	AXI monitor 10 filter settings
AXI_MON10_FILTER7	0x52c	AXI monitor 10 filter settings
AXI_MON10_FILTER8	0x530	AXI monitor 10 filter settings
AXI_MON10_RPT0	0x540	AXI monitor 10 cycle count
AXI_MON10_RPT1	0x544	AXI monitor 10 hit count
AXI_MON10_RPT2	0x548	AXI monitor 10 byte count
AXI_MON10_RPT3	0x54c	AXI monitor 10 latency count
AXI_MON11_CTRL	0x580	AXI monitor 11 control
AXI_MON11_INPUT	0x584	AXI monitor 11 input selection
AXI_MON11_FILTER0	0x590	AXI monitor 11 filter settings
AXI_MON11_FILTER1	0x594	AXI monitor 11 filter settings
AXI_MON11_FILTER2	0x598	AXI monitor 11 filter settings
AXI_MON11_FILTER3	0x59c	AXI monitor 11 filter settings
AXI_MON11_FILTER4	0x5a0	AXI monitor 11 filter settings
AXI_MON11_FILTER5	0x5a4	AXI monitor 11 filter settings
AXI_MON11_FILTER6	0x5a8	AXI monitor 11 filter settings
AXI_MON11_FILTER7	0x5ac	AXI monitor 11 filter settings
AXI_MON11_FILTER8	0x5b0	AXI monitor 11 filter settings
AXI_MON11_RPT0	0x5c0	AXI monitor 11 cycle count
AXI_MON11_RPT1	0x5c4	AXI monitor 11 hit count
AXI_MON11_RPT2	0x5c8	AXI monitor 11 byte count
AXI_MON11_RPT3	0x5cc	AXI monitor 11 latency count

4.1.5.2 AXI Register Description

Base Address: 0x0800_4000

AXI_CTRL0_1

Offset Address: 0x4b4

Bits	Name	Access	Description	Reset
9:0	axi1_rd_timeout_val	R/W	After an AXI read transaction is granted, a timeout counter starts to count. When it counts to axi<n>_rd_timeout_val, the corresponding channel has the highest priority.	0x0
11:10	Reserved			
12	axi1_rd_timeout_en	R/W	If set to 1, enables the timeout function for the read channel of port n.	0x0
31:13	Reserved			

AXI_CTRL1_1

Offset Address: 0x4b8

Bits	Name	Access	Description	Reset
9:0	axi1_wr_timeout_val	R/W	After an AXI write transaction is granted, a timeout counter starts to count. When it counts to axi<n>_wr_timeout_val, the corresponding channel has the highest priority.	0x0
11:10	Reserved			
12	axi1_wr_timeout_en	R/W	If set to 1, enables the timeout function for the write channel of port n.	0x0
31:13	Reserved			

AXI_CTRL0_2

Offset Address: 0x564

Bits	Name	Access	Description	Reset
9:0	axi2_rd_timeout_val	R/W	After an AXI read transaction is granted, a timeout counter starts to count. When it counts to axi<n>_rd_timeout_val, the corresponding channel has the highest priority.	0x0
11:10	Reserved			
12	axi2_rd_timeout_en	R/W	If set to 1, enables the timeout function for the read channel of port n.	0x0
31:13	Reserved			

AXI_CTRL1_2

Offset Address: 0x568

Bits	Name	Access	Description	Reset
9:0	axi2_wr_timeout_val	R/W	After an AXI write transaction is granted, a timeout counter starts to count. When it counts to axi<n>_wr_timeout_val, the corresponding channel has the highest priority.	0x0
11:10	Reserved			
12	axi2_wr_timeout_en	R/W	If set to 1, enables the timeout function for the write channel of port n.	0x0
31:13	Reserved			

AXI_CTRL0_3

Offset Address: 0x614

Bits	Name	Access	Description	Reset
9:0	axi3_rd_timeout_val	R/W	After an AXI read transaction is granted, a timeout counter starts to count. When it counts to axi<n>_rd_timeout_val, the corresponding channel has the highest priority.	0x0
11:10	Reserved			
12	axi3_rd_timeout_en	R/W	If set to 1, enables the timeout function for the read channel of port n.	0x0
31:13	Reserved			

AXI_CTRL1_3

Offset Address: 0x618

Bits	Name	Access	Description	Reset
9:0	axi3_wr_timeout_val	R/W	After an AXI write transaction is granted, a timeout counter starts to count. When it counts to axi<n>_wr_timeout_val, the corresponding channel has the highest priority.	0x0
11:10	Reserved			
12	axi3_wr_timeout_en	R/W	If set to 1, enables the timeout function for the write channel of port n.	0x0
31:13	Reserved			

基址 0x0800_8000

AXI_MON0_CTRL

Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	axi_mon0_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon0_clear	R/W	Clear all the counter.	0x0
2	axi_mon0_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon0_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon0_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon0_irq	RO	Assert when all axi_mon<n>_hit_sel suscces.	
31:8	Reserved			

AXI_MON0_INPUT

Offset Address: 0x004

Bits	Name	Access	Description	Reset
5:0	axi_mon0_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0

Bits	Name	Access	Description	Reset
31:6	Reserved			

AXI_MON0_FILTER0

Offset Address: 0x010

Bits	Name	Access	Description	Reset
9:0	axi_mon0_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

AXI_MON0_FILTER1

Offset Address: 0x014

Bits	Name	Access	Description	Reset
31:0	axi_mon0_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

AXI_MON0_FILTER2

Offset Address: 0x018

Bits	Name	Access	Description	Reset
7:0	axi_mon0_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

AXI_MON0_FILTER3

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
31:0	axi_mon0_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

AXI_MON0_FILTER4

Offset Address: 0x020

Bits	Name	Access	Description	Reset
7:0	axi_mon0_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

AXI_MON0_FILTER5

Offset Address: 0x024

Bits	Name	Access	Description	Reset
23:0	axi_mon0_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

AXI_MON0_FILTER6

Offset Address: 0x028

Bits	Name	Access	Description	Reset
23:0	axi_mon0_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

AXI_MON0_FILTER7

Offset Address: 0x02c

Bits	Name	Access	Description	Reset
7:0	axi_mon0_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon0_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon0_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

AXI_MON0_FILTER8

Offset Address: 0x030

Bits	Name	Access	Description	Reset
0	axi_mon0_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon0_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon0_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon0_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

AXI_MON0_RPT0

Offset Address: 0x040

Bits	Name	Access	Description	Reset
31:0	axi_mon0_cycle_count	RO	AXI monitor 0 cycle count, counting after func_en assert	

AXI_MON0_RPT1

Offset Address: 0x044

Bits	Name	Access	Description	Reset
31:0	axi_mon0_hit_count	RO	AXI monitor 0 hit count, counting after func_en assert	

AXI_MON0_RPT2

Offset Address: 0x048

Bits	Name	Access	Description	Reset
31:0	axi_mon0_byte_count	RO	AXI monitor 0 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

AXI_MON0_RPT3

Offset Address: 0x04c

Bits	Name	Access	Description	Reset
31:0	axi_mon0_latency_count	RO	AXI monitor 0 latency count, counting after func_en assert, += oustanding	

AXI_MON1_CTRL

Offset Address: 0x080

Bits	Name	Access	Description	Reset
0	axi_mon1_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon1_clear	R/W	Clear all the counter.	0x0
2	axi_mon1_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon1_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon1_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon1_irq	RO	Assert when all axi_mon<n>_hit_sel succes.	
31:8	Reserved			

AXI_MON1_INPUT

Offset Address: 0x084

Bits	Name	Access	Description	Reset
5:0	axi_mon1_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

AXI_MON1_FILTER0

Offset Address: 0x090

Bits	Name	Access	Description	Reset
9:0	axi_mon1_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

AXI_MON1_FILTER1

Offset Address: 0x094

Bits	Name	Access	Description	Reset

Bits	Name	Access	Description	Reset
31:0	axi_mon1_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

AXI_MON1_FILTER2

Offset Address: 0x098

Bits	Name	Access	Description	Reset
7:0	axi_mon1_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

AXI_MON1_FILTER3

Offset Address: 0x09c

Bits	Name	Access	Description	Reset
31:0	axi_mon1_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

AXI_MON1_FILTER4

Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
7:0	axi_mon1_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

AXI_MON1_FILTER5

Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
23:0	axi_mon1_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

AXI_MON1_FILTER6

Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
23:0	axi_mon1_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

AXI_MON1_FILTER7

Offset Address: 0x0ac

Bits	Name	Access	Description	Reset
7:0	axi_mon1_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon1_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon1_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

AXI_MON1_FILTER8

Offset Address: 0x0b0

Bits	Name	Access	Description	Reset
0	axi_mon1_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon1_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon1_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon1_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

AXI_MON1_RPT0

Offset Address: 0x0c0

Bits	Name	Access	Description	Reset
31:0	axi_mon1_cycle_count	RO	AXI monitor 1 cycle count, counting after func_en assert	

AXI_MON1_RPT1

Offset Address: 0x0c4

Bits	Name	Access	Description	Reset
31:0	axi_mon1_hit_count	RO	AXI monitor 1 hit count, counting after func_en assert	

AXI_MON1_RPT2

Offset Address: 0x0c8

Bits	Name	Access	Description	Reset
31:0	axi_mon1_byte_count	RO	AXI monitor 1 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

AXI_MON1_RPT3

Offset Address: 0x0cc

Bits	Name	Access	Description	Reset
31:0	axi_mon1_latency_count	RO	AXI monitor 1 latency count, counting after func_en assert, += oustanding	

AXI_MON2_CTRL

Offset Address: 0x100

Bits	Name	Access	Description	Reset
0	axi_mon2_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon2_clear	R/W	Clear all the counter.	0x0
2	axi_mon2_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon2_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon2_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon2_irq	RO	Assert when all axi_mon<n>_hit_sel succes.	

Bits	Name	Access	Description	Reset
31:8	Reserved			

AXI_MON2_INPUT

Offset Address: 0x104

Bits	Name	Access	Description	Reset
5:0	axi_mon2_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

AXI_MON2_FILTER0

Offset Address: 0x110

Bits	Name	Access	Description	Reset
9:0	axi_mon2_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

AXI_MON2_FILTER1

Offset Address: 0x114

Bits	Name	Access	Description	Reset
31:0	axi_mon2_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

AXI_MON2_FILTER2

Offset Address: 0x118

Bits	Name	Access	Description	Reset
7:0	axi_mon2_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

AXI_MON2_FILTER3

Offset Address: 0x11c

Bits	Name	Access	Description	Reset
31:0	axi_mon2_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

AXI_MON2_FILTER4

Offset Address: 0x120

Bits	Name	Access	Description	Reset
7:0	axi_mon2_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

AXI_MON2_FILTER5

Offset Address: 0x124

Bits	Name	Access	Description	Reset
23:0	axi_mon2_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

AXI_MON2_FILTER6

Offset Address: 0x128

Bits	Name	Access	Description	Reset
23:0	axi_mon2_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

AXI_MON2_FILTER7

Offset Address: 0x12c

Bits	Name	Access	Description	Reset
7:0	axi_mon2_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon2_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon2_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

AXI_MON2_FILTER8

Offset Address: 0x130

Bits	Name	Access	Description	Reset
0	axi_mon2_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon2_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon2_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon2_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

AXI_MON2_RPT0

Offset Address: 0x140

Bits	Name	Access	Description	Reset
31:0	axi_mon2_cycle_count	RO	AXI monitor 2 cycle count, counting after func_en assert	

AXI_MON2_RPT1

Offset Address: 0x144

Bits	Name	Access	Description	Reset
31:0	axi_mon2_hit_count	RO	AXI monitor 2 hit count, counting after func_en assert	

AXI_MON2_RPT2

Offset Address: 0x148

Bits	Name	Access	Description	Reset
31:0	axi_mon2_byte_count	RO	AXI monitor 2 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

AXI_MON2_RPT3

Offset Address: 0x14c

Bits	Name	Access	Description	Reset
31:0	axi_mon2_latency_count	RO	AXI monitor 2 latency count, counting after func_en assert, += oustanding	

AXI_MON3_CTRL

Offset Address: 0x180

Bits	Name	Access	Description	Reset
0	axi_mon3_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon3_clear	R/W	Clear all the counter.	0x0
2	axi_mon3_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon3_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon3_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon3_irq	RO	Assert when all axi_mon<n>_hit_sel suscces.	
31:8	Reserved			

AXI_MON3_INPUT

Offset Address: 0x184

Bits	Name	Access	Description	Reset
5:0	axi_mon3_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

AXI_MON3_FILTER0

Offset Address: 0x190

Bits	Name	Access	Description	Reset
9:0	axi_mon3_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0

Bits	Name	Access	Description	Reset
31:10	Reserved			

AXI_MON3_FILTER1

Offset Address: 0x194

Bits	Name	Access	Description	Reset
31:0	axi_mon3_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

AXI_MON3_FILTER2

Offset Address: 0x198

Bits	Name	Access	Description	Reset
7:0	axi_mon3_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

AXI_MON3_FILTER3

Offset Address: 0x19c

Bits	Name	Access	Description	Reset
31:0	axi_mon3_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

AXI_MON3_FILTER4

Offset Address: 0x1a0

Bits	Name	Access	Description	Reset
7:0	axi_mon3_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

AXI_MON3_FILTER5

Offset Address: 0x1a4

Bits	Name	Access	Description	Reset
23:0	axi_mon3_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

AXI_MON3_FILTER6

Offset Address: 0x1a8

Bits	Name	Access	Description	Reset
23:0	axi_mon3_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

AXI_MON3_FILTER7

Offset Address: 0x1ac

Bits	Name	Access	Description	Reset
7:0	axi_mon3_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon3_hit_size	R/W	hit = Ax_size == hit_size	0x0

Bits	Name	Access	Description	Reset
11	Reserved			
13:12	axi_mon3_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

AXI_MON3_FILTER8

Offset Address: 0x1b0

Bits	Name	Access	Description	Reset
0	axi_mon3_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon3_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon3_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon3_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

AXI_MON3_RPT0

Offset Address: 0x1c0

Bits	Name	Access	Description	Reset
31:0	axi_mon3_cycle_count	RO	AXI monitor 3 cycle count, counting after func_en assert	

AXI_MON3_RPT1

Offset Address: 0x1c4

Bits	Name	Access	Description	Reset
31:0	axi_mon3_hit_count	RO	AXI monitor 3 hit count, counting after func_en assert	

AXI_MON3_RPT2

Offset Address: 0x1c8

Bits	Name	Access	Description	Reset
31:0	axi_mon3_byte_count	RO	AXI monitor 3 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

AXI_MON3_RPT3

Offset Address: 0x1cc

Bits	Name	Access	Description	Reset
31:0	axi_mon3_latency_count	RO	AXI monitor 3 latency count, counting after func_en assert, += oustanding	

AXI_MON4_CTRL

Offset Address: 0x200

Bits	Name	Access	Description	Reset
0	axi_mon4_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon4_clear	R/W	Clear all the counter.	0x0
2	axi_mon4_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0

Bits	Name	Access	Description	Reset
4	axi_mon4_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon4_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon4_irq	RO	Assert when all axi_mon<n>_hit_sel suscces.	
31:8	Reserved			

AXI_MON4_INPUT

Offset Address: 0x204

Bits	Name	Access	Description	Reset
5:0	axi_mon4_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

AXI_MON4_FILTER0

Offset Address: 0x210

Bits	Name	Access	Description	Reset
9:0	axi_mon4_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

AXI_MON4_FILTER1

Offset Address: 0x214

Bits	Name	Access	Description	Reset
31:0	axi_mon4_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

AXI_MON4_FILTER2

Offset Address: 0x218

Bits	Name	Access	Description	Reset
7:0	axi_mon4_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

AXI_MON4_FILTER3

Offset Address: 0x21c

Bits	Name	Access	Description	Reset
31:0	axi_mon4_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

AXI_MON4_FILTER4

Offset Address: 0x220

Bits	Name	Access	Description	Reset
7:0	axi_mon4_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

AXI_MON4_FILTER5

Offset Address: 0x224

Bits	Name	Access	Description	Reset
23:0	axi_mon4_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

AXI_MON4_FILTER6

Offset Address: 0x228

Bits	Name	Access	Description	Reset
23:0	axi_mon4_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

AXI_MON4_FILTER7

Offset Address: 0x22c

Bits	Name	Access	Description	Reset
7:0	axi_mon4_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon4_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon4_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

AXI_MON4_FILTER8

Offset Address: 0x230

Bits	Name	Access	Description	Reset
0	axi_mon4_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon4_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon4_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon4_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

AXI_MON4_RPT0

Offset Address: 0x240

Bits	Name	Access	Description	Reset
31:0	axi_mon4_cycle_count	RO	AXI monitor 4 cycle count, counting after func_en assert	

AXI_MON4_RPT1

Offset Address: 0x244

Bits	Name	Access	Description	Reset
31:0	axi_mon4_hit_count	RO	AXI monitor 4 hit count, counting after func_en assert	

AXI_MON4_RPT2

Offset Address: 0x248

Bits	Name	Access	Description	Reset
31:0	axi_mon4_byte_count	RO	AXI monitor 4 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

AXI_MON4_RPT3

Offset Address: 0x24c

Bits	Name	Access	Description	Reset
31:0	axi_mon4_latency_count	RO	AXI monitor 4 latency count, counting after func_en assert, += oustanding	

AXI_MON5_CTRL

Offset Address: 0x280

Bits	Name	Access	Description	Reset
0	axi_mon5_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon5_clear	R/W	Clear all the counter.	0x0
2	axi_mon5_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon5_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon5_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon5_irq	RO	Assert when all axi_mon<n>_hit_sel succes.	
31:8	Reserved			

AXI_MON5_INPUT

Offset Address: 0x284

Bits	Name	Access	Description	Reset
5:0	axi_mon5_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

AXI_MON5_FILTER0

Offset Address: 0x290

Bits	Name	Access	Description	Reset
9:0	axi_mon5_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len	0x0

Bits	Name	Access	Description	Reset
			bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	
31:10	Reserved			

AXI_MON5_FILTER1

Offset Address: 0x294

Bits	Name	Access	Description	Reset
31:0	axi_mon5_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

AXI_MON5_FILTER2

Offset Address: 0x298

Bits	Name	Access	Description	Reset
7:0	axi_mon5_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

AXI_MON5_FILTER3

Offset Address: 0x29c

Bits	Name	Access	Description	Reset
31:0	axi_mon5_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

AXI_MON5_FILTER4

Offset Address: 0x2a0

Bits	Name	Access	Description	Reset
7:0	axi_mon5_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

AXI_MON5_FILTER5

Offset Address: 0x2a4

Bits	Name	Access	Description	Reset
23:0	axi_mon5_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

AXI_MON5_FILTER6

Offset Address: 0x2a8

Bits	Name	Access	Description	Reset
23:0	axi_mon5_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

AXI_MON5_FILTER7

Offset Address: 0x2ac

Bits	Name	Access	Description	Reset
7:0	axi_mon5_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon5_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon5_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

AXI_MON5_FILTER8

Offset Address: 0x2b0

Bits	Name	Access	Description	Reset
0	axi_mon5_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon5_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon5_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon5_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

AXI_MON5_RPT0

Offset Address: 0x2c0

Bits	Name	Access	Description	Reset
31:0	axi_mon5_cycle_count	RO	AXI monitor 5 cycle count, counting after func_en assert	

AXI_MON5_RPT1

Offset Address: 0x2c4

Bits	Name	Access	Description	Reset
31:0	axi_mon5_hit_count	RO	AXI monitor 5 hit count, counting after func_en assert	

AXI_MON5_RPT2

Offset Address: 0x2c8

Bits	Name	Access	Description	Reset
31:0	axi_mon5_byte_count	RO	AXI monitor 5 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

AXI_MON5_RPT3

Offset Address: 0x2cc

Bits	Name	Access	Description	Reset
31:0	axi_mon5_latency_count	RO	AXI monitor 5 latency count, counting after func_en assert, += oustanding	

AXI_MON6_CTRL

Offset Address: 0x300

Bits	Name	Access	Description	Reset

Bits	Name	Access	Description	Reset
0	axi_mon6_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon6_clear	R/W	Clear all the counter.	0x0
2	axi_mon6_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon6_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon6_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon6_irq	RO	Assert when all axi_mon<n>_hit_sel succes.	
31:8	Reserved			

AXI_MON6_INPUT

Offset Address: 0x304

Bits	Name	Access	Description	Reset
5:0	axi_mon6_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

AXI_MON6_FILTER0

Offset Address: 0x310

Bits	Name	Access	Description	Reset
9:0	axi_mon6_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

AXI_MON6_FILTER1

Offset Address: 0x314

Bits	Name	Access	Description	Reset
31:0	axi_mon6_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

AXI_MON6_FILTER2

Offset Address: 0x318

Bits	Name	Access	Description	Reset
7:0	axi_mon6_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

Bits	Name	Access	Description	Reset
31:8	Reserved			

AXI_MON6_FILTER3

Offset Address: 0x31c

Bits	Name	Access	Description	Reset
31:0	axi_mon6_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

AXI_MON6_FILTER4

Offset Address: 0x320

Bits	Name	Access	Description	Reset
7:0	axi_mon6_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

AXI_MON6_FILTER5

Offset Address: 0x324

Bits	Name	Access	Description	Reset
23:0	axi_mon6_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

AXI_MON6_FILTER6

Offset Address: 0x328

Bits	Name	Access	Description	Reset
23:0	axi_mon6_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

AXI_MON6_FILTER7

Offset Address: 0x32c

Bits	Name	Access	Description	Reset
7:0	axi_mon6_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon6_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon6_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

AXI_MON6_FILTER8

Offset Address: 0x330

Bits	Name	Access	Description	Reset
0	axi_mon6_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon6_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon6_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon6_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

AXI_MON6_RPT0

Offset Address: 0x340

Bits	Name	Access	Description	Reset
31:0	axi_mon6_cycle_count	RO	AXI monitor 6 cycle count, counting after func_en assert	

AXI_MON6_RPT1

Offset Address: 0x344

Bits	Name	Access	Description	Reset
31:0	axi_mon6_hit_count	RO	AXI monitor 6 hit count, counting after func_en assert	

AXI_MON6_RPT2

Offset Address: 0x348

Bits	Name	Access	Description	Reset
31:0	axi_mon6_byte_count	RO	AXI monitor 6 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

AXI_MON6_RPT3

Offset Address: 0x34c

Bits	Name	Access	Description	Reset
31:0	axi_mon6_latency_count	RO	AXI monitor 6 latency count, counting after func_en assert, += oustanding	

AXI_MON7_CTRL

Offset Address: 0x380

Bits	Name	Access	Description	Reset
0	axi_mon7_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon7_clear	R/W	Clear all the counter.	0x0
2	axi_mon7_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon7_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon7_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon7_irq	RO	Assert when all axi_mon<n>_hit_sel succes.	
31:8	Reserved			

AXI_MON7_INPUT

Offset Address: 0x384

Bits	Name	Access	Description	Reset
5:0	axi_mon7_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

AXI_MON7_FILTER0

Offset Address: 0x390

Bits	Name	Access	Description	Reset
9:0	axi_mon7_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

AXI_MON7_FILTER1

Offset Address: 0x394

Bits	Name	Access	Description	Reset
31:0	axi_mon7_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

AXI_MON7_FILTER2

Offset Address: 0x398

Bits	Name	Access	Description	Reset
7:0	axi_mon7_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

AXI_MON7_FILTER3

Offset Address: 0x39c

Bits	Name	Access	Description	Reset
31:0	axi_mon7_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

AXI_MON7_FILTER4

Offset Address: 0x3a0

Bits	Name	Access	Description	Reset
7:0	axi_mon7_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

AXI_MON7_FILTER5

Offset Address: 0x3a4

Bits	Name	Access	Description	Reset
23:0	axi_mon7_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

AXI_MON7_FILTER6

Offset Address: 0x3a8

Bits	Name	Access	Description	Reset
23:0	axi_mon7_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

AXI_MON7_FILTER7

Offset Address: 0x3ac

Bits	Name	Access	Description	Reset
7:0	axi_mon7_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon7_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon7_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

AXI_MON7_FILTER8

Offset Address: 0x3b0

Bits	Name	Access	Description	Reset
0	axi_mon7_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon7_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon7_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon7_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

AXI_MON7_RPT0

Offset Address: 0x3c0

Bits	Name	Access	Description	Reset
31:0	axi_mon7_cycle_count	RO	AXI monitor 7 cycle count, counting after func_en assert	

AXI_MON7_RPT1

Offset Address: 0x3c4

Bits	Name	Access	Description	Reset
31:0	axi_mon7_hit_count	RO	AXI monitor 7 hit count, counting after func_en assert	

AXI_MON7_RPT2

Offset Address: 0x3c8

Bits	Name	Access	Description	Reset
31:0	axi_mon7_byte_count	RO	AXI monitor 7 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

AXI_MON7_RPT3

Offset Address: 0x3cc

Bits	Name	Access	Description	Reset

Bits	Name	Access	Description	Reset
31:0	axi_mon7_latency_count	RO	AXI monitor 7 latency count, counting after func_en assert, += oustanding	

AXI_MON8_CTRL

Offset Address: 0x400

Bits	Name	Access	Description	Reset
0	axi_mon8_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon8_clear	R/W	Clear all the counter.	0x0
2	axi_mon8_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon8_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon8_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon8_irq	RO	Assert when all axi_mon<n>_hit_sel suscces.	
31:8	Reserved			

AXI_MON8_INPUT

Offset Address: 0x404

Bits	Name	Access	Description	Reset
5:0	axi_mon8_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

AXI_MON8_FILTER0

Offset Address: 0x410

Bits	Name	Access	Description	Reset
9:0	axi_mon8_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

AXI_MON8_FILTER1

Offset Address: 0x414

Bits	Name	Access	Description	Reset
31:0	axi_mon8_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

AXI_MON8_FILTER2

Offset Address: 0x418

Bits	Name	Access	Description	Reset
7:0	axi_mon8_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

AXI_MON8_FILTER3

Offset Address: 0x41c

Bits	Name	Access	Description	Reset
31:0	axi_mon8_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

AXI_MON8_FILTER4

Offset Address: 0x420

Bits	Name	Access	Description	Reset
7:0	axi_mon8_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

AXI_MON8_FILTER5

Offset Address: 0x424

Bits	Name	Access	Description	Reset
23:0	axi_mon8_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

AXI_MON8_FILTER6

Offset Address: 0x428

Bits	Name	Access	Description	Reset
23:0	axi_mon8_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

AXI_MON8_FILTER7

Offset Address: 0x42c

Bits	Name	Access	Description	Reset
7:0	axi_mon8_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon8_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon8_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

AXI_MON8_FILTER8

Offset Address: 0x430

Bits	Name	Access	Description	Reset
0	axi_mon8_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon8_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0

Bits	Name	Access	Description	Reset
10:8	axi_mon8_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon8_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

AXI_MON8_RPT0

Offset Address: 0x440

Bits	Name	Access	Description	Reset
31:0	axi_mon8_cycle_count	RO	AXI monitor 8 cycle count, counting after func_en assert	

AXI_MON8_RPT1

Offset Address: 0x444

Bits	Name	Access	Description	Reset
31:0	axi_mon8_hit_count	RO	AXI monitor 8 hit count, counting after func_en assert	

AXI_MON8_RPT2

Offset Address: 0x448

Bits	Name	Access	Description	Reset
31:0	axi_mon8_byte_count	RO	AXI monitor 8 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

AXI_MON8_RPT3

Offset Address: 0x44c

Bits	Name	Access	Description	Reset
31:0	axi_mon8_latency_count	RO	AXI monitor 8 latency count, counting after func_en assert, += oustanding	

AXI_MON9_CTRL

Offset Address: 0x480

Bits	Name	Access	Description	Reset
0	axi_mon9_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon9_clear	R/W	Clear all the counter.	0x0
2	axi_mon9_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon9_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon9_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon9_irq	RO	Assert when all axi_mon<n>_hit_sel suscces.	
31:8	Reserved			

AXI_MON9_INPUT

Offset Address: 0x484

Bits	Name	Access	Description	Reset
5:0	axi_mon9_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

AXI_MON9_FILTER0

Offset Address: 0x490

Bits	Name	Access	Description	Reset
9:0	axi_mon9_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

AXI_MON9_FILTER1

Offset Address: 0x494

Bits	Name	Access	Description	Reset
31:0	axi_mon9_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

AXI_MON9_FILTER2

Offset Address: 0x498

Bits	Name	Access	Description	Reset
7:0	axi_mon9_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

AXI_MON9_FILTER3

Offset Address: 0x49c

Bits	Name	Access	Description	Reset
31:0	axi_mon9_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

AXI_MON9_FILTER4

Offset Address: 0x4a0

Bits	Name	Access	Description	Reset
7:0	axi_mon9_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

AXI_MON9_FILTER5

Offset Address: 0x4a4

Bits	Name	Access	Description	Reset
23:0	axi_mon9_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

AXI_MON9_FILTER6

Offset Address: 0x4a8

Bits	Name	Access	Description	Reset
23:0	axi_mon9_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

AXI_MON9_FILTER7

Offset Address: 0x4ac

Bits	Name	Access	Description	Reset
7:0	axi_mon9_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon9_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon9_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

AXI_MON9_FILTER8

Offset Address: 0x4b0

Bits	Name	Access	Description	Reset
0	axi_mon9_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon9_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon9_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon9_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

AXI_MON9_RPT0

Offset Address: 0x4c0

Bits	Name	Access	Description	Reset
31:0	axi_mon9_cycle_count	RO	AXI monitor 9 cycle count, counting after func_en assert	

AXI_MON9_RPT1

Offset Address: 0x4c4

Bits	Name	Access	Description	Reset
31:0	axi_mon9_hit_count	RO	AXI monitor 9 hit count, counting after func_en assert	

AXI_MON9_RPT2

Offset Address: 0x4c8

Bits	Name	Access	Description	Reset

Bits	Name	Access	Description	Reset
31:0	axi_mon9_byte_count	RO	AXI monitor 9 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

AXI_MON9_RPT3

Offset Address: 0x4cc

Bits	Name	Access	Description	Reset
31:0	axi_mon9_latency_count	RO	AXI monitor 9 latency count, counting after func_en assert, += oustanding	

AXI_MON10_CTRL

Offset Address: 0x500

Bits	Name	Access	Description	Reset
0	axi_mon10_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon10_clear	R/W	Clear all the counter.	0x0
2	axi_mon10_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon10_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon10_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon10_irq	RO	Assert when all axi_mon<n>_hit_sel suscces.	
31:8	Reserved			

AXI_MON10_INPUT

Offset Address: 0x504

Bits	Name	Access	Description	Reset
5:0	axi_mon10_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

AXI_MON10_FILTER0

Offset Address: 0x510

Bits	Name	Access	Description	Reset
9:0	axi_mon10_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

AXI_MON10_FILTER1

Offset Address: 0x514

Bits	Name	Access	Description	Reset
31:0	axi_mon10_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

AXI_MON10_FILTER2

Offset Address: 0x518

Bits	Name	Access	Description	Reset
7:0	axi_mon10_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

AXI_MON10_FILTER3

Offset Address: 0x51c

Bits	Name	Access	Description	Reset
31:0	axi_mon10_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

AXI_MON10_FILTER4

Offset Address: 0x520

Bits	Name	Access	Description	Reset
7:0	axi_mon10_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

AXI_MON10_FILTER5

Offset Address: 0x524

Bits	Name	Access	Description	Reset
23:0	axi_mon10_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

AXI_MON10_FILTER6

Offset Address: 0x528

Bits	Name	Access	Description	Reset
23:0	axi_mon10_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

AXI_MON10_FILTER7

Offset Address: 0x52c

Bits	Name	Access	Description	Reset
7:0	axi_mon10_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon10_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon10_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

AXI_MON10_FILTER8

Offset Address: 0x530

Bits	Name	Access	Description	Reset
0	axi_mon10_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon10_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon10_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon10_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

AXI_MON10_RPT0

Offset Address: 0x540

Bits	Name	Access	Description	Reset
31:0	axi_mon10_cycle_count	RO	AXI monitor 10 cycle count, counting after func_en assert	

AXI_MON10_RPT1

Offset Address: 0x544

Bits	Name	Access	Description	Reset
31:0	axi_mon10_hit_count	RO	AXI monitor 10 hit count, counting after func_en assert	

AXI_MON10_RPT2

Offset Address: 0x548

Bits	Name	Access	Description	Reset
31:0	axi_mon10_byte_count	RO	AXI monitor 10 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

AXI_MON10_RPT3

Offset Address: 0x54c

Bits	Name	Access	Description	Reset
31:0	axi_mon10_latency_count	RO	AXI monitor 10 latency count, counting after func_en assert, += oustanding	

AXI_MON11_CTRL

Offset Address: 0x580

Bits	Name	Access	Description	Reset
0	axi_mon11_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon11_clear	R/W	Clear all the counter.	0x0
2	axi_mon11_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon11_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon11_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			

Bits	Name	Access	Description	Reset
7	axi_mon11_irq	RO	Assert when all axi_mon<n>_hit_sel succes.	
31:8	Reserved			

AXI_MON11_INPUT

Offset Address: 0x584

Bits	Name	Access	Description	Reset
5:0	axi_mon11_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

AXI_MON11_FILTER0

Offset Address: 0x590

Bits	Name	Access	Description	Reset
9:0	axi_mon11_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

AXI_MON11_FILTER1

Offset Address: 0x594

Bits	Name	Access	Description	Reset
31:0	axi_mon11_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

AXI_MON11_FILTER2

Offset Address: 0x598

Bits	Name	Access	Description	Reset
7:0	axi_mon11_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

AXI_MON11_FILTER3

Offset Address: 0x59c

Bits	Name	Access	Description	Reset
31:0	axi_mon11_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

AXI_MON11_FILTER4

Offset Address: 0x5a0

Bits	Name	Access	Description	Reset
7:0	axi_mon11_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

AXI_MON11_FILTER5

Offset Address: 0x5a4

Bits	Name	Access	Description	Reset
23:0	axi_mon11_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

AXI_MON11_FILTER6

Offset Address: 0x5a8

Bits	Name	Access	Description	Reset
23:0	axi_mon11_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

AXI_MON11_FILTER7

Offset Address: 0x5ac

Bits	Name	Access	Description	Reset
7:0	axi_mon11_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon11_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon11_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

AXI_MON11_FILTER8

Offset Address: 0x5b0

Bits	Name	Access	Description	Reset
0	axi_mon11_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon11_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon11_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon11_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

AXI_MON11_RPT0

Offset Address: 0x5c0

Bits	Name	Access	Description	Reset
31:0	axi_mon11_cycle_count	RO	AXI monitor 11 cycle count, counting after func_en assert	

AXI_MON11_RPT1

Offset Address: 0x5c4

Bits	Name	Access	Description	Reset

Bits	Name	Access	Description	Reset
31:0	axi_mon11_hit_count	RO	AXI monitor 11 hit count, counting after func_en assert	

AXI_MON11_RPT2

Offset Address: 0x5c8

Bits	Name	Access	Description	Reset
31:0	axi_mon11_byte_count	RO	AXI monitor 11 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

AXI_MON11_RPT3

Offset Address: 0x5cc

Bits	Name	Access	Description	Reset
31:0	axi_mon11_latency_count	RO	AXI monitor 11 latency count, counting after func_en assert, += oustanding	

4.1.6 DDRC Register

4.1.6.1 DDRC Register Overview

Base Address: 0x0800_4000

Name	Address Offset	Description
DRAM_REF_CTRL	0x064	DRAM refresh parameter
DRAM_MRD0	0x0dc	DRAM MR value
DRAM_MRD1	0x0e0	DRAM MR value

4.1.6.2 DDRC Register Description

Base Address: 0x0800_4000

DRAM_REF_CTRL

Offset Address: 0x064

Bits	Name	Access	Description	Reset
9:0	t_rfc	R/W	Specify tRFC Unit: ddr core clock cycles	0x8c
15:10	Reserved			
27:16	t_refi	R/W	Specify tREFI Unit: 32 ddr core clocks	0x62
31:28	Reserved			

DRAM_MRD0

Offset Address: 0x0dc

Bits	Name	Access	Description	Reset
15:0	ddr_mr1	R/W	DDR3: Write value for MR1 register	0x510
31:16	ddr_mr0	R/W	DDR3: Write value for MR0 register	0x0

DRAM_MRD1

Offset Address: 0x0e0

Bits	Name	Access	Description	Reset
15:0	ddr_mr3	R/W	DDR3: Write value for MR3 register	0x0
31:16	ddr_mr2	R/W	DDR3: Write value for MR2 register	0x0

4.2 SPI NOR Flash Controller

4.2.1 Overview

Support external SPI NOR flash data access.

4.2.2 Characteristic

- Support one SPI NOR chip select.
- Support Dual/Qual read/write operation.
- Support various specifications of devices.
  - Support 3 Byte address device and 4 Byte address device.
  - Suppport up to 256MB capacity.
- SPI NOR is one of boot devices.

4.2.3 Function Description

4.2.3.1 Interface Description

SPI NOR Flash controller can support three SPI NOR interface types: Standard SPI, Dual SPI and Qual SPI interface mode.

- Standard SPI Interface Mode:

Standard SPI interface mode has 1 bit data input line and 1 bit data output line. The figure below shows the write and write operation sequence diagrams of standard SPI interface mode.

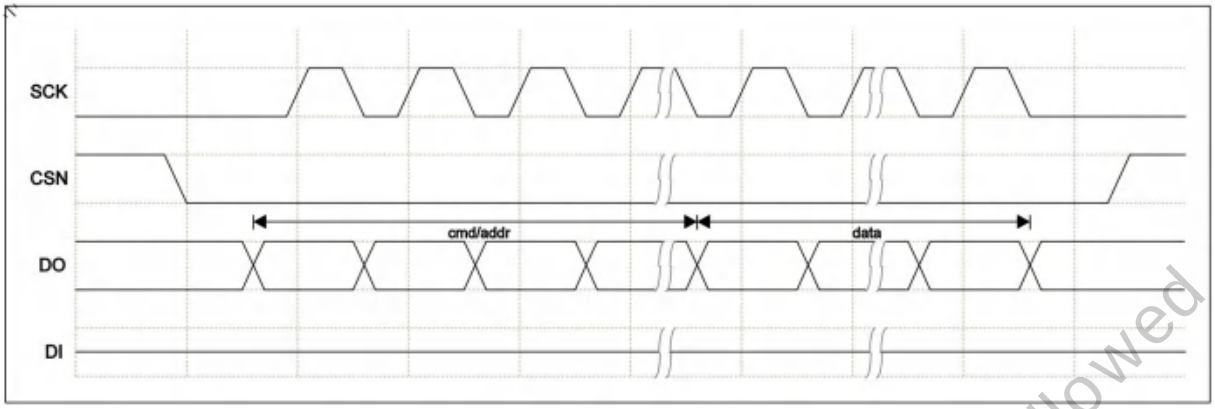


Figure 4- 2 Standard SPI Interface Mode Write Operation Sequence

Sequence description:

- command/address/dummy cycles are output on DO line in single bit serial mode.
- Data is output on DO line in single bit serial mode.

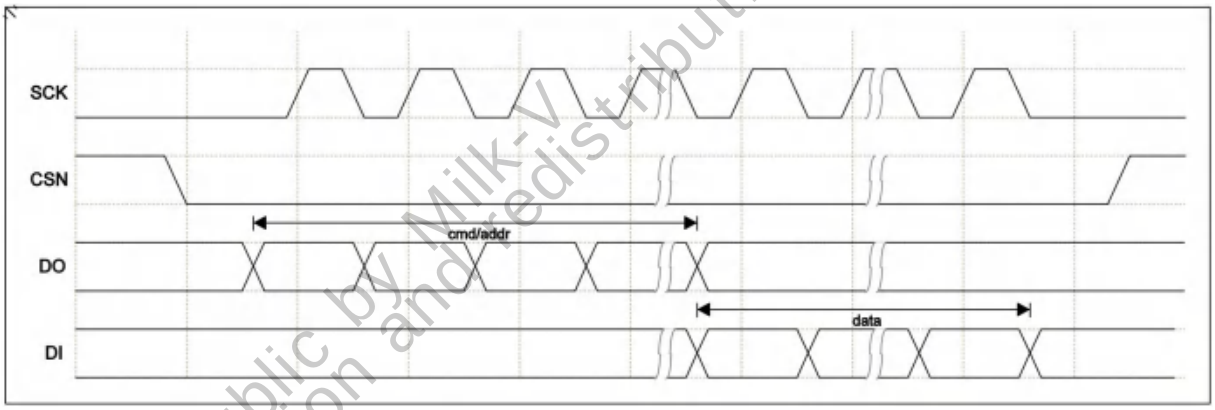


Figure 4- 3 Standard SPI Interface Mode Read Operation Sequence

Sequence description:

- Command/Address/Dummy Cycles are output on DO line in single bit serial mode.
- Data is output on DI line in single bit serial mode.

- Dual-Input SPI Interface Mode:

In Dual Input SPI interface mode, two bit data lines are paralleled in data input phase. The figure below shows the Dual Input SPI interface mode operation sequence diagram.

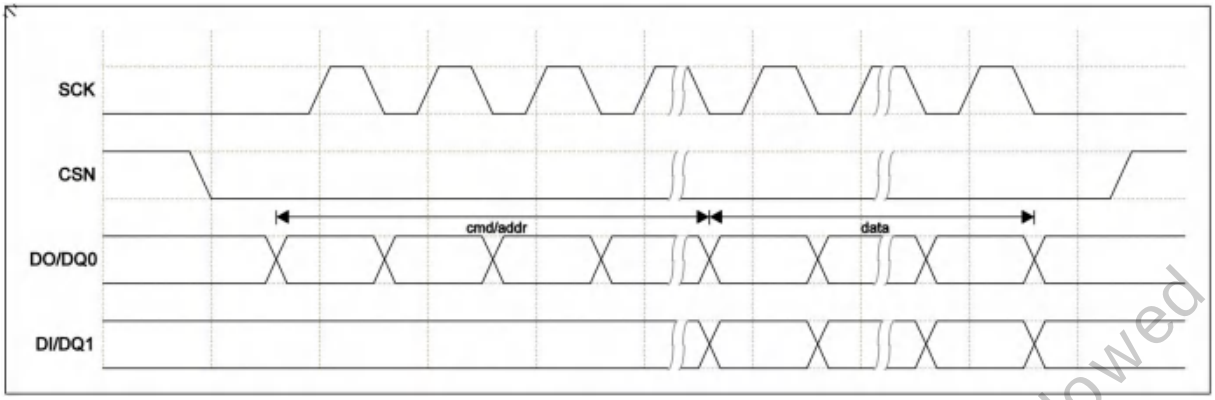


Figure 4- 4 Dual-Input SPI Interface Sequence

Sequence description:

- Command/Address/Dummy Cycles are output on DO line in single bit serial mode.
- Data is input (read) on the DO / DI line in 2 bits mode.

- Dual-IO SPI Interface Mode:

In Dual IO SPI interface mode, two bit data lines are paralleled in address output and data input stages. The figure below shows the Dual IO SPI interface mode operation sequence.

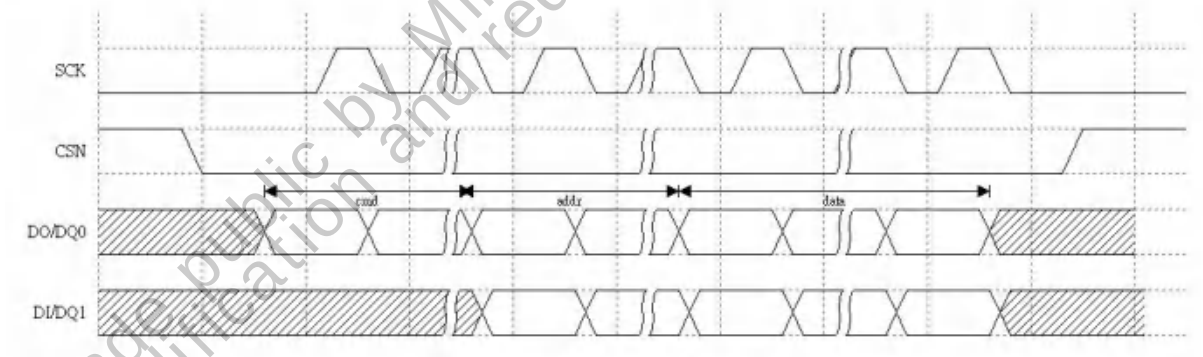


Figure 4- 5 Dual-IO SPI Interface Sequence

Sequence description:

- Command is output on DO line in single bit serial mode.
- Address/Dummy Cycles/Data output (write) or input (read) on DO / DI line in 2 bits mode.

- Quad-Input SPI Interface Mode:

In Quad Input SPI interface mode, 4 bit data line are paralleled in data input phase. The figure below shows the Quad Input SPI interface mode operation sequence.

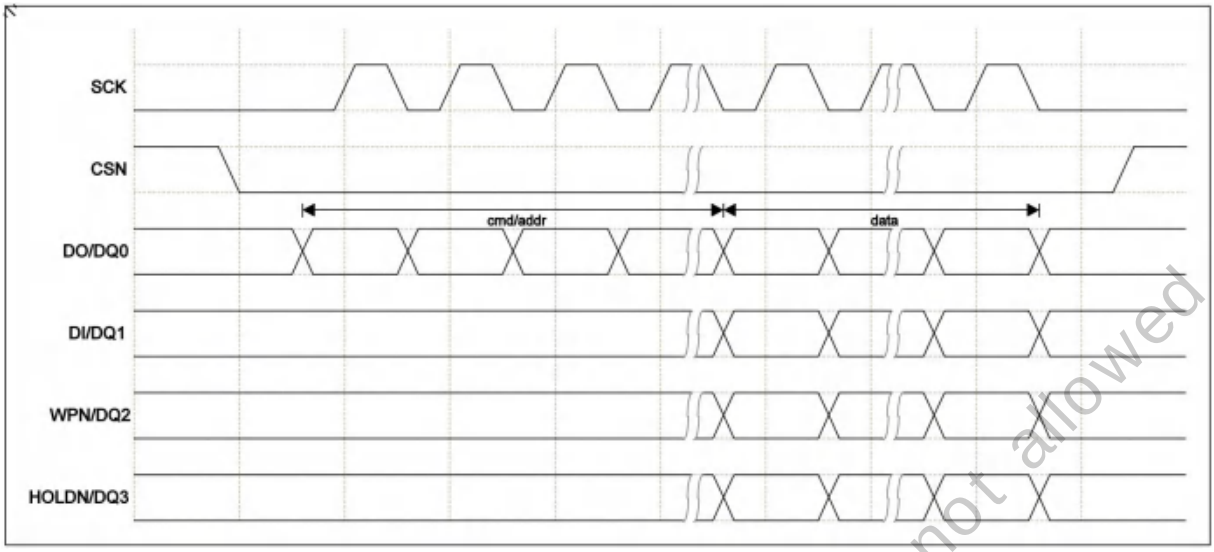


Figure 4- 6 Quad-Input SPI Interface Sequence

Sequence description:

- Command/Address/Dummy Cycles are output on DO line in single bit serial mode.
- Data is input (read) in DO / DI / WPN / HOLDN in 4 bits mode.

- Quad-IO SPI Interface Mode:

In Quad IO SPI interface mode, two bit data lines are paralleled in address output and data input stages. The figure below shows the Quad IO SPI interface mode operation sequence.

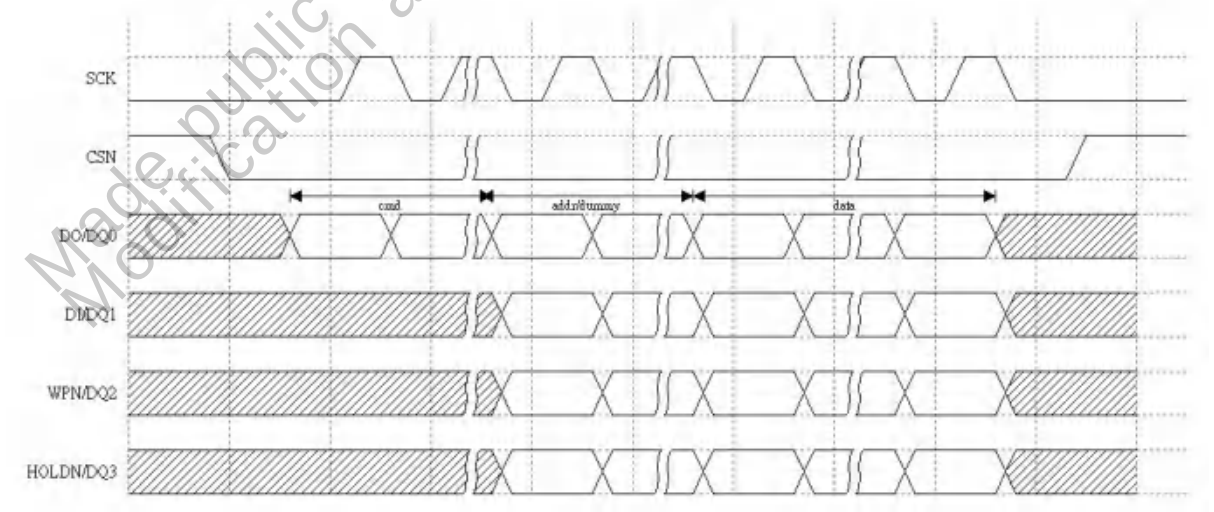


Figure 4- 7 Quad-IO SPI Interface Sequence

Sequence description:

- Command is output on DO line in single bit serial mode.
- Address/Dummy Cycles/Data is input (read) in DO / DI / WPN / HOLDN in 4 bits mode.

4.2.3.2 Boot Function

SPI NOR Boot data is located at chip address 0x1000_0000~0x1FFF_FFFF, directly mapped to SPI NOR flash continuous address space 0x0000_0000~0x0FFF_FFFF. The controller can support flashs up to 256MB, 4 bytes address mode is required if you need to use flashs larger than 16MB. The reset state of the controller is in 3 bytes address mode, and 4 bytes address mode could be enabled after configuration, so SPI_ NOR flash needs to support 3bytes / 4bytes address mode.

4.2.3.3 Register Operation

The user configures the controller register, such as operation command, address, etc., and finally configures reg_go_busy register to issue spi transition . The controller issues commands to the device according to registers configuration.

4.2.3.4 DMA Operation

- DMMR Reading Mode:

When SPI_NOR controller operating in dmmr mode, the content in flash is directly mapped to chip address space 0x1000_0000~0x1FFF_FFFF. The system DMA can use memory-to-memory mode to copy SPI_NOR data to DDR.

- Non-DMMR Read and Write Mode:

Instructions, addresses and data must be transmitted through FF_ PORT. Read instruction, write instruction, instruction length and data length should be configured first, and then write data to FF_PORT by CPU or DMA to issue instructions and addresses.

4.2.4 Workflow

4.2.4.1 Initialization Process

Step 1. Configure SPI clock divider according to the flash device and issued command.
Step 2. Configure the interrupt control register.

4.2.4.2 Device Status Register Operation

Step 1. Configure transmission data length.
Step 2. Configure transmission mode.
Step 3. Configure reg_go_busy.
Step 4. Write the transfer content to the cache.
Step 5. Check INT_STS and wait for the operation to be finished.

4.2.4.3 SPI NOR Flash Address Mode Switching Process

For SPI_NOR flash device, it supports 3-Byte and 4-Byte Flash address modes, and the mode can be dynamically switched when system is working. Please follow these steps to switch the flash address mode:

Step 1. Complete all operation on SPI_NOR flash device.
Step 2. According to the device requirements, configure the device's register in register operation mode to enable 4-byte mode.
Step 3. Configure the register [reg_byte4en] in SPI_NOR flash controller to 4-byte mode and complete the switch from 3 bytes address mode to 4 bytes address mode.

4.2.4.4 DMA Read Operation Flow

Step.1 Disable DMMR mode and DMA_EN mode.
Step.2 Write FF_PT as 1 to clear FIFO and reset read / write index.
Step.3 Enable DMMR mode

Step.4 Configure the system DMA as mem-to-mem transfer mode. DST_TR_WIDTH = 0x2 (transaction width is 32bit)、DST_MSIZE = 0x0 (burst transaction length =1)、BLOCK_TS = TRAN_NUM/4 -1 . Block_TS/ DST_TR_WIDTH/ DST_MSIZE should be configured according to the actual transmission length.
Step. 5 Enable system DMA channel and start memory moving
Step. 6 When the corresponding channel is interrupted, the DMA read is completed.

4.2.4.5 DMA Write Operation Flow

Step. 1 Disable DMMR mode and DMA_EN mode.
Step.2 Write FF_PT as 1 to clear FIFO and reset read / write index
Step.3 Configure the system DMA channel mapping and map the selected DMA channel to 39: dma_req_spi_nor.
Step.4 Configure the system DMA as mem-to-mem transfer mode. DST_TR_WIDTH = 0x2 (transaction width is 32bit)、DST_MSIZE = 0x0 (burst transaction length =1)、BLOCK_TS = TRAN_NUM/4 -1 . Block_TS/ DST_TR_WIDTH/ DST_MSIZE should be configured according to the actual transmission length.
Step. 5 Enable system DMA channel
Step. 6 Configure SPI_NOR Register TRAN_NUM, excluding instruction and address.
Step.7 Configure SPI_NOR Register TRAN_CSR，tran_mode = 0x2 (tx only), fast_mode, bus_width, addr_bn, dma_en=0 and reg_go_busy. Ex: TRAN_CSR = 0x0000BC2A
Step.8 Write to command address to SPI_NOR Register FF_Port.
Step.9 Query SPI_NOR Register rdat_ff_PT = 0, ensure that the command and address are sent.
Step.10 Configure SPI_NOR Register TRAN_CSR and enable dma_en.
Step.11 Detect SPI_NOR Register INT_STS and wait for the operation to complete. This indicates that the buffer content has been written to the device.

4.2.4.6 Other Reminders

Before the device operation is completed, do not change the relevant register configuration, otherwise the operation may be abnormal.

4.2.5 Register overview

Base Address 0x10000000

Name	Address Offset	Description
SPI_CTRL	0x000	SPI_NOR operation control
CE_CTRL	0x004	CE operation control
DLY_CTRL	0x008	Delay control
DMMR_CTRL	0x00c	DMMR mode contro
TRAN_CSR	0x010	Transmission control
TRAN_NUM	0x014	Transfer frame count
FF_PORT	0x018	FIFO write/read port
FF_PT	0x020	FIFO pointer status
INT_STS	0x028	Interrupt status
INT_EN	0x02c	Interrupt enable

4.2.6 Register Discription

SPI_CTRL

Offset Address: 0x000

Bits	Name	Access	Description	Reset
10:0	sck_div	R/W	SPI Clock Divider SCK frequency = HCLK frequency / (2(SckDiv+ 1))	0x9
11	Reserved			
12	cpha	R/W	Clock Phase 0：When the chip selection is valid, the first clock edge of SCK starts to sample data. 1：When the chip selection is valid, the second clock edge of SCK starts to sample data.	0x0
13	cpol	R/W	Clock Polarity 0：Low level when SCK is idle 1：High level when SCK is idle	0x0
14	hold_o	R/W	HOLD pin output level	0x1
15	wp_o	R/W	WP pin output level	0x1
19:16	frame_len	R/W	Frame length for sending and receiving	0x8

Bits	Name	Access	Description	Reset
20	lsb_first	R/W	LSBF: Least Significant Bit First 0: Frame MSB first 1: Frame LSB first	0x0
21	srst	R/W	Write 1 to reset all state machines and interrupt flag bits	0x0
31:22	Reserved			

CE_CTRL

Offset Address: 0x004

Bits	Name	Access	Description	Reset
0	ce_manual	R/W	CEManual controls the level of CE pin.	0x0
1	ce_manual_en	R/W	CE Manual Enable 0：The level of CE pin is controlled by hardware state machine 1：The level of CE pin is controlled by CEManual register.	0x0
31:2	Reserved			

DLY_CTRL

Offset Address: 0x008

Bits	Name	Access	Description	Reset
3:0	frame_interval	R/W	Control the frame interval between two adjacent frames of data: T = TSCK * FmIntvl (no SCK pulse in the frame interval). If the frame interval between two adjacent frames of data is 0, there is no frame interval.	0x0
7:4	Reserved			
11:8	cet	R/W	CET controls the time that CE is effective in advance of the first clock edge of SCK before the beginning of a transmission and the time that CE remains effective relative to the last clock edge of SCK after the end of transmission. This time is calculated as t = tsck * (CET + 1)	0x3
13:12	smp_en_dly	R/W	Receive Sampling Delay Option. Delay the sample cycle (in IP working clock) after the rising edge of SCK for sampling.	0x0
15:14	rx_pipe_ctrl	R/W	Receiving sampling clock edge options. 0: Normal sampling 1: Sampling at the negative edge of SCK for high-speed transmission	0x0
31:16	Reserved			

DMMR_CTRL

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
0	dmmr_mode	R/W	When the bit is 1, the read address on AHB will be directly mapped to SPI flash, and the controller will automatically	0x1

Bits	Name	Access	Description	Reset
			read data from the corresponding address of SPI flash SPI flash can be used as Rom. when dmmr is 1, register in IP can be written but not read.	
31:1	Reserved			

TRAN_CSR

Offset Address: 0x010

Bits	Name	Access	Description	Reset
1:0	tran_mode	R/W	Transfer Mode 00: No Tx, No Rx 01: Rx only 10: Tx only 11: Tx and Rx TranMode indicates the sending and receiving mode of transmission data except command and address	0x0
2	Reserved			
3	fast_mode	R/W	FastMode: 0: Normal Mode 1: Fast Mode	0x0
5:4	bus_width	R/W	Bus Width 00: 1 bit bus 01: 2 bit bus 10: 4 bit bus 11: Reserved	0x0
6	dma_en	R/W	0: DMA Disable 1: DMA Enable When tranmode is 11 (sending and receiving simultaneously), DMA transmission is not supported	0x0
7	miso_cked	R/W	Level value of miso_i pin	0x0
10:8	addr_bn	R/W	Address Byte Number represents the number of bytes of the current flash transfer address field, and 0 represents no address field.	0x3
11	with_cmd	R/W	With Command 0: Current transmission without command 1: Current transmission with command	0x1
13:12	ff_trg_lvl	R/W	FFTrgLvl controls under what conditions FIFO generates interrupts and DMA requests. 00：1 Byte 01：2 Bytes 10：4 Bytes 11：8 Bytes For transmit, when the number of free bytes in FIFO is greater than or equal to the number of bytes defined by fftrglvl, interrupt and DMA request are generated;	0x3

Bits	Name	Access	Description	Reset
			For receive, interrupt and DMA request are generated when the number of effective bytes in FIFO is greater than or equal to the number of bytes defined by FFtrglvl.	
14	Reserved			
15	go_busy	R/W	Writing 0 to this bit doesn't work. Writing 1 to this position 1 will start a transmission. After the transmission, this bit will be cleared automatically. Before initiating a new transmission, the software should query the register, and only when the register is 0 can a new transmission be initiated.	0x0
19:16	dummy_cyc	R/W	dummy cycle count	0x0
20	byte4en	R/W	4 bytes address cycle enable in dmmr_mode	0x0
21	byte4cmd	R/W	4 bytes address cmd enable in dmmr_mode	0x0
31:22	Reserved			

TRAN_NUM

Offset Address: 0x014

Bits	Name	Access	Description	Reset
15:0	rdat_tran_num	R/W	When not under dmmr_mode, TRAN_NUM is the number of frames sent and received in a transmission	0x0
31:16	Reserved			

FF_PORT

Offset Address: 0x018

Bits	Name	Access	Description	Reset
31:0	rdat_ff_port	R/W	FIFO write read address	0x0

FF_PT

Offset Address: 0x020

Bits	Name	Access	Description	Reset
3:0	rdat_ff_pt	R/W	Read as the number of effective data bytes in FIFO, write as FIFO clear	0x0
7:4	Reserved			
9:8	wrcnt	R/W	Current FIFO, write byte offset indicator status	0x0
12:10	rdpt	R/W	Current FIFO, read byte offset indicator status	0x0
31:13	Reserved			

INT_STS

Offset Address: 0x028

Bits	Name	Access	Description	Reset

Bits	Name	Access	Description	Reset
0	tran_done_int	R/W	The interrupt is generated once every successful transmission of a frame of data	0x0
1	Reserved			
2	rdff_int	R/W	The interrupt is generated once for each successful received frame.	0x0
3	wrff_int	R/W	After receiving the interrupt, CPU writes frame data to FIFO.	0x0
4	rx_frame_int	R/W	The CPU reads frame data from FIFO after receiving the interrupt .	0x0
5	tx_frame_int	R/W	This interrupt marks the completion of a transmission.	0x0
31:6	Reserved			

INT_EN

Offset Address: 0x02c

Bits	Name	Access	Description	Reset
0	tran_done_int_en	R/W	Enable interrupt tran_done_int	0x0
1	Reserved			
2	rdff_int_en	R/W	Enable interrupt rdff_int	0x0
3	wrff_int_en	R/W	Enable interrupt wrff_int	0x0
4	rx_frame_int_en	R/W	Enable interrupt rx_frame_int	0x0
5	tx_frame_int_en	R/W	Enable interrupt tx_frame_int	0x0
31:6	Reserved			

4.3 SPI NAND Flash Controller

4.3.1 Overview

Support external SPI NAND flash data access.

4.3.2 Characteristics

One chip selection pin.

Support SPI NAND Flash X1 / x2 / X4 read / write operation.

Support multiple specifications of SPI NAND flash device.

2KB and 4KB page size.

64 pages/block and 128 pages/block devices.

Support the BOOT function of SPI NAND.

4.3.3 Function Description

4.3.3.1 Interface Description

SPI NAND flash controller supports three SPI NAND interface types, Standard SPI, X2 interface mode and X4 interface mode.

Standard SPI Interface Mode:
Standard SPI interface mode has 1 bit data input and 1 bit data output. Write operation timing sequence of standard SPI interface mode is shown in Figure 4- 8 Standard SPI Write Operation Timing
. Read operation timing sequence of standard SPI interface mode is shown in Figure 4- 9 Standard SPI Read Operation Timing
Sequence description：

- command/address/dummy cycles are output on DO line in single bit serial mode.
- Data is inptut from DI in single bit serial mode.

.

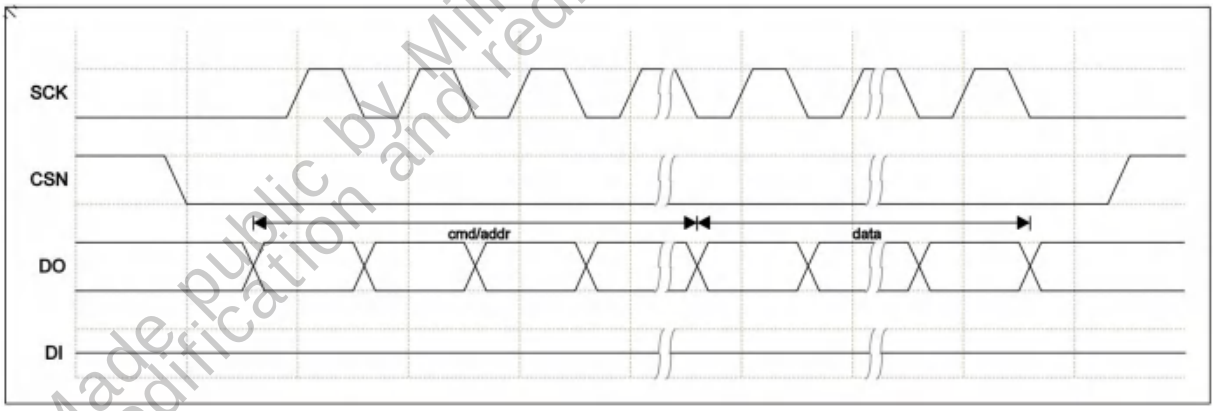


Figure 4- 8 Standard SPI Write Operation Timing

Sequence description:

- command/address/dummy cycles are output on DO in single bit serial mode.
- Data is output on DO in single bit serial mode.

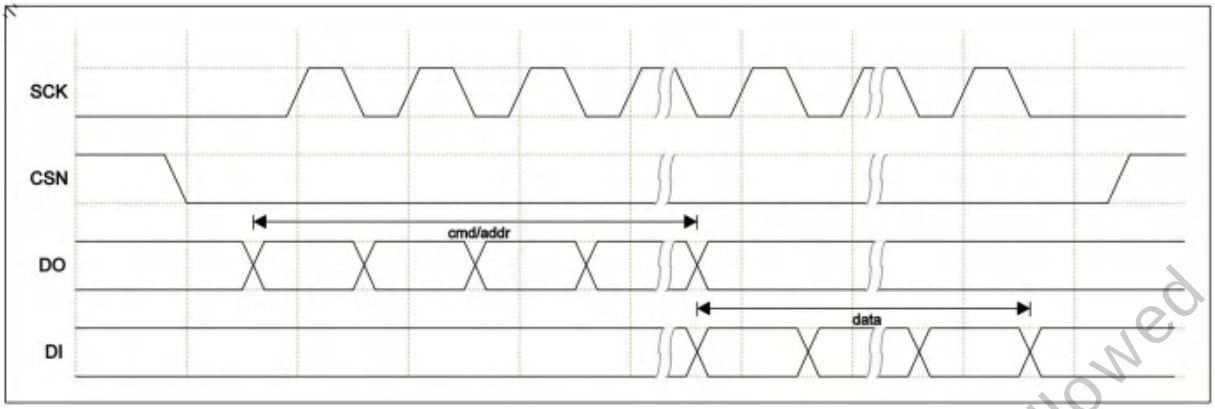


*Figure 4- 9 Standard SPI **Read Operation Timing***

Sequence description:

- command/address/dummy cycles are output on DO line in single bit serial mode.
- Data is inptut from DI in single bit serial mode.

X2 Interface Mode:

Data input and output use two common I/O pins in X2 interface mode. The operation timing sequence is shown in Figure 4- 10.

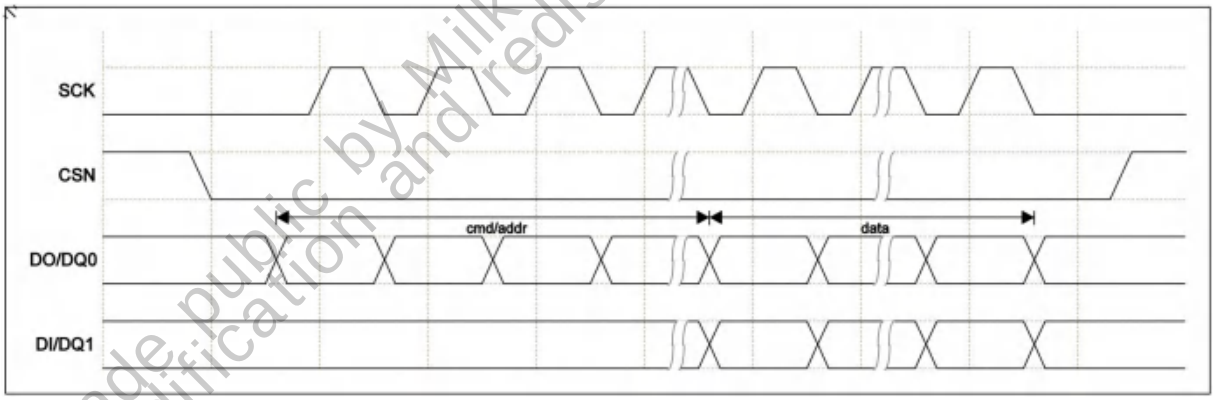


Figure 4- 10 SPI Nand x2 Interface Mode Operation Sequence

Sequence description:

- command/address/dummy cycles are output on DO in single bit serial mode.
- Data output (write) or input (read) on DO / DI line in 2 bits mode.

X4 Interface Mode:

Data input and output use four common I/O pins in X4 interface mode. The operation timing sequence is shown in Figure 4- 11.

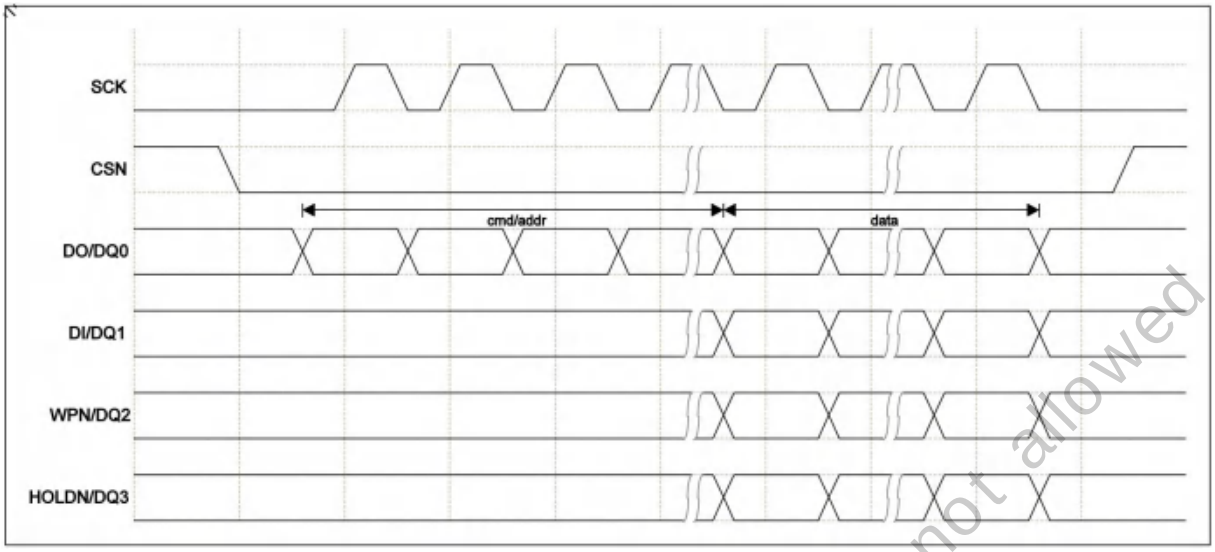


Figure 4- 11 SPI Nand x4 Interface Mode Operation Sequence

Sequence description:

- command/address/dummy cycles are output on DO in single bit serial mode.
- Data output (write) or input (read) on DO / DI / WPN / HOLDN in 4 bits mode.

4.3.3.2 SPI NAND FLASH Address Description

When issuing the read-write operation of SPI NAND flash, the column address is issued according to the specific operation.

Write operation: configure column address in PROGRAM LOAD operation and row address in PROGRAM EXCUTE operation.

Read operation: configure row address in PAGE READ TO CACHE operation and column address in PAGE READ operation.

The address distribution is completed by the controller. The software needs to configure reg_ trx_ cmd_ idx according to the operation instructions, and address configures reg_ trx_ cmd_ cnt0 and reg_ trx_ cmd_ cnt1.

4.3.3.3 Boot Function

Because the SPI NAND flash address space is discontinuous and there is the possibility of bad blocks, boot data can not be directly mapped to flash.

It supports adaptive boot function, and can automatically update the adapter information according to the data of block0. For controller boot operation requirement that physical Block0 must be a good block, and other blocks can be automatically skipped if they are bad blocks.

4.3.3.4 Register Opreation

The software configures the operation related register, such as operation command, address, etc., and set reg_ trx_start register to issue the operation. The controller sends the operation to the device according to the software configuration value. If there are data content to transfer to the device, the internal DMA will be used.

4.3.3.5 Built-in DMA Operation Method

Support built-in system DMA mode to improve read/write operations speed. In this way, access internal or external memory space by dma bus directly.
Step 1. Configure DMA channel.
Step 2. Configure the source and destination address.
Step 3. Configure the transmission format and data length.

4.3.3.6 TIMEOUT Function

The maximum one second timeout could be set by software. The timout mechanism used to protect system if device does not respond normally.

4.3.4 Operation Flow

4.3.4.1 Initialization Process

Step 1. (if the timing parameter needs to be adjusted) Based on device timing to configure time sequence register reg_trx_sck_h and reg_ trx_ sck_ l.
Step 2. Configure the interrupt control register reg_ trx_ done_ int_ en.

4.3.4.2 Device Register Operation Process

Step 1. Configure the transmission content length reg_ trx_ cmd_ cont_ size and reg_ trx_ data_ size.
Step 2. Configure the device instruction and its related contents reg_ trx_ cmd_ id, reg_ trx_ cmd_ cont0 and reg_ trx_ cmd_ cont1.
Step 3. Set reg_ trx_ start register to issue operation.
Step 4. When reg _ trx_ done_ int is asserted, it indicates that the operation is completed.

4.3.4.3 Erase Operation Process

Flash must be erased before programming, and WREN should be set before erase operation.
Step 1. Configure the transmission content length reg_ trx_ cmd_ cont_ size.
Step 2. Configure the device instruction and its related contents reg_ trx_ cmd_ id and reg_ trx_ cmd_ cont0.
Step 3. Set reg_ trx_ start register to issue operation.
Step 4. When reg _ trx_ done_ int is asserted , it indicates that the operation is completed.

4.3.4.4 Built-in DMA Read Operation Process

Step 1. Configure the system DMA register, and refer to 错误! 未找到引用源。.
Step 2. Configure the transmission content length reg_ trx_ cmd_ cont_ size and reg_ trx_ data_ size.
Step 3. Configure the device instruction and its related contents reg_ trx_ cmd_ id, reg_ trx_ cmd_ cont0 and reg_ trx_ cmd_ cont1.
Step 4. Configure reg_ trx_ start register to issue operation.
Step 5. . When reg _ trx_ done_ int is asserted, it indicates that the operation is completed and data is written to memory buffer.

4.3.4.5 Built-in DMA Write Operation Process

Step 1. Configure the system dmaregister, and refer to 错误! 未找到引用源。.
Step 2. Configure the transmission content length reg_ trx_ cmd_ cont_ size and reg_ trx_ data_ size.
Step 3. Configure the device instruction and its related contents reg_ trx_ cmd_ id, reg_ trx_ cmd_ cont0 and reg_ trx_ cmd_ cont1.
Step 4. Set reg_ trx_ start register to issue operation.
Step 5. When reg _ trx_ done_ int is asserted, the buffer content has been written to the device cache.

4.3.4.6 Other Reminders

RESET command is required before normal access or after abnormal reset in some SPI NAND device. Therefore, for better device compatibility, the first transmission instruction after device power up or abnormal reset should be RESET command. Before the device operation is completed, do not change the relevant register configuration, otherwise the operation may be abnormal.

4.3.5 Data Structure (NAND Flash/SPI NAND Flash)

4.3.5.1 2KB page_size

For common 2KB page_size device, the available spare area for software is 64 Bytes. The actual size of the spare area is depended on the device page structure. Data structure of buffer and flash page is shown as follow.

User Data	Data(2048)	OOB(64)

4.3.5.2 4KB page_size

For common 4KB page_size device, the available spare area for software is 256 Bytes. The actual size of the spare area is depended on the device page structure. Data structure of buffer and flash page is shown as follow.

User Data	Data(4096)	OOB(256)

4.3.6 Register Overview

Name	Address Offset	Description
reg_ctrl	0x000	transmission control
reg_timing_ctrl	0x004	timing control
reg_trx_size	0x008	number of content size
reg_int_en	0x010	interrupt enable
reg_int_clr	0x014	interrupt clear
reg_int_sts	0x01c	interrupt status
reg_cont0	0x030	content 0
reg_cont1	0x034	content 1
reg_cmplt_cnt	0x058	number of transferred bytes
reg_tx_data	0x060	tx data
reg_rx_data	0x064	rx data

4.3.7 Register description

reg_ctrl

Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	reg_trx_start	W1T	trigger spi transmission start	
31:1	Reserved			

reg_timing_ctrl

Offset Address: 0x004

Bits	Name	Access	Description	Reset
1:0	reg_trx_time_start	R/W	time for cs assert to 1st command bit unit: sck period	0x0
3:2	Reserved			
7:4	reg_trx_time_end	R/W	time for last data bit to cs de-assert unit: sck period	0x0
15:8	Reserved			
19:16	reg_trx_sck_h	R/W	time for sck high unit: source clock period	0x0
23:20	reg_trx_sck_l	R/W	time for sck low unit: source clock period	0x1
31:24	Reserved			

reg_trx_size

Offset Address: 0x008

Bits	Name	Access	Description	Reset
2:0	reg_trx_cmd_cont_size	R/W	numbers of command content byte	0x0
3	Reserved			

Bits	Name	Access	Description	Reset
5:4	reg_trx_dummy_size	R/W	numbers of dummy byte	0x0
15:6	Reserved			
28:16	reg_trx_data_size	R/W	numbers of data byte	0x0
31:29	Reserved			

reg_int_en

Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	reg_trx_done_int_en	R/W	trx_done interrupt enable	0x1
31:1	Reserved			

reg_int_clr

Offset Address: 0x014

Bits	Name	Access	Description	Reset
0	reg_trx_done_int_clr	W1T	trx_done interrupt clear	
31:1	Reserved			

reg_int_sts

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
0	reg_trx_done_int	RO	trx_done interrpt	
31:1	Reserved			

reg_cont0

Offset Address: 0x030

Bits	Name	Access	Description	Reset
7:0	reg_trx_cmd_idx	R/W	spi flash command value	0x0
31:8	reg_trx_cmd_cont0	R/W	spi flash address, or other contents	0x0

reg_cont1

Offset Address: 0x034

Bits	Name	Access	Description	Reset
31:0	reg_trx_cmd_cont1	R/W	spi flash address, or other contents	0x0

reg_cmplt_cnt

Offset Address: 0x058

Bits	Name	Access	Description	Reset
12:0	reg_cmplt_cnt	RO	number of transferred bytes	
31:13	Reserved			

reg_tx_data

Offset Address: 0x060

Bits	Name	Access	Description	Reset
31:0	reg_tx_data	RO	spi tx data	

reg_rx_data

Offset Address: 0x064

Bits	Name	Access	Description	Reset
31:0	reg_rx_data	RO	spi rx data	

5 Network interface

5.1 Ethernet MAC

5.1.1 Overview

The core chip supports two Ethernet Macs to receive and transmit network data.
An Ethernet MAC with built-in 10 / 100Mbps Fast Ethernet transmitter can work in 10 / 100Mbps full duplex or half duplex mode..
.

5.1.2 Function description

The Ethernet module has the following features:

1. Ethernet MAC0 is paired with a built-in 10/100Mbps Fast Ethernet Transceiver and a built-in Ethernet PHY to support 10/100Mbit/s rates.
2. Supports full-duplex or half-duplex operation modes.

- Supports CRC verification of input frames.
- Supports adding CRC verification to output frames.
- Supports short frame padding.
- Supports internal loopback in full-duplex mode.
- Supports counting of received and transmitted frames.
- Supports receive and transmit packet buffer.
- Supports COE (Checksum Offload Engine) checksum offload engine functionality.

5.1.3 Data flow overview

The conceptual data stream of Ethernet switching interface is shown in Figure 51.

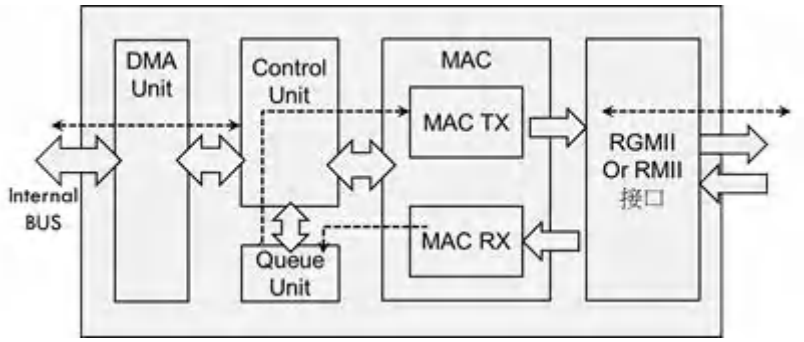


Chart 5- 1 GMAC conceptual data stream

5.1.4 Single port function configuration description

5.1.5 Ethernet transceiver frame management function

CPU first configures Ethernet MAC receive and transmit descriptor list buffer and content composition of descriptor list, for example, setting of frame address and packet type size parameters

During receiving, Ethernet MAC receives all kinds of packets, and according to the CPU configuration, it receives descriptor list messages, such as packet cache information, including packet cache starting address, packet cache depth, etc., and stores the received packets in DDR. And then inform the CPU to do the follow-up processing.

During transmitting, according to the packet cache information of sending descriptor list configured by CPU, such as packet cache starting address, packet length and other packet information, Ethernet MAC carries the packets stored in DDR, assembles them into packets, and then sends them to the network interface. Then inform the CPU that the packet has been sent.

5.1.6 Ethernet packet receiving interrupt management function

5.1.6.1.1 Interrupt generation

Set the receive direction interrupt and configure Re_Int_Enable bit[6] = 1, CPU queries receive interrupt status Reg_Int_Status bit[6].

5.1.6.1.2 Interrupt clearing

CPU query receive interrupt status Reg_Int_Status bit[6], write 1 to clear the interrupt status.

5.1.7 Configure the working state of PHY chip

Ethernet MAC provides MDIO interface to config PHY chip. MDIO interface is divided into read operation and write operation. The main register controlling MDIO interface is Reg_MdioAddr and Reg_MdioData.

The configuration steps of read operation are as follows：

Configure the MDIO control register with the following settings:

Reg_MdioAddr bit [15:11] config the PHY chip address. Please config based on plan according to PHY chip or version

Reg_MdioAddr bit [10:6] sets the PHY internal register address to read and write.

Reg_MdioAddr bit [1] writes 0 (read action command).

Finally, Set Reg_MdioAddr bit [0] = 1 to start reading.

The MDIO interface will receive the read back data to Reg_MdioData bit [15:0], and change Reg_MdioData to 0.

The configuration steps of write operation are as follows：

Configure the MDIO control register with the following settings:

Reg_MdioData bit [15:11] sets the PHY chip address of clause45 mode

Reg_MdioData bit [10:6] sets the value written to PHY chip

Reg_MdioAddr bit [1] writes 1 (write action command).

Finally, Set Reg_MdioAddr bit [0] = 1 to start writing:

The MDIO interface will Reg_Mdi0Addr bit [0] will change to 0 after writing.

5.1.8 Working mode switching

Working mode of Ethernet MAC：

Ethernet MAC0 supports the use of built-in ephy function. Its working mode is RMII (10 / 100M)

The speed and mode switch register is set as follows:

Configure ETH0 Reg_MacConfig bit[14] = (100M:1, 10M:0) ;

Note: this configuration is not allowed when the chip is working normally. It is recommended to configure it during initialization.

5.1.9 Typical applications

5.1.10 Register offset address description

Ethernet MAC 0/1 Register offset address space:

ETH0_MAC : 0x0451_000~0x0451_FFFF

5.1.11 GMAC register overview

Table 5- 1 GMAC register overview

Name	Address Offset	Description
Reg_MacConfig	0x000	Native MAC operational status register
Reg_MdioAddr	0x010	MDIO operation register
Reg_MdioData	0x014	MDIO data read/write register
Reg_MacAddr0_High	0x040	Native MAC address register #0 high 16 bits
Reg_MacAddr0_Low	*0x044*	*Native MAC address register #0 low 32 bits*
Reg_MacAddr1_High	0x048	Native MAC address register #1 high 16 bits
Reg_MacAddr1_Low	0x04c	Native MAC address register #1 low 32 bits
Reg_Tx_Packet_Num_Good_Bad	0x118	Transmit good frame and bad frame count register
Reg_Tx_Bcast_Packets_Good	0x11c	Transmit good broadcast frame count register
Reg_Tx_Mcast_Packets_Good	0x120	Transmit good multicast frame count register
Reg_Tx_Ucast_Packets_Good_Bad	0x13c	Transmit good unicast frame and bad frame count register
Reg_Tx_Mcast_Packets_Good_Bad	0x140	Transmit good multicast frame and bad frame count register
Reg_Tx_Bcast_Packets_Good_Bad	0x144	Transmit good broadcast frame and bad frame count register
Reg_Rx_Packets_Num_Good_B	0x180	Receive good frame and bad frame count register

Name	Address Offset	Description
ad		
Reg_Rx_Bcast_Packets_Good	0x18c	Receiver successful good broadcast frames count register
Reg_Rx_Mcast_Packets_Good	0x190	Receiver successful good multicast frames count register
Reg_Rx_CRC_Error_Packets	0x194	Receiver CRC error frames count register
Reg_Rx_Ucast_Packets_Good	0x1c4	Receiver successful good unicast frames count register
Reg_Int_Enable	0x101c	Interrupt enable register
Reg_Int_Status	0x1014	Interrupt status register

5.1.12 GMAC register description

Reg_MacConfig

Offset Address: 0x000

Bits	Name	Access	Description	Reset
1:0	Reserved			
2	RX_EN	R/W	MAC Receive Enable RegisterMAC	0x0
3	TX_EN	R/W	MAC Transmit Enable RegisterMAC	0x0
6:4	Reserved			
7	APCS_EN	R/W	Automatic Pad or CRC Stripping Control Enable Register	0x0
9:8	Reserved			
10	CHKS_EN	R/W	IP Checksum Offload Enable Register	0x0
11	DUPLEX_MODE	R/W	Full/Half Duplex Mode Register, 1 (Enables Full Duplex Mode)	0x0
12	LPBK_MODE	R/W	Loopback Mode Control Register	0x0
13	Reserved			
14	SPEED_MODE	R/W	Speed Mode Register 1'b1: 100M, 1'b0: 10M	0x0
16:15	Reserved			
19:17	IPG_VAL	R/W	Transmit Packet Spacing Control Register	0x0
22:20	Reserved			
23	WD_DISABLE	R/W	Watchdog Disable Register	0x0
24	Reserved			
25	CRC_STRIP_EN	R/W	CRC Stripping Type Packet Control Enable Register	0x0
31:26	Reserved			

Reg_MdioAddr

Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	GO	R/W	MDIO operation completion indication 1: Start operation 0: operation completed	0x0
1	CMD	R/W	MDIO operation command type Write (1'b1), Read (1'b0)	0x0
5:2	Reserved			
10:6	RegAddr	R/W	External PHY address configuration register	0x0
15:11	PhyAddr	R/W	PHY device internal register address register	0x0
31:16	Reserved			

Reg_MdioData

Offset Address: 0x014

Bits	Name	Access	Description	Reset
15:0	MdioData	R/W	MDIO writes or reads back the data register from PHY	0x0
31:16	RegAddrC45	R/W		0x0

Reg_MacAddr0_High

Offset Address: 0x040

Bits	Name	Access	Description	Reset
15:0	Addr_High	R/W	MAC address register #0 bit[47:32]	0x0
30:16	Reserved			
31	AddrEN	R/W	Address Enable	0x0

Reg_MacAddr0_Low

Offset Address: 0x044

Bits	Name	Access	Description	Reset
31:0	Addr_Low	R/W	MAC address register #0 bit[31:0]	0x0

Reg_MacAddr1_High

Offset Address: 0x048

Bits	Name	Access	Description	Reset
15:0	Addr1_High	R/W	MAC address register #1 bit[47:32]	0x0
23:16	Reserved			
29:24	Addr1_MASK	R/W	Addr1 Mask Byte	0x0
30	Addr1_TYPE	R/W	1'b1: compare SA 1'b0: compare DA	0x0
31	Addr1_EN	R/W	Addr1 Enable	0x0

Reg_MacAddr1_Low

Offset Address: 0x04c

Bits	Name	Access	Description	Reset
31:0	Addr1_Low	R/W	MAC address register #1 bit[31:0]	0x0

Reg_Tx_Byte_Num_Good_Bad

Offset Address: 0x118

Bits	Name	Access	Description	Reset
31:0	TxByteNumGB	RO	Successfully sent good packet and bad packet byte count register	

Reg_Tx_Bcast_Packets_Good

Offset Address: 0x11c

Bits	Name	Access	Description	Reset
31:0	TxBcG	RO	Successfully sent broadcast frame count register of good packets	

Reg_Tx_Mcast_Packets_Good

Offset Address: 0x120

Bits	Name	Access	Description	Reset
31:0	TxMcG	RO	Successfully sent multicast frame count register of good packets	

Reg_Tx_Ucast_Packets_Good_Bad

Offset Address: 0x13c

Bits	Name	Access	Description	Reset
31:0	TxUcGB	RO	Successfully sent unicast frame count register of good packets and wrong packets	

Reg_Tx_Mcast_Packets_Good_Bad

Offset Address: 0x140

Bits	Name	Access	Description	Reset
31:0	TxMcGB	RO	Successfully sent multicast frame count register for good packets and bad packets.	

Reg_Tx_Bcast_Packets_Good_Bad

Offset Address: 0x144

Bits	Name	Access	Description	Reset
31:0	TxBcGB	RO	Successfully sent statistics register of the number of broadcast frames of good packets and wrong packets	

Reg_Rx_Packets_Num_Good_Bad

Offset Address: 0x180

Bits	Name	Access	Description	Reset
31:0	RxPktGB	RO	Successfullly received statistics register of frame number of good packets and wrong packets.	

Reg_Rx_Bcast_Packets_Good

Offset Address: 0x18c

Bits	Name	Access	Description	Reset
31:0	RxBcG	RO	Receive successful good packet broadcast frame count register.	

Reg_Rx_Mcast_Packets_Good

Offset Address: 0x190

Bits	Name	Access	Description	Reset
31:0	RxMcG	RO	Received good packets' multicast frame count statistics register.	

Reg_Rx_CRC_Error_Packets

Offset Address: 0x194

Bits	Name	Access	Description	Reset
31:0	RxCrcERR	RO	Receiver CRC error frame count register.	

Reg_Rx_Ucast_Packets_Good

Offset Address: 0x1c4

Bits	Name	Access	Description	Reset
31:0	RxUcG	RO	Counter register for the number of successfully received unicast frames.	

Reg_Int_Enable

Offset Address: 0x101c

Bits	Name	Access	Description	Reset
0	TxInt_EN0	R/W	Send interrupt enable register	0x0
5:1	Reserved			
6	RxInt_EN0	R/W	Receive interrupt enable register	0x0
31:7	Reserved			

Reg_Int_Status

Offset Address: 0x1014

Bits	Name	Access	Description	Reset
0	TxInt_ST0	RO	Send interrupt state register	
5:1	Reserved			
6	RxInt_ST0	RO	Receive interrupt state register	
31:7	Reserved			

5.2 Ethernet PHY

5.2.1 Overview

The chip provides a set of built-in Ethernet 10/100 Base-TX compliant PHY interfaces.

5.2.2 Function description

Support IEEE 802.3 10/100 Base-TX compliant.
Support full duplex and half duplex and auto-negotiation function.
Support Auto-MDIX function to auto detect crossover cable or straight-through cable and flip the TX and RX accordingly.
Support WOL (Wake on LAN) over Ethernet.

5.2.3 Functional block diagram

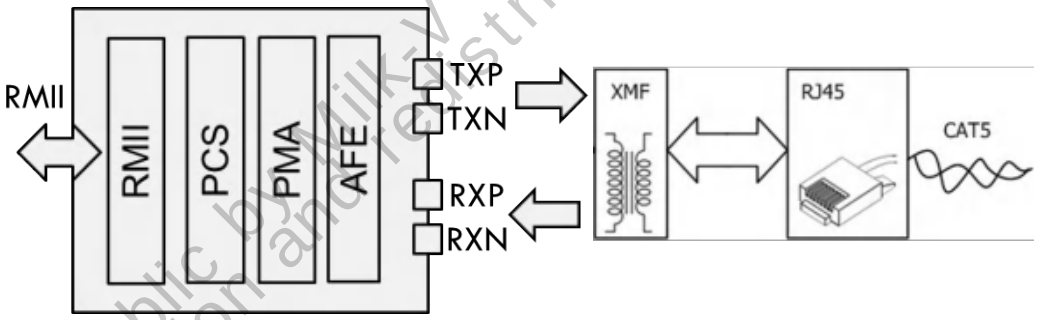


Chart 5- 2 built-in 10/100 Ethernet PHY chart

The 10 / 100Mbps transmission and receiving is on the standard category 5(CAT5) twisted pair table. The transmission and receiving signals are connected to the RJ45 standard interface through a transformer.

6 Video and image codec

6.1 Overalll overview

The video and image codec hardware unit integrates video codec unit (VCU) and JPEG codec unit (JCU). VCU supports H.265/HEVC, H.264/AVC international standards, while JCU supports JPEG and Motion-JPEG. VCU/JPU provides real-time, high performance, low delay, low power consumption, small bus bandwidth and CPU utilization.

6.2 VCU (Video Codec Unit)

6.2.1 Overview

VCU (Video Codec Unit) includes two functions: video encoding (VENC) and video decoding (VDEC). With software control, it can encode and decode simultaneously according to application requirements.

6.2.2 Features

VENC module supports the following features:

- Support ITU-T H.265/HEVC Main Profile @Level 4 Main Tier coding
- Support I and P frames
- Support 1/2、1/4 pixel precision motion compensation
- Support CTU 64 coding unit
- Inter prediction supports 32x32, 16X16, 8x8 and other PU types
- Inter prediction supports Merge/Skip mode
- Intra prediction supports 32x32, 16X16, 8x8 and other PU types
- Support TU types such as 32x32, 16X16, 8x8 and 4X4
- Support CABAC entropy coding
- Support De-blocking filter

Support QPMap
Support H.265 HSVC time domain layering (HSVC-T)
Support ITU-T H.264/AVC High Profile/Main Profile/Constrained Baseline Profile@Level 4.2 coding
Support I and P frames
Support 1/2、1/4 pixel precision motion compensation
Inter frame prediction supports PU types such as 16x16, 16x8, 8x16 and 8x8
Intra prediction supports 16x16, 8x8, 4x4 and other PU types
Support TU types such as 8x8 and 4X4
Support CABAC、CAVLC entropy coding
Support De-blocking filter
Support QPMap
Support H.264 SVC time domain layering (SVC-T)
The following input image formats are supported
Planar YCbCr4:2:0
H.265/H.264 multi-stream encoding performance：
2880x1620@20ps+720x576@20ps encoding
It supports configurable image resolution
Minimum image resolution：256x256
Maximum image resolution：2880x1620
Support Region Of Interest coding (ROI)
Region of interest coding for up to 8 regions is supported
Support CBR/VBR/FIXQP/QPMAP rate control mode
Support slice output interrupt
Output code rate H.265 up to 25Mbps
Output bit rate H.264 up to 50Mbps

6.2.3 Function description

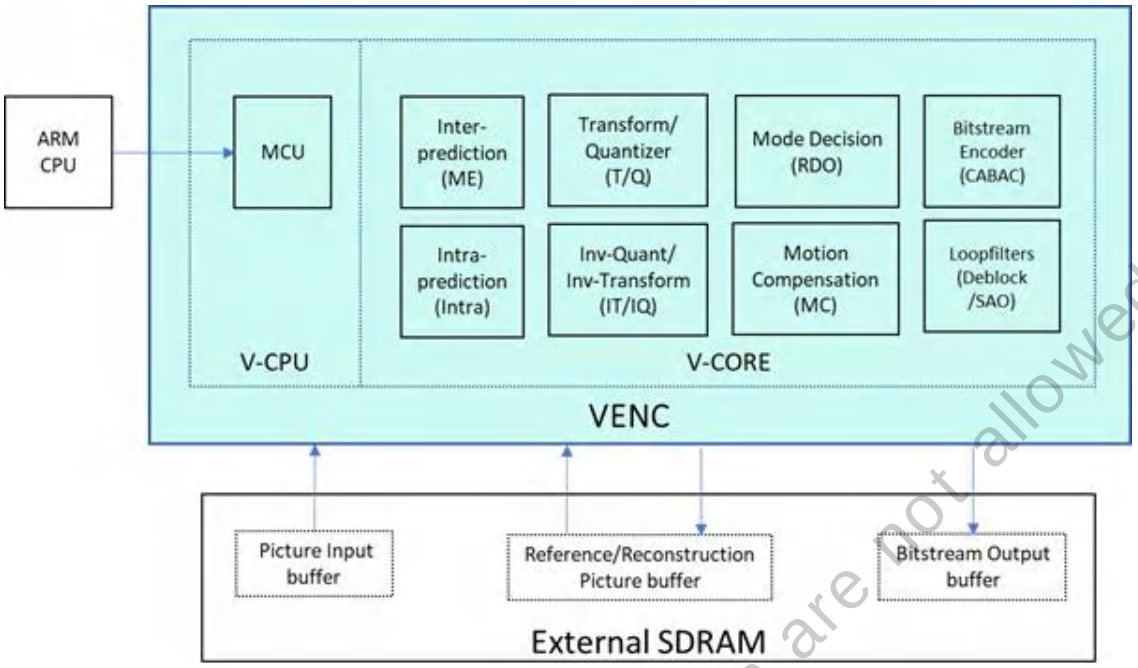


Figure 6- 1 ***VENC function block diagram***

The functional block diagram of VENC is shown in Figure 6-1. The function of VENC module is divided into two groups: V-CPU and V-CORE. The main V-CORE group implements motion estimation (ME) / Inter-prediction, Intra-prediction, Transform / Quantizer, Inv-Quant/Inv-Transform, Mode Decision, Motion Compensation, entropy coding and stream generation, deblocking filtering and SAO (only supporting h.265) And so on. The V-CPU group is mainly built with a MCU (micro control unit) and its required on-chip memory, which receives commands from the upper ARM software and controls the generation of bitstream in V-CORE. ARM CPU software completes coding control processing such as rate control and interrupt processing.

As shown in the figure, before starting VENC for video encoding, CPU software needs to allocate the following three types of buffers in external SDRAM.

Picture Input buffer

Before encoding, the original image to be encoded is usually written into the buffer by the video input unit or the video processing unit. During the encoding process, VENC will read the image from this image input buffer and start encoding.

Reference/Reconstruction Picture buffer

In the process of encoding, VENC will write the reconstructed image to this buffer as the reference image of the original image to be encoded. When P-frame encoding, the reference image will be read from this buffer and compared with the original image to construct the best motion vector.

Bitstream Output buffer

During the encoding process, VENC will write the encoded bistream to this buffer. This buffer is usually read by the software and packaged in the next stage

6.3 JCU (JPEG Codec Unit)

6.3.1 Overview

JCU (JPEG Codec Unit) contains JPEG image or Motion-JPEG video encoding (JPE) and decoding (JPD) two major functions. As same as VCU, through software control, according to the application requirements, JPU can simultaneously do encoding and decoding.

6.3.2 Features

JPE module supports the following features:

Support ISO/IEC 10918-1 Baseline Profile JPEG coding

- Support image coding with YCbCr4:0:0, YCbCr4:2:0, YCbCr4:2:2, YCbCr4:4:4 chroma sampling format
- Support up to 3 quantization tables
- Support 8-bit sampling accuracy.

Encoding performance

- minimum image resolution : 16x16
- maximum image resolution : 2880x1620
- Motion-JPEG encoding up to 5M(2880x1620, YCbCr4:2:0)@20fps
- Motion-JPEG output rate range：20Kbps ~ 200Mbps

JPD module supports the following features:

Support ISO/IEC 10918-1 Baseline Profile JPEG decoding：

- Support image decoding of YCbCr4:0:0, YCbCr4:2:0, YCbCr4:2:2, YCbCr4:4:4 chroma sampling formats
- Support up to 3 quantization tables
- Support 8-bit sampling accuracy

Decoding performance

- minimum image resolution：16x16
- maximum image resolution：2880x1620

- Motion-JPEG decoding up to 5M(2880x1620, YCbCr4:2:0)@20fps

6.3.3 Function description

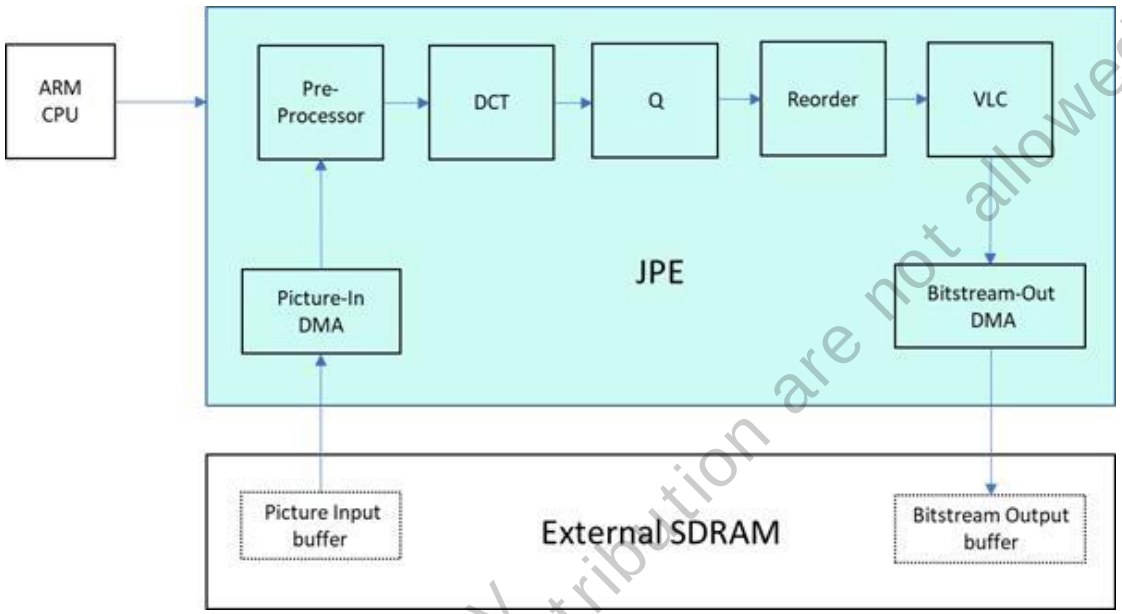


Figure 6- 2 JPE function block diagram

The functional block diagram of JPE is shown in Figure 6-2. JPE hardware implements the processing of image input DMA, level shift, DCT (Discrete Cosine Transform), Quantization, Reorder, VLC coding and bitstream output DMA, while CPU software completes the coding control processing of quantization table configuration and interrupt processing.

As shown in the figure, before starting JPE for encoding, CPU software needs to allocate the following two types of buffers in external SDRAM：

Picture Input buffer

Before encoding, the original image to be encoded is usually written into the buffer by the video input unit or the video processing unit. During the encoding process, JPE will read the image from this image input buffer, and then start encoding.

Bitstream output buffer

During the encoding process, JPE will write the encoded bitstream to this buffer. This buffer is usually read by the software and packaged in the next stage.

。

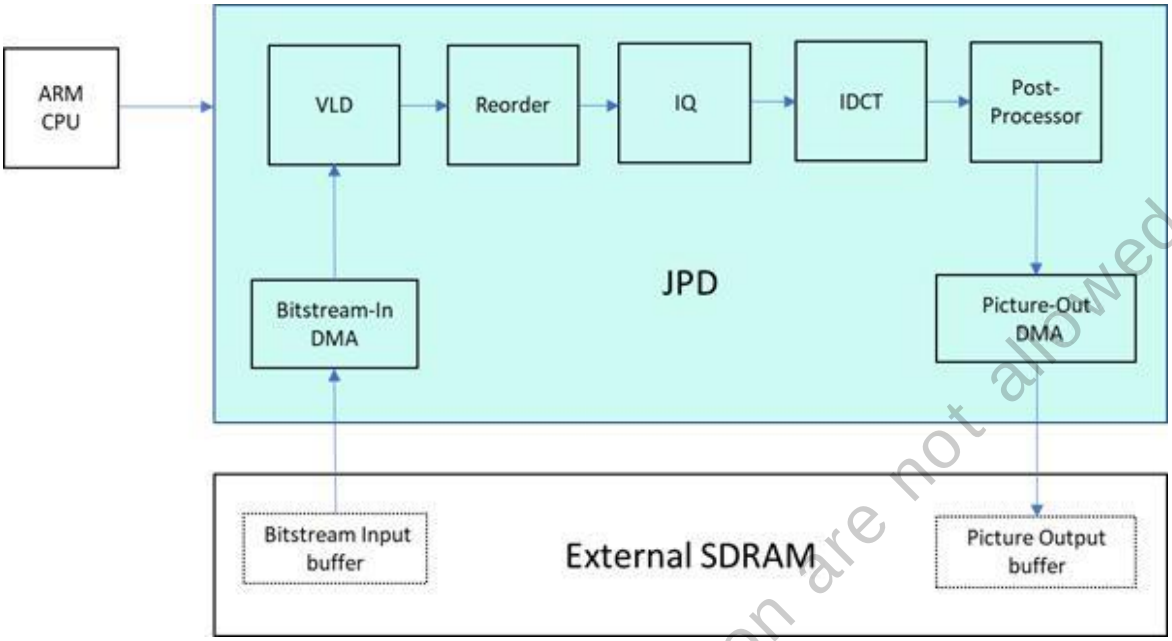


Figure 6- 3 JPD function block diagram

As shown in the figure, JPD function block diagram is shown in Figure 6-3. JPD hardware implements the processing with large amount of computation, such as bitstream input DMA, VLD decoding, Reorder, Inverse Quantization, IDCT, post processor, level shift and image output DMA, while ARM CPU software completes decoding control processing, such as package header decoding and interrupt processing.

As shown in the figure, before starting JPD for decoding, the ARM CPU software needs to allocate the following two types of buffers in the External SDRAM：

Bitstream input buffer

Usually, the bitstream to be decoded is written into this buffer by software before decoding, and then read by JPD during decoding.

Picture output buffer

JPD will write the decoded image into this buffer during decoding.

7 Video and graphics processing

7.1 VPSS (sc_top)

7.1.1 Overview

The Video Processing Subsystem (VPSS) implements video processing functions and supports both online modes (ISP-VPSS online and ISP-VPSS-VC fully online) and offline modes. It includes video masking, privacy masking, video cropping, scaling, mirror, flip, 180-degree rotation, LBA amplitude ratio conversion, circular masking, OSD overlay, and multi-area stitching.

7.1.2 Function description

The characteristics of VPSs are as follows:

When the output width is less than 2880, the video source with maximum input width of 4096 is supported

Support video source with maximum input width of 2880 when the output width is above 2880

Support up to 3 channels of video output

3-channel video output can be configured independently

The input data is 420/422/single component/RGB planer/ Packet RGB/NV12/NV21/422-packet/420 semi-planar

The output data is 420/422/ component /RGB planer/ Packet RGB/ NV12/NV21/422-packet/420 semi-planar/HSV/BF16

Support video clipping

Support mirror / flip / 180 degree rotation

Support low delay mode of output channel

LBA format ratio conversion: adjust the image format ratio by adding a fixed background color border to the upper and lower edges or left and right edges of the image

Support circular occlusion function

Supports OSD and video overlay in 8 regions, and uses this feature to achieve video masking.
Provides privacy masking function (pixel-based, grid 8x8, RGB332 format).
OSD input format: ARGB8888/ARGB4444/ARGB1555/8-bit LUT/4-bit LUT/bit-font.
OSD font supports color inversion with background brightness.
Supports OSD compression/decompression to save memory space.
Scaling factor supports 1/32-32 times.

7.2 LDC (Lens Distortion Correction)

7.2.1 Overview

Lens Distortion Correction (LDC) corrects lens distortion and rotates (90/270) a frame of image. It mainly consists of two functions: geometric distortion correction and affine transformation.

7.2.2 Function description

- LDC features are as follows:
  - Lens Distortion Correction
    - Supports video sources with a maximum input width of 4096 and a maximum output width of 4096.
    - Supports output to Dynamic Random-Access Memory (DRAM).
    - When outputting to DRAM, supports NV12/NV21 and 8-bit input/output.
    - When outputting to DRAM, supports single Y 1-plane and 8-bit input/output.
    - Supports 90/270-degree rotation.
    - Supports correction of barrel and pincushion distortion.
      - Supports up to 20% correction of barrel distortion.
      - Supports up to 20% correction of pincushion distortion.
      - Supports barrel expansion function.
    - Maximum output performance of 240 megapixels per second.

- Affine Transformation
  - Supports video sources with a maximum input width of 4096 and a maximum output width of 4096.
  - Affine transformation only supports output to DRAM.
  - Supports single Y-plane, N21/NV12, and 8-bit input/output.
  - Maximum output performance of 240 megapixels per second.

8 AI engine

8.1 TPU (Tensor Processing Unit)

8.1.1 Overview

TPU is an AI acceleration engine for deep learning neural network, which can be used to accelerate image classification, object detection, face detection and recognition, segmenataion, and LSTM, etc.. Figure 8-1 shows the block diagram between TPU and CPU on the chip. The main function of TPU is to offload CPU work and to accelerate Computer Vision and Speech related operations. The two blocks communicate through interruption.

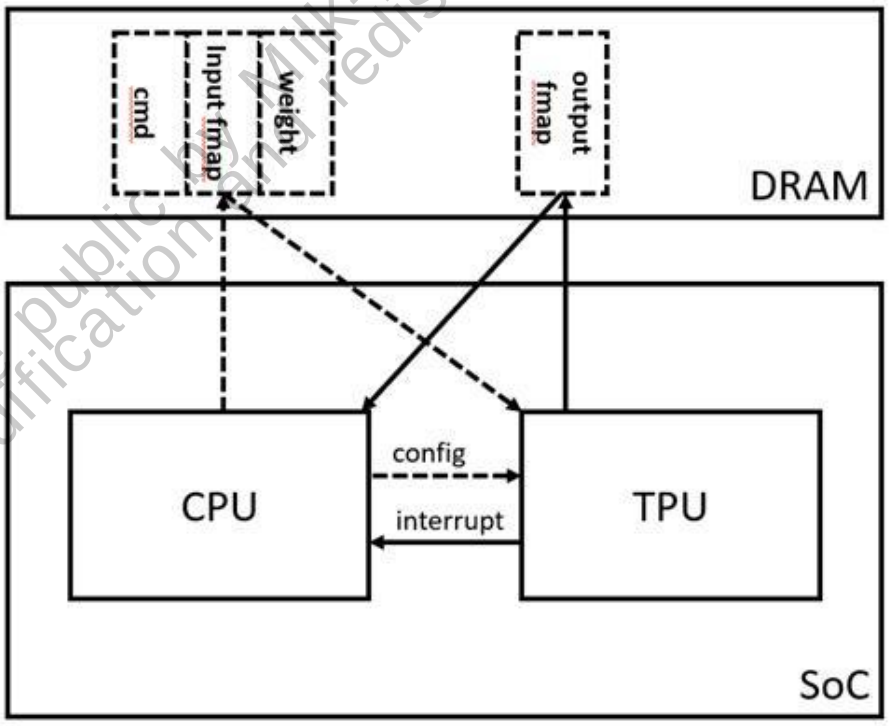


Figure 8- 1 TPU working mode diagram

8.1.2 Characteristics

The main features of TPU are as following:

- Support a wide range of AI models, such as Resnet、Vgg16、GoogleNet、Yolo、MobileNet, LSTM etc.
- Support dynamic voltage and frequency scaling
- Support high-performance and low-power CNN convolution
- Support high-performance, low-power fully connected layer matrix multiplication and addition
- Support various activation functions (such as: ReLU、tanh、Sigmoid, etc.)
- Support Elementwise tensor operations (including AND、OR、XOR、ADD、SUB、MIN、MAX、SHIFT、MUL、MAC)
- Support pooling operation (AVG and MAX)
- Support strided copy
- Support strided convolution
- Support Zero-Padding and Zero-Insertion
- support deconvolution and dilated convolution
- support depthwise separable convolution
- Support lossless compression
- Support LUT function
- Support Inverse Sqrt Root、Exponential、Sqrt Root、Division and other nonlinear operations
- Support 8-bit and 16-bit operation modes
- Support image batch and tiling processing
- Support high-performance multi-dimensional tensor movement and transpose
- Support high performance matrix movement and transpose
- Support performance monitoring unit (PMU)

9 Video interface

9.1 VI

9.1.1 Overview

Video input unit VI (Video Input) is a camera video data receiving camera module, which can support receiving video data through MIPI Rx interface or BT.656, BT.601, interface and DC (Digital Camera) signal, and then send it to the next level of image processing module (ISP). The functional block diagram of VI is shown in Figure 9.1.
VI is divided into two physical sub modules, which are MIPI RX and VI Proc. MIPI RX module receives and processes different video data, while VI Proc module will integrate different video signals into a single video signal required by ISP module.

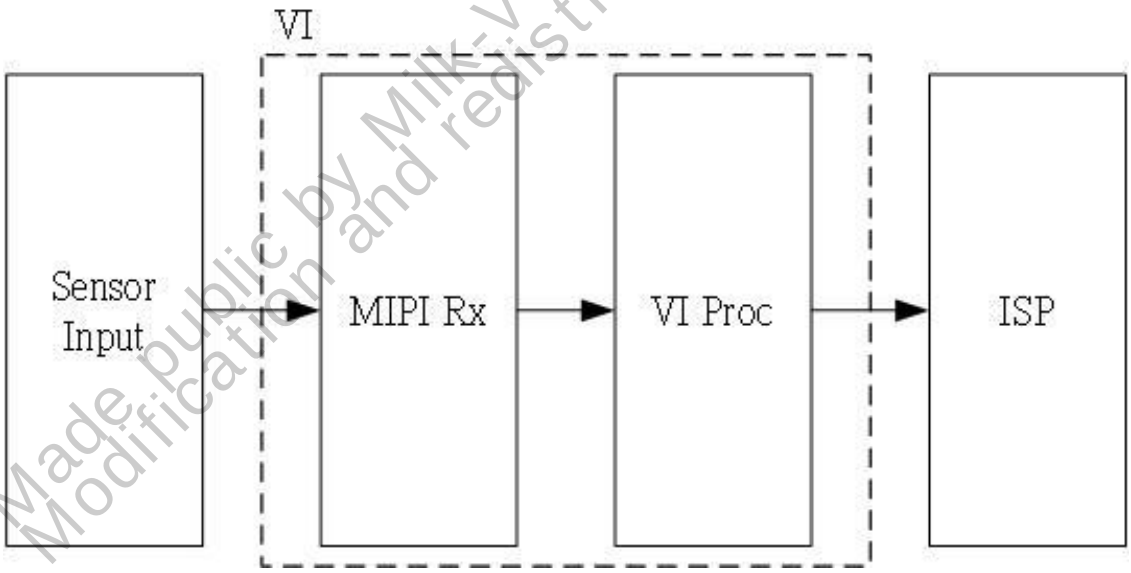


Figure 9- 1 VI function block diagram

9.1.2 Characteristics

MIPI Rx supports up to two sensor data inputs simultaneously.

• Single sensor supports linear input of up to 5M (2688x1944, 2880x1620) @ 20fps.
• MIPI Rx input supports a maximum data width of 12 bits.
• Supports MIPI Rx multi-channel fusion input (1, 2, 4 channels).
• Supports BT.656, BT.601 (only supports progressive mode).

DC interface
• Supports MIPI CSI-2 interface.
• Supports YUV422 format input via MIPI interface.

9.1.3 Mode function description

9.1.3.1 Typical applications

VI can support a variety of timing input and different interfaces, and do video input acquisition for different encoding methods. The system can use the register to configure different function modes to adapt to different video interfaces.
The VI module can support up to two input channels. The typical inputs are as follows:

• 1 channel 5M(2688x1944, 2880x1620) linear @20fps input

9.1.3.2 BT.656 interface timing

VI also supports BT.656 interface timing of Y / C combined input. During transmission, SAV and EAV are also used to indicate the beginning and end of valid line data, but only 8bit is used to transmit video signal, and brightness and chroma are transmitted in time-sharing mode, as shown in Figure 9-2

Figure 9- 2 BT.656horizontal interface timing

The difference between BT.656 and BT.1120 is only 16 bit (BT.1120) and 8 bit (BT.656) for image transmission, and other vertical timing and synchronous code formats are the same.

9.1.3.3 BT.601 interface timing

In addition to using synchronization codes bt.1120 and BT.656, VI supports BT.601 interface timing using a variety of different synchronization signals. The actual video data can be set to 16bit mode of Y / C separate input or 8bit mode of Y / C combined time-sharing input by register, while the synchronization mode can be set to vhs, vde or vsde mode by register. The detailed sequence is shown in the figure below.

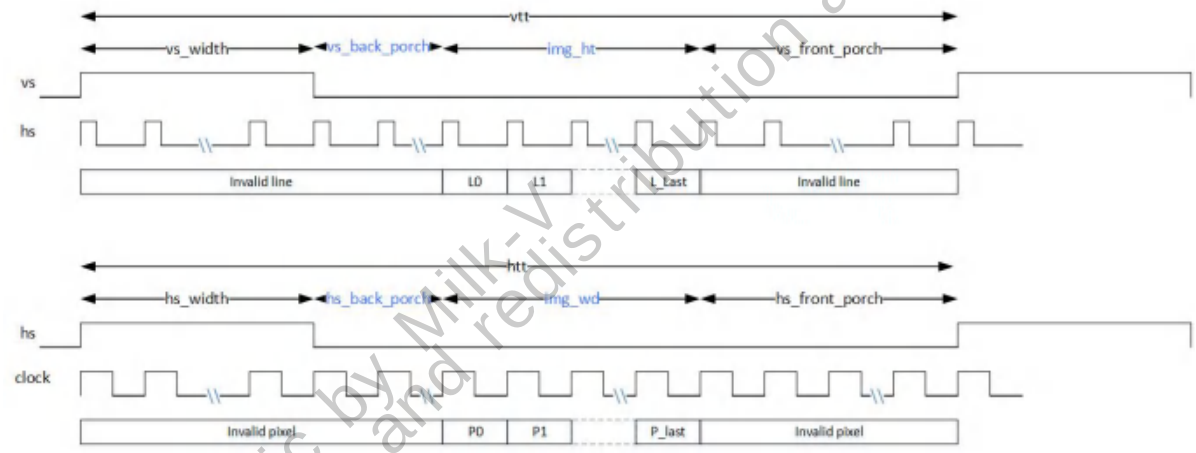


Figure 9- 3 BT.601 vhs Synchronization mode

The input synchronization signal of VHS mode is frame synchronization signal (VS) and line synchronization signal (HS). The system must set the number of hidden lines after frame (vs_back_porch), image height (img_ht), pixel number after line (hs_back_porch) and image width (img_wd)

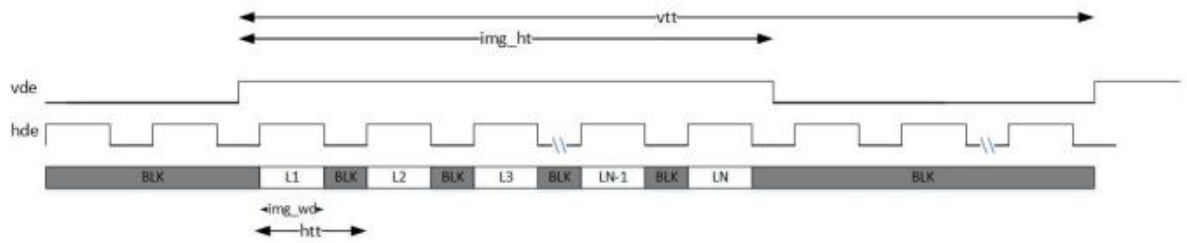


Figure 9- 4 BT.601 vde Synchronization mode

vde mode synchronization signals are frame valid signal (vde) and row valid signal (hde). In this mode, the system does not need to set the parameters related to time sequence and phase sequence. VI module will receive data according to hde / vde signal to update the frame according to vde signal.。

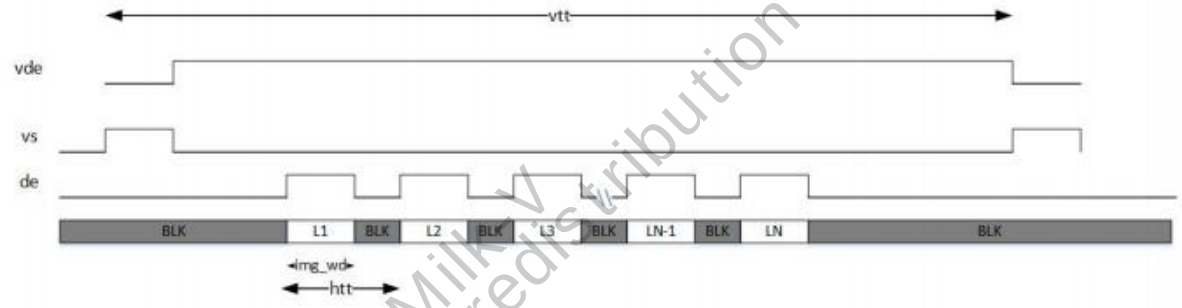


figure 9- 5 BT.601 vsde Synchronization mode

The synchronization signals of vsde mode are frame synchronization signal (vs) and effective pixel flag (de). In this mode, the system does not need to set the parameters related to time sequence and phase sequence. The VI module will receive data according to the de signal to update the frame according to the vs signal.

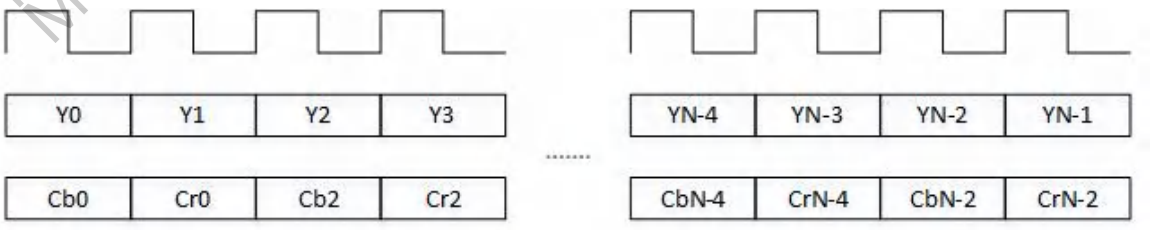


Figure 9- 6 BT.601 Y/C separate 16bit mode

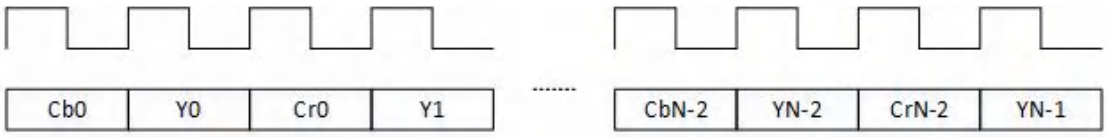


Figure 9- 7 BT.601 Y/C merge 8bit mode

9.1.3.4 Digital camera (DC) interface timing

VI supports the transmission of RAW format analog BT digital camera (DC) interface timing. In DC interface, it can support 8bit, 10bit, and 12bit three different modes. It can also use register settings to receive video signals by using receive synchronization code or three different synchronization modes similar to BT.601

Figure 9- 8 DC Synchronization code mode

Figure 9- 2 DC Synchronous signal mode -vhs mode

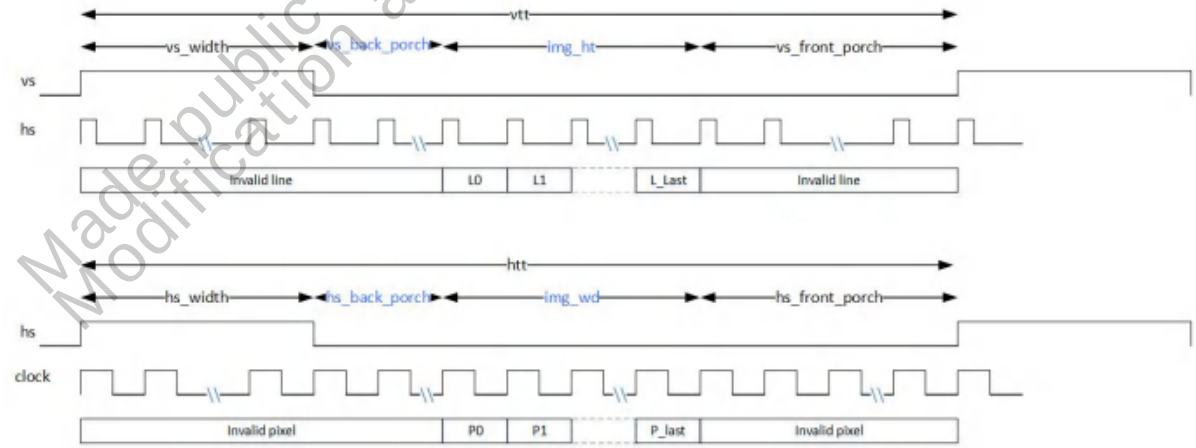


Figure 9- 10 DC Synchronous signal mode -VDE mode

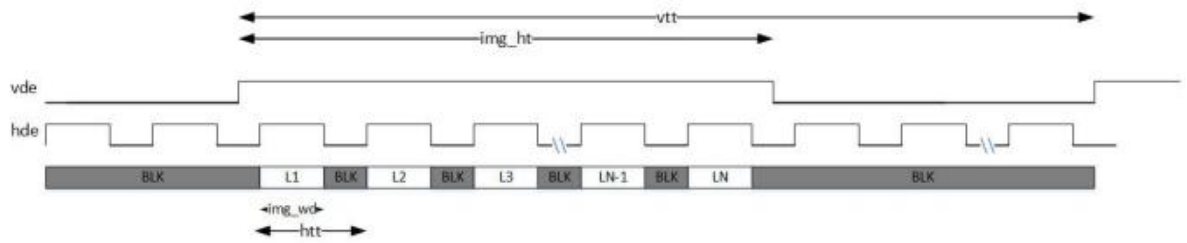
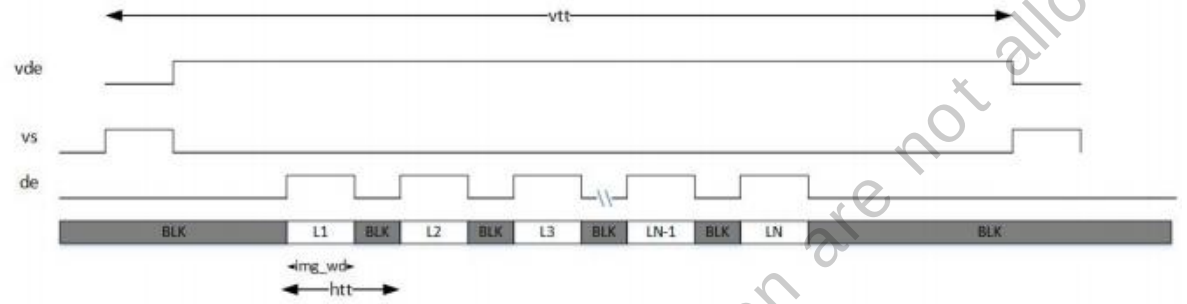


Figure9- 11 DC Synchronous signal mode -VSDE mode



9.1.4 Image storage mode

The images stored in DRAM are divided into Bayer 12bit and YCbCr 8 bit formats. Y / Cb / Cr is stored separately in three different DRAM positions. The arrangement of images in two formats (12bit / 8bit) in DRAM is shown in the figure below.

Figure 9-xx Bayer 12 bit Image storage mode

Bayer12bit		Pixel0	Pixel1	Pixel2	Pixel3				
		Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
DRAM Address	AddrN+0	4	5	6	7	8	9	10	11
	AddrN+1	4	5	6	7	8	9	10	11
	AddrN+2	0	1	2	3	0	1	2	3
	AddrN+3	4	5	6	7	8	9	10	11
	AddrN+4	4	5	6	7	8	9	10	11
	AddrN+5	0	1	2	3	0	1	2	3

Figure 9-xx YCbCr 8bit Image storage mode

YCbCr8bit		Pixel0	Pixel1	Pixel2	Pixel3				
		Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
DRAM Address	AddrN+0	0	1	2	3	4	5	6	7
	AddrN+1	0	1	2	3	4	5	6	7
	AddrN+2	0	1	2	3	4	5	6	7
	AddrN+3	0	1	2	3	4	5	6	7

9.1.5 VI register overview

CV1835 chip has two sets of the same VI module, the internal register offset address is the same, and the base address is 0x0A0C2000 and 0x0A0C4000。

Figure 9- 3 VI register overview

Name	Address Offset	Description
REG_00	0x000	MODE
REG_10	0x010	TTL_MODE_0

Name	Address Offset	Description
REG_14	0x014	TTL_MODE_1
REG_18	0x018	TTL_MODE_2
REG_1C	0x01c	TTL_MODE_3
REG_20	0x020	TTL_MODE_4
REG_24	0x024	TTL_MODE_5
REG_28	0x028	TTL_MODE_6
REG_30	0x030	TTL_MODE_7
REG_40	0x040	HDR_MODE_0
REG_44	0x044	HDR_MODE_1
REG_48	0x048	HDR_MODE_2
REG_50	0x050	BLC_MODE
REG_54	0x054	BLC_MODE_0
REG_58	0x058	BLC_MODE_1
REG_60	0x060	VI_PINMUX_0
REG_64	0x064	VI_PINMUX_1
REG_68	0x068	VI_PINMUX_2
REG_6C	0x06c	VI_PINMUX_3
REG_70	0x070	VI_PINMUX_4
REG_74	0x074	VI_PINMUX_5
REG_80	0x080	BT_PATH_0
REG_88	0x088	BT_PATH_2
REG_8C	0x08c	BT_PATH_3
REG_90	0x090	BT_PATH_4
REG_94	0x094	BT_PATH_5
REG_98	0x098	BT_PATH_6
REG_9C	0x09c	BT_PATH_7
REG_A0	0x0a0	BT_PATH_8
REG_A4	0x0a4	BT_PATH_A
REG_B0	0x0b0	CROP_0
REG_B4	0x0b4	CROP_1
REG_D0	0x0d0	MODE_CTRL
REG_D4	0x0d4	SYNC_CODE_0
REG_D8	0x0d8	SYNC_CODE_1
REG_DC	0x0dc	SYNC_CODE_2
REG_E0	0x0e0	SYNC_CODE_3
REG_E4	0x0e4	SYNC_CODE_4
REG_E8	0x0e8	SYNC_CODE_5
REG_EC	0x0ec	SYNC_CODE_6
REG_F0	0x0f0	SYNC_CODE_7
REG_F4	0x0f4	SYNC_CODE_8
REG_F8	0x0f8	SYNC_CODE_9
REG_FC	0x0fc	VS_GEN
REG_100	0x100	SYNC_CODE_A
REG_104	0x104	SYNC_CODE_B
REG_108	0x108	HDR_PATTEN_2
REG_110	0x110	HISPI_MODE_CTRL_0
REG_114	0x114	HISPI_MODE_CTRL_1
REG_118	0x118	HISPI_MODE_CTRL_2
REG_11C	0x11c	HISPI_MODE_CTRL_3
REG_120	0x120	HISPI_MODE_CTRL_4
REG_124	0x124	HISPI_MODE_CTRL_5

9.1.6 VI register overview

REG_00

Offset Address: 0x000

Bits	Name	Access	Description	Reset
2:0	reg_sensor_mac_mode	R/W	Sensor mode 3'b000: Disable 3'b001: CSI 3'b010: Sub-LVDS 3'b011: TTL	0x0
3	reg_bt_demux_enable	R/W	BT Demux enable	0x0
4	reg_csi_ctrl_enable	R/W	CSI controller enable	0x0
5	reg_csi_vs_inv	R/W	CSI VS inverse	0x1
6	reg_csi_hs_inv	R/W	CSI HS inverse	0x1
7	Reserved			
8	reg_sublvds_ctrl_enable	R/W	Sub-LVDS controller enable	0x0
9	reg_sublvds_vs_inv	R/W	Sub-LVDS VS inverse	0x1
10	reg_sublvds_hs_inv	R/W	Sub-LVDS HS inverse	0x1
11	reg_sublvds_hdr_inv	R/W	Sub-LVDS HDR inverse	0x1
31:12	Reserved			

REG_10

Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	reg_ttl_ip_en	R/W	TTL enable	0x0
2:1	reg_ttl_sensor_bit	R/W	TTL bit mode 2'b00: 8-bit 2'b01: 10-bit 2'b10: 12-bit 2'b11: 16-bit	0x0
3	Reserved			
5:4	reg_ttl_bt_fmt_out	R/W	TTL BT output format 2'b00: {Cb,Y},{Cr,Y} 2'b01: {Cr,Y},{Cb,Y} 2'b10: {Y,Cb},{Y,Cr} 2'b11: {Y,Cr},{Y,Cb}	0x2
7:6	Reserved			
11:8	reg_ttl_fmt_in	R/W	TTL input format 4'b0000: bt_2x with sync pattern, 9-bit BT656 4'b0001: bt_1x with sync pattern, 17-bit BT1120 4'b0010: bt_2x without sync pattern, 11-bit BT601 (vhs mode) 4'b0011: bt_1x without sync pattern, 19-bit BT601 (vhs mode) 4'b0100: bt_2x without sync pattern, 11-bit BT601 (vde mode) 4'b0101: bt_1x without sync pattern, 19-bit BT601 (vde mode)	0x0

Bits	Name	Access	Description	Reset
			4'b0110: bt_2x without sync pattern, 11-bit BT601 (vsde mode) 4'b0111: bt_1x without sync pattern, 19-bit BT601 (vsde mode) 4'b100x: sensor with sync pattern 4'b101x: sensor without sync pattern, use vs + hs (vhs mode) 4'b110x: sensor without sync pattern, use vde + hde (vde mode) 4'b111x: sensor without sync pattern, use vs + hde (vsde mode)	
13:12	reg_ttl_bt_data_seq	R/W	TTL bt data sequence 2'b00: Cb0-Y0-Cr0-Y1 2'b01: Cr0-Y0-Cb0-Y1 2'b10: Y0-Cb0-Y1-Cr0 2'b11: Y0-Cr0-Y1-Cb0	0x0
14	reg_ttl_vs_inv	R/W	TTL vs inverse	0x0
15	reg_ttl_hs_inv	R/W	TTL hs inverse	0x0
31:16	Reserved			

REG_14

Offset Address: 0x014

Bits	Name	Access	Description	Reset
11:0	reg_ttl_vs_bp	R/W	TTL vsync back porch setting	0x0
15:12	Reserved			
27:16	reg_ttl_hs_bp	R/W	TTL hsync back porch setting	0x0
31:28	Reserved			

REG_18

Offset Address: 0x018

Bits	Name	Access	Description	Reset
11:0	reg_ttl_img_wd	R/W	TTL image width setting	0x0
15:12	Reserved			
27:16	reg_ttl_img_ht	R/W	TTL image height setting	0x0
31:28	Reserved			

REG_1C

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
15:0	reg_ttl_sync_0	R/W	TTL sync code 0	0x0
31:16	reg_ttl_sync_1	R/W	TTL sync code 1	0x0

REG_20

Offset Address: 0x020

Bits	Name	Access	Description	Reset
15:0	reg_ttl_sync_2	R/W	TTL sync code 2	0x0
31:16	Reserved			

REG_24

Offset Address: 0x024

Bits	Name	Access	Description	Reset
15:0	reg_ttl_sav_vld	R/W	TTL valid line SAV	0x0
31:16	reg_ttl_sav_blk	R/W	TTL blanking line SAV	0x0

REG_28

Offset Address: 0x028

Bits	Name	Access	Description	Reset
15:0	reg_ttl_eav_vld	R/W	TTL valid line EAV	0x0
31:16	reg_ttl_eav_blk	R/W	TTL blanking line EAV	0x0

REG_30

Offset Address: 0x030

Bits	Name	Access	Description	Reset
2:0	reg_vi_sel	R/W	VI input mode select 3'h1: RAW 3'h2: BT601 3'h3: BT656 3'h4: BT1120 else: reserved	0x0
3	reg_vi_from	R/W	VI input from VI0 or VI1 1'b0: from VI0 1'b1: from VI1	0x0
4	reg_vi_clk_inv	R/W	VI clock inverse	0x0
5	reg_vi_v_sel_vs	R/W	1'b1: vs_in signal as vs 1'b0: vs_in signal as vde	0x1
6	reg_vi_vs_dbg	R/W	vsync source select	0x0
7	Reserved			
8	reg_pad_vi0_clk_inv	R/W	vi0 clk inverse	0x0
9	reg_pad_vi1_clk_inv	R/W	vi1 clk inverse	0x0
10	reg_pad_vi2_clk_inv	R/W	vi2 clk inverse	0x0
31:11	Reserved			

REG_40

Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	reg_sensor_mac_hdr_en	R/W	Sensor mac hdr manual mode enable Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
1	reg_sensor_mac_hdr_vsinv	R/W	Sensor mac vsync output inverse Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
2	reg_sensor_mac_hdr_hsinv	R/W	Sensor mac hsync output inverse Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
3	reg_sensor_mac_hdr_deinv	R/W	Sensor mac de output inverse Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
4	reg_sensor_mac_hdr_hdr0inv	R/W	Sensor mac hdr[0] output inverse Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
5	reg_sensor_mac_hdr_hdr1inv	R/W	Sensor mac hdr[1] output inverse Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
6	reg_sensor_mac_hdr_blcinv	R/W	Sensor mac blc output inverse	0x0

Bits	Name	Access	Description	Reset
			Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	
7	Reserved			
8	reg_sensor_mac_hdr_mode	R/W	Sensor mac hdr mode 1'b1 stands for HiSPi S-SP HDR mode, remove HDR blanking line Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:9	Reserved			

REG_44

Offset Address: 0x044

Bits	Name	Access	Description	Reset
12:0	reg_sensor_mac_hdr_shift	R/W	Sensor mac hdr long exposure shift (long exposure lines before 1st short exposure line) Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
15:13	Reserved			
28:16	reg_sensor_mac_hdr_vsize	R/W	Sensor mac hdr vsize Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:29	Reserved			

REG_48

Offset Address: 0x048

Bits	Name	Access	Description	Reset
12:0	reg_sensor_mac_info_line_num	R/W	Info line number Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x1
15:13	Reserved			
16	reg_sensor_mac_rm_info_line	R/W	Remove info line Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:17	Reserved			

REG_50

Offset Address: 0x050

Bits	Name	Access	Description	Reset
0	reg_sensor_mac_blc0_en	R/W	BLC0 mode enable	0x0
1	reg_sensor_mac_blc1_en	R/W	BLC1 mode enable	0x0
31:2	Reserved			

REG_54

Offset Address: 0x054

Bits	Name	Access	Description	Reset
12:0	reg_sensor_mac_blc0_start	R/W	BLC0 start line number	0x0

Bits	Name	Access	Description	Reset
15:13	Reserved			
28:16	reg_sensor_mac_blc0_size	R/W	BLC0 line size	0x4
31:29	Reserved			

REG_58

Offset Address: 0x058

Bits	Name	Access	Description	Reset
12:0	reg_sensor_mac_blc1_start	R/W	BLC1 start line number	0x0
15:13	Reserved			
28:16	reg_sensor_mac_blc1_size	R/W	BLC1 line size	0x4
31:29	Reserved			

REG_60

Offset Address: 0x060

Bits	Name	Access	Description	Reset
5:0	reg_vi_vs_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
7:6	Reserved			
13:8	reg_vi_hs_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
15:14	Reserved			
21:16	reg_vi_vde_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
23:22	Reserved			
29:24	reg_vi_hde_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
31:30	Reserved			

REG_64

Offset Address: 0x064

Bits	Name	Access	Description	Reset
5:0	reg_vi_d0_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
7:6	Reserved			
13:8	reg_vi_d1_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
15:14	Reserved			
21:16	reg_vi_d2_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
23:22	Reserved			
29:24	reg_vi_d3_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
31:30	Reserved			

REG_68

Offset Address: 0x068

Bits	Name	Access	Description	Reset
5:0	reg_vi_d4_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
7:6	Reserved			
13:8	reg_vi_d5_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
15:14	Reserved			
21:16	reg_vi_d6_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
23:22	Reserved			
29:24	reg_vi_d7_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
31:30	Reserved			

REG_6C

Offset Address: 0x06c

Bits	Name	Access	Description	Reset
5:0	reg_vi_d8_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
7:6	Reserved			
13:8	reg_vi_d9_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
15:14	Reserved			
21:16	reg_vi_d10_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
23:22	Reserved			
29:24	reg_vi_d11_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
31:30	Reserved			

REG_70

Offset Address: 0x070

Bits	Name	Access	Description	Reset
5:0	reg_vi_d12_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
7:6	Reserved			
13:8	reg_vi_d13_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
15:14	Reserved			
21:16	reg_vi_d14_sel	R/W	vi pin select [5]: from VI1 or VI0	0x0

Bits	Name	Access	Description	Reset
			[4:0]: from which VI pad count	
23:22	Reserved			
29:24	reg_vi_d15_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
31:30	Reserved			

REG_74

Offset Address: 0x074

Bits	Name	Access	Description	Reset
2:0	reg_vi_bt_d0_sel	R/W	vi bt pin select from which VI2 pad count	0x0
3	Reserved			
6:4	reg_vi_bt_d1_sel	R/W	vi bt pin select from which VI2 pad count	0x1
7	Reserved			
10:8	reg_vi_bt_d2_sel	R/W	vi bt pin select from which VI2 pad count	0x2
11	Reserved			
14:12	reg_vi_bt_d3_sel	R/W	vi bt pin select from which VI2 pad count	0x3
15	Reserved			
18:16	reg_vi_bt_d4_sel	R/W	vi bt pin select from which VI2 pad count	0x4
19	Reserved			
22:20	reg_vi_bt_d5_sel	R/W	vi bt pin select from which VI2 pad count	0x5
23	Reserved			
26:24	reg_vi_bt_d6_sel	R/W	vi bt pin select from which VI2 pad count	0x6
27	Reserved			
30:28	reg_vi_bt_d7_sel	R/W	vi bt pin select from which VI2 pad count	0x7
31	Reserved			

REG_80

Offset Address: 0x080

Bits	Name	Access	Description	Reset
0	reg_bt_clr_sync_lost_1t	R/W	Clear sync_lost signal	0x0
1	reg_bt_ip_en	R/W	BT path enable	0x0
2	reg_bt_ddr_mode	R/W	BT DDR mode	0x0
3	reg_bt_hs_gate_by_vde	R/W	HS gating by VDE	0x0
4	reg_bt_vs_inv	R/W	vsync inverse	0x0
5	reg_bt_hs_inv	R/W	hsync inverse	0x0
6	reg_bt_vs_as_vde	R/W	input vsync as vde	0x0
7	reg_bt_hs_as_hde	R/W	input hsync as hde	0x0
14:8	reg_bt_sw_en_clk	R/W	Clock gating software enable [0]: delay control clock enable [1]: timing demultiplexer clock enable [2]: timing gen clock enable [3]: rx decode 0 clock enable [4]: rx decode 1 clock enable	0x0

Bits	Name	Access	Description	Reset
			[5]: rx decode 2 clock enable [6]: rx decode 3 clock enable	
15	Reserved			
17:16	reg_bt_demux_ch	R/W	Demux setting 2'h0: No demux 2'h1: Demux 2 2'h2: Demux 3 2'h3: Demux 4	0x0
19:18	Reserved			
22:20	reg_bt_fmt_sel	R/W	3'b000 : bt_2x with sync pattern, 9-bit BT656 (clock + 8-bit data) 3'b001 : bt_1x with sync pattern, 17-bit BT1120 (clock + 16-bit data) 3'b010 : bt_2x without sync pattern, 11-bit BT601 (clock + 8-bit data + vs + hs) (vhs_mode) 3'b011 : bt_1x without sync pattern, 19-bit BT601 (clock + 16-bit data + vs + hs) (vhs_mode) 3'b100 : bt_2x without sync pattern, 11-bit BT601 (clock + 8-bit data + vde + hde) (vde_mode) 3'b101 : bt_1x without sync pattern, 19-bit BT601 (clock + 16-bit data + vde + hde) (vde_mode) 3'b110 : bt_2x without sync pattern, 11-bit BT601 (clock + 8-bit data + vs + hde) (vsde_mode) 3'b111 : bt_1x without sync pattern, 19-bit BT601 (clock + 16-bit data + vs + hde) (vsde_mode)	0x0
31:23	Reserved			

REG_88

Offset Address: 0x088

Bits	Name	Access	Description	Reset
11:0	reg_bt_img_wd_m1	R/W	BT image width	0x0
15:12	Reserved			
27:16	reg_bt_img_ht_m1	R/W	BT image height	0x0
31:28	Reserved			

REG_8C

Offset Address: 0x08c

Bits	Name	Access	Description	Reset
11:0	reg_bt_vs_bp_m1	R/W	BT vsync back porch	0x0
15:12	Reserved			
27:16	reg_bt_hs_bp_m1	R/W	BT hsync back porch	0x0
31:28	Reserved			

REG_90

Offset Address: 0x090

Bits	Name	Access	Description	Reset
7:0	reg_bt_vs_fp_m1	R/W	BT vsync front porch	0x0

Bits	Name	Access	Description	Reset
15:8	reg_bt_hs_fp_m1	R/W	BT hsync front porch	0x0
31:16	Reserved			

REG_94

Offset Address: 0x094

Bits	Name	Access	Description	Reset
7:0	reg_bt_sync_0	R/W	BT sync code byte 0	0x0
15:8	reg_bt_sync_1	R/W	BT sync code byte 1	0x0
23:16	reg_bt_sync_2	R/W	BT sync code byte 2	0x0
31:24	Reserved			

REG_98

Offset Address: 0x098

Bits	Name	Access	Description	Reset
7:0	reg_bt_sav_vld_0	R/W	BT valid SAV sync code for demux 0	0x0
15:8	reg_bt_sav_blk_0	R/W	BT blank SAV sync code for demux 0	0x0
23:16	reg_bt_eav_vld_0	R/W	BT valid EAV sync code for demux 0	0x0
31:24	reg_bt_eav_blk_0	R/W	BT blank EAV sync code for demux 0	0x0

REG_9C

Offset Address: 0x09c

Bits	Name	Access	Description	Reset
7:0	reg_bt_sav_vld_1	R/W	BT valid SAV sync code for demux 1	0x0
15:8	reg_bt_sav_blk_1	R/W	BT blank SAV sync code for demux 1	0x0
23:16	reg_bt_eav_vld_1	R/W	BT valid EAV sync code for demux 1	0x0
31:24	reg_bt_eav_blk_1	R/W	BT blank EAV sync code for demux 1	0x0

REG_A0

Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
7:0	reg_bt_sav_vld_2	R/W	BT valid SAV sync code for demux 2	0x0
15:8	reg_bt_sav_blk_2	R/W	BT blank SAV sync code for demux 2	0x0
23:16	reg_bt_eav_vld_2	R/W	BT valid EAV sync code for demux 2	0x0
31:24	reg_bt_eav_blk_2	R/W	BT blank EAV sync code for demux 2	0x0

REG_A4

Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
7:0	reg_bt_sav_vld_3	R/W	BT valid SAV sync code for demux 3	0x0
15:8	reg_bt_sav_blk_3	R/W	BT blank SAV sync code for demux 3	0x0
23:16	reg_bt_eav_vld_3	R/W	BT valid EAV sync code for demux 3	0x0
31:24	reg_bt_eav_blk_3	R/W	BT blank EAV sync code for demux 3	0x0

REG_B0

Offset Address: 0x0b0

Bits	Name	Access	Description	Reset
12:0	reg_sensor_mac_crop_start_x	R/W	Pixels before reg_sensor_mac_crop_start_x will be cropped in each line if enable reg_sensor_mac_crop_en.	0xFFF
15:13	Reserved			
28:16	reg_sensor_mac_crop_end_x	R/W	Pixels after reg_sensor_mac_crop_end_x will be	0xFFF

Bits	Name	Access	Description	Reset
			cropped in each line if enable reg_sensor_mac_crop_en.	
30:29	Reserved			
31	reg_sensor_mac_crop_en	R/W	enable crop function	0x0

REG_B4

Offset Address: 0x0b4

Bits	Name	Access	Description	Reset
12:0	reg_sensor_mac_crop_start_y	R/W	Lines before reg_sensor_mac_crop_start_y will be cropped in each frame if enable reg_sensor_mac_crop_en.	0xFFF
15:13	Reserved			
28:16	reg_sensor_mac_crop_end_y	R/W	Lines after reg_sensor_mac_crop_end_y will be cropped in each frame if enable reg_sensor_mac_crop_en.	0xFFF
31:29	Reserved			

REG_D0

Offset Address: 0x0d0

Bits	Name	Access	Description	Reset
0	reg_ttl_as_slvds_enable	R/W	Sub-LVDS lane enable for each lane	0x0
7:1	Reserved			
9:8	reg_ttl_as_slvds_bit_mode	R/W	Sub-LVDS bit mode 2'b00: 8-bit 2'b01: 10-bit 2'b10: 12-bit	0x2
10	reg_ttl_as_slvds_data_reverse	R/W	Sub-LVDS data packet bit inverse	0x0
11	Reserved			
12	reg_ttl_as_slvds_hdr_mode	R/W	Sub-LVDS HDR mode enable	0x0
13	reg_ttl_as_slvds_hdr_pattern	R/W	Sub-LVDS HDR pattern mode 1'b0: pattern 1 1'b1: pattern 2	0x0
31:14	Reserved			

REG_D4

Offset Address: 0x0d4

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_sync_1st	R/W	Sub-LVDS SYNC code 1st word	0xFFF
15:12	Reserved			
27:16	reg_ttl_as_slvds_sync_2nd	R/W	Sub-LVDS SYNC code 2nd word	0x000
31:28	Reserved			

REG_D8

Offset Address: 0x0d8

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_sync_3rd	R/W	Sub-LVDS SYNC code 3rd word	0x000
15:12	Reserved			
27:16	reg_ttl_as_slvds_norm_bk_sav	R/W	Normal mode blanking SAV	0xAB0
31:28	Reserved			

REG_DC

Offset Address: 0x0dc

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_norm_bk_eav	R/W	Normal mode blanking EAV	0xB60
15:12	Reserved			
27:16	reg_ttl_as_slvds_norm_sav	R/W	Normal mode active SAV	0x800
31:28	Reserved			

REG_E0

Offset Address: 0x0e0

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_norm_eav	R/W	Normal mode active EAV	0x9D0
15:12	Reserved			
27:16	reg_ttl_as_slvds_n0_bk_sav	R/W	HDR mode n0 blanking SAV	0x2B0
31:28	Reserved			

REG_E4

Offset Address: 0x0e4

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_n0_bk_eav	R/W	HDR mode n0 blanking EAV	0x360
15:12	Reserved			
27:16	reg_ttl_as_slvds_n1_bk_sav	R/W	HDR mode n1 blanking SAV	0x6B0
31:28	Reserved			

REG_E8

Offset Address: 0x0e8

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_n1_bk_eav	R/W	HDR mode n1 blanking EAV	0x760
15:12	Reserved			
27:16	reg_ttl_as_slvds_n0_lef_sav	R/W	Sub-LVDS mode: n0 long exposure sav Sub-LVDS 12-bit LEF SAV n0 (801) Sub-LVDS 10-bit LEF SAV n0 (004) HiSPi P-SP mode: SOL T1 (800)	0x801
31:28	Reserved			

REG_EC

Offset Address: 0x0ec

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_n0_lef_eav	R/W	Sub-LVDS mode: n0 long exposure eav Sub-LVDS 12-bit LEF EAV n0 (9D1) Sub-LVDS 10-bit LEF EAV n0 (1D4) HiSPi P-SP mode: EOL T1 (A00)	0x9D1
15:12	Reserved			
27:16	reg_ttl_as_slvds_n0_sef_sav	R/W	Sub-LVDS mode: n0 short exposure sav Sub-LVDS 12-bit SEF SAV n0 (802) Sub-LVDS 10-bit SEF SAV n0 (008) HiSPi P-SP mode: SOL T2 (820)	0x802
31:28	Reserved			

REG_F0

Offset Address: 0x0f0

Bits	Name	Access	Description	Reset

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_n0_sef_eav	R/W	Sub-LVDS mode: n0 short exposure eav Sub-LVDS 12-bit SEF EAV n0 (9D2) Sub-LVDS 10-bit SEF EAV n0 (1d8) HiSPi P-SP mode: EOL T2 (A20)	0x9D2
15:12	Reserved			
27:16	reg_ttl_as_slvds_n1_lef_sav	R/W	Sub-LVDS mode: n1 long exposure sav Sub-LVDS 12-bit LEF SAV n1 (C01) Sub-LVDS 10-bit LEF SAV n1 (404) HiSPi P-SP mode: SOF T1 (C00)	0xC01
31:28	Reserved			

REG_F4

Offset Address: 0x0f4

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_n1_lef_eav	R/W	Sub-LVDS mode: n1 long exposure eav Sub-LVDS 12-bit LEF EAV n1 (DD1) Sub-LVDS 10-bit LEF EAV n1 (5D4) HiSPi P-SP mode: EOF T1 (E00)	0xDD1
15:12	Reserved			
27:16	reg_ttl_as_slvds_n1_sef_sav	R/W	Sub-LVDS mode: n1 short exposure sav Sub-LVDS 12-bit SEF SAV n1 (C02) Sub-LVDS 10-bit SEF SAV n1 (408) HiSPi P-SP mode: SOF T2 (C20)	0xC02
31:28	Reserved			

REG_F8

Offset Address: 0x0f8

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_n1_sef_eav	R/W	Sub-LVDS mode: n1 short exposure eav Sub-LVDS 12-bit SEF EAV n1 (DD2) Sub-LVDS 10-bit SEF EAV n1 (5D8) HiSPi P-SP mode: EOF T2 (E20)	0xDD2
31:12	Reserved			

REG_FC

Offset Address: 0x0fc

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_vs_gen_sync_code	R/W	vs generate sync code value using scenario: HiSPi P-SP HDR	0xC00
12	reg_ttl_as_slvds_vs_gen_by_sync_code	R/W	vs generate by identical sync code using scenario: HiSPi P-SP HDR	0x0
31:13	Reserved			

REG_100

Offset Address: 0x100

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_n0_lsef_sav	R/W	SAV for n0 long & short exposure both exist line only used for pattern 2	0x803
15:12	Reserved			
27:16	reg_ttl_as_slvds_n0_lsef_eav	R/W	EAV for n0 long & short exposure both exist line only used for pattern 2	0x9D3

Bits	Name	Access	Description	Reset
31:28	Reserved			

REG_104

Offset Address: 0x104

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_n1_lsef_sav	R/W	SAV for n1 long & short exposure both exist line only used for pattern 2	0xC03
15:12	Reserved			
27:16	reg_ttl_as_slvds_n1_lsef_eav	R/W	EAV for n1 long & short exposure both exist line only used for pattern 2	0xDD3
31:28	Reserved			

REG_108

Offset Address: 0x108

Bits	Name	Access	Description	Reset
13:0	reg_ttl_as_slvds_hdr_p2_hsize	R/W	Hsize for pattern 2	0xF0
15:14	Reserved			
29:16	reg_ttl_as_slvds_hdr_p2_hblank	R/W	Hblank size for pattern 2	0x14
31:30	Reserved			

REG_110

Offset Address: 0x110

Bits	Name	Access	Description	Reset
0	reg_ttl_as_hispi_mode	R/W	HiSPi mode enable 1'b0: Sub-LVDS 1'b1: HiSPi	0x0
1	reg_ttl_as_hispi_use_hsize	R/W	HiSPi DE de-assert by register count	0x0
3:2	Reserved			
4	reg_ttl_as_hispi_hdr_psp_mode	R/W	HiSPi P-SP HDR mode enable	0x0
31:5	Reserved			

REG_114

Offset Address: 0x114

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_hispi_norm_sof	R/W	HiSPi SOF sync code	0xC00
15:12	Reserved			
27:16	reg_ttl_as_hispi_norm_eof	R/W	HiSPi EOF sync code	0xE00
31:28	Reserved			

REG_118

Offset Address: 0x118

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_hispi_hdr_t1_sof	R/W	HiSPi HDR T1 SOF	0xC00
15:12	Reserved			
27:16	reg_ttl_as_hispi_hdr_t1_eof	R/W	HiSPi HDR T1 EOF	0xE00
31:28	Reserved			

REG_11C

Offset Address: 0x11c

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_hispi_hdr_t1_sol	R/W	HiSPi HDR T1 SOL	0x800
15:12	Reserved			
27:16	reg_ttl_as_hispi_hdr_t1_eol	R/W	HiSPi HDR T1 EOL	0xA00
31:28	Reserved			

REG_120

Offset Address: 0x120

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_hispi_hdr_t2_sof	R/W	HiSPi HDR T2 SOF	0xC20
15:12	Reserved			
27:16	reg_ttl_as_hispi_hdr_t2_eof	R/W	HiSPi HDR T2 EOF	0xE20
31:28	Reserved			

REG_124

Offset Address: 0x124

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_hispi_hdr_t2_sol	R/W	HiSPi HDR T2 SOL	0x820
15:12	Reserved			
27:16	reg_ttl_as_hispi_hdr_t2_eol	R/W	HiSPi HDR T2 EOL	0xA20
31:28	Reserved			

9.2 MIPI Rx

9.2.1 Overview

The main function of MIPI Rx (Mobile Industry Processor Interface Receiver) module is to receive the video data transmitted by CMOS sensor. It supports different serial video signal input such as MIPI D-PHY, sub LVDS (Low-Voltage Differential Signal) and HiSPi (High-Speed Serial Pixel Interface), and then the processing is transformed into internal video timing, which is transmitted to the next level of video processing module (ISP).

MIPI Rx module can be divided into PHY and Controller. PHY module integrates analog and digital parts, mainly converting serial signals into parallel signals, while Controller module is responsible for decoding different video data formats and transmitting them to the back-end video processing module (ISP). The functional block diagram and its position in the system are shown in Figure 9-13.

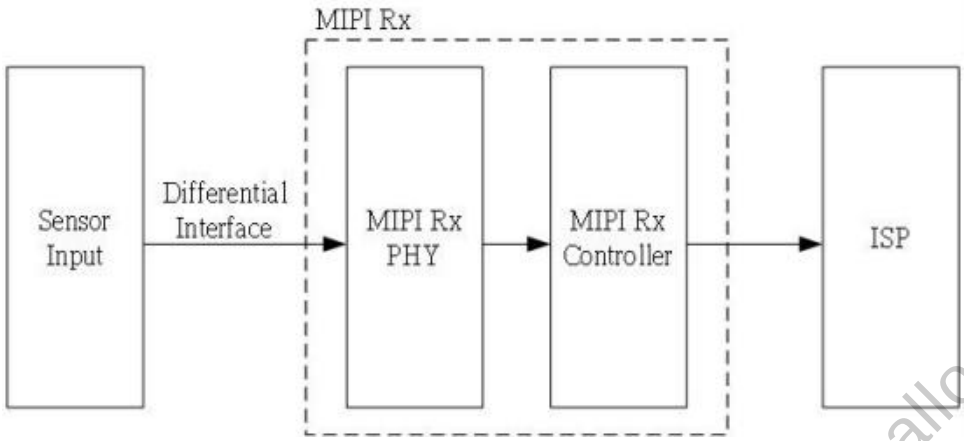


Figure 9- 4 MIPI Rx Functional Block Diagrams and Position

9.2.2 Charateristics

- Supported MIPI DPHY-ver2.1
- Maximum support for single sensor linear input is 5M (2688x1944, 2880x1620) @20fps
- Up to 4-Lane MIPI D-PHY interface is supported for a single input, with a maximum of 1.5Gbps/Lane
- Support for parsing RAW8/ RAW10/ RAW12/RAW16 data types
- Support for parsing YUV422 8-bit/ YUV422 10-bit data types
- Support for configurable number of lanes and lane order.

9.2.3 Function Description

9.2.3.1 Typical Application

In applications that use image sensors, the MIPI Rx module registers are set, and the MIPI Rx also supports the transmission of different speeds and resolutions, and is compatible with multiple image sensor formats.

The MIPI Rx is only responsible for interface timing conversion and decoding, and does not handle the image processing part. Therefore, it can support any resolution and

frame rate as long as the bandwidth is satisfied. The MIPI Rx bandwidth has two limitations: the interface data rate of the PHY and the internal processing speed. The input interface supports a maximum of 1.5Gbps/Lane, and the maximum internal processing speed is 600M*1pixels/s.

	Common mode voltage	Differential mode voltage	Maximum clock frequency	Maximum data rate per lane
MIPI DPHY	200mV	200mV	750MHz	1.5Gbps

Figure 9- 5 Interface Types Supported by MIPI Rx

9.2.3.2 MIPI Interface Data Formats

MIPI specification is developed and maintained by different working groups, corresponding to different applications. MIPI Rx supports D-PHY and CSI-2 (Camera Serial Interface). D-PHY specifies the transmission specification of physical layer, while CSI-2 specifies the format and protocol of Camera output packet.

• D-PHY

D-PHY is a high-speed physical layer standard issued by MIPI Alliance, which specifies the physical characteristics and transmission protocol of interface layer. D-PHY adopts the low-voltage differential signal technology of 200mV source synchronization, and the data green rate range of each Lane supports up to 2500Mbps. D-PHY can work in two modes: low power (LP) and high speed (HS).

• CSI-2

CSI-2 is a data protocol for camera, which specifies the data packet format of communication between host and peripheral.

CSI-2 can support image applications with different pixel formats, and the minimum granularity of data transmission is byte. In order to enhance the performance of CSI-2, the number of data lanes can be selected. CSI-2 protocol specifies the mechanism for the sender to package pixel data into bytes, and it indicates the way to allocate and manage multiple data lanes. Byte data is organized in the form of packets, which are transmitted between SOT and EOT. The receiver parses the corresponding packets according to the protocol and recovers the original pixel data.

MIPI Rx supports the parsing of RAW8/ RAW10/ RAW12/ YUV422-8bit/ YUV422-10bit data types.

CSI-2 data packet is divided into long packet and short packet, including check code, which can correct and detect errors.

Both long packets and short packets are transmitted between SOT and EOT. In the gap of data transmission, D-PHY is in LP mode. The transmission mechanism of CSI-2 packet is shown in the figure. PH and PF represent Packet Header and Packet Footer respectively.

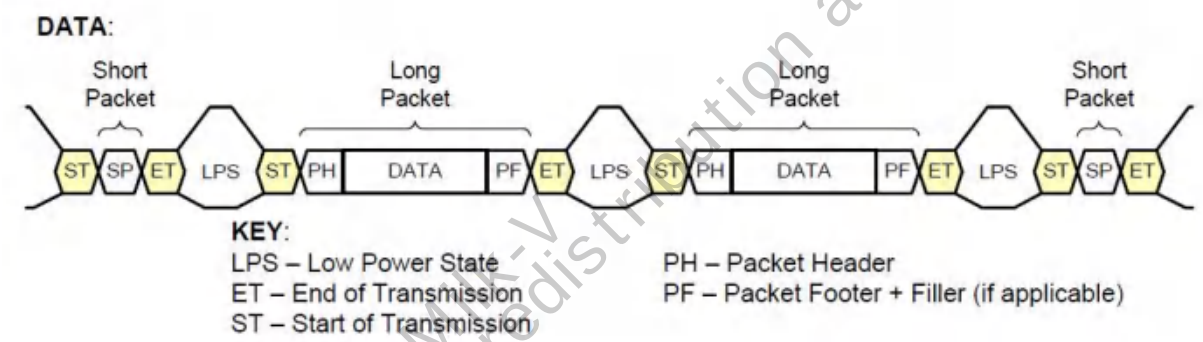


Figure 9- 6 Transmission Mechanism of Data Packet

The long packet is used to transmit effective pixel data, which is divided into five parts: Data ID, Word Count, ECC, PAYLOAD and CHECKSUM.

The Data ID contains Virtual Channel and Data Type. Virtual Channel controls the channel used for transmission, and different channels can be used to transmit different data. Data Type specifies the type of data to be transmitted.

Word Count represents the amount of data that the receiver needs to receive.

ECC is an error correcting code, which can correct or detect the error of Data Type and Word Count.

PAYLOAD is the pixel data to be transmitted.

CHECKSUM is the check sum generated by the linear feedback shift register, which is used to check the PAYLOAD data.
The structure of the long package is shown in Figure 9-16.

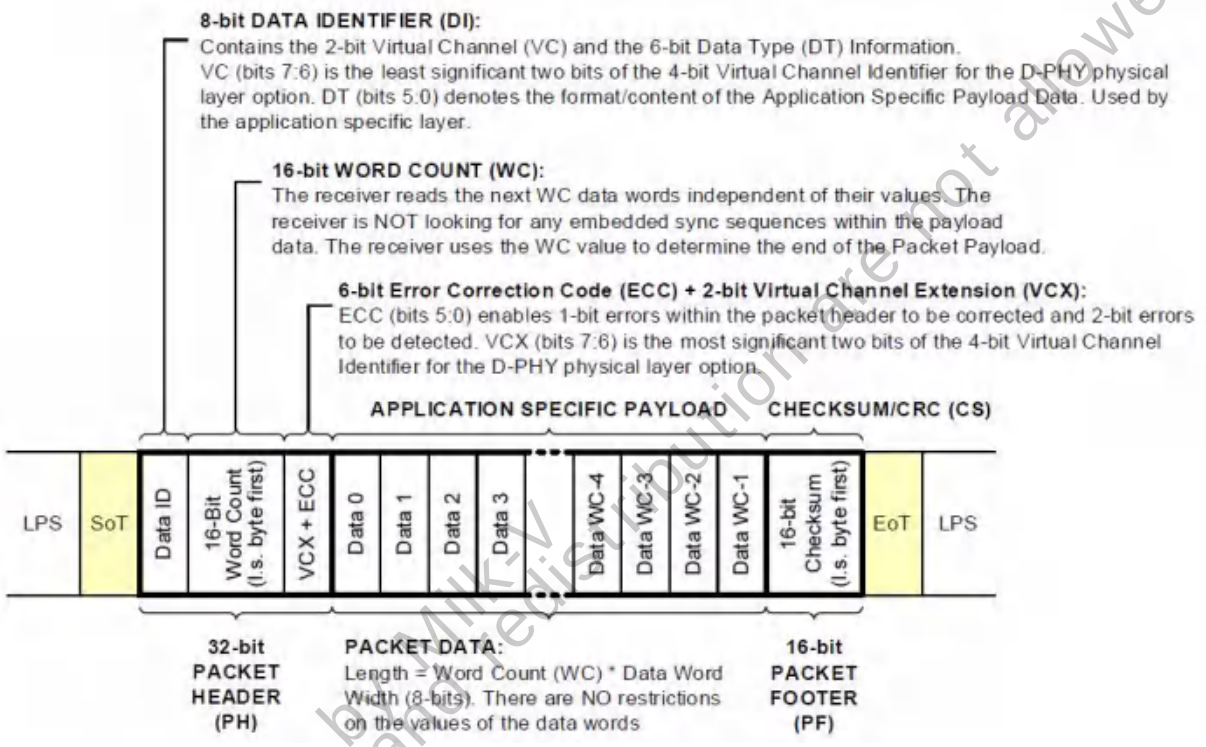


Figure 9- 7 CSI-2 Long Packet Format

The short packet is used to transmit information synchronously, including Data ID, Data Field and ECC. Its format is shown in Figure 9-17.

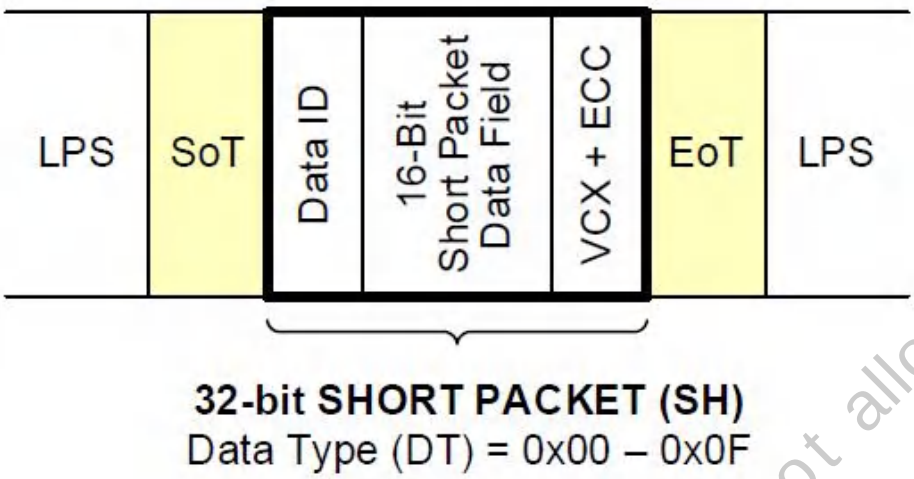


Figure 9- 8 CSI-2 Short Packet Format

MIPI Rx supports six video formats, including YUV422-8bit, YUV422-10bit, RAW8, RAW10, RAW12, and RAW16. Different data formats are transmitted as follows.

The transmission mode of YUV422-8bit is in the form of UYVY, as shown in Figure 9-23.

Figure 9- 18 YUV422 8-bit Data Transmission Sequence

The correspondence of packets to video signals is shown in Figure 9-19.

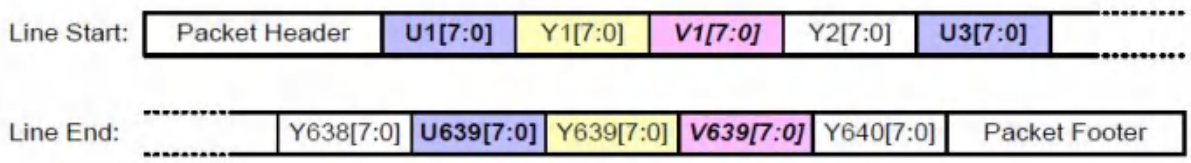


Figure 109 YUV422 8-bit Transmission

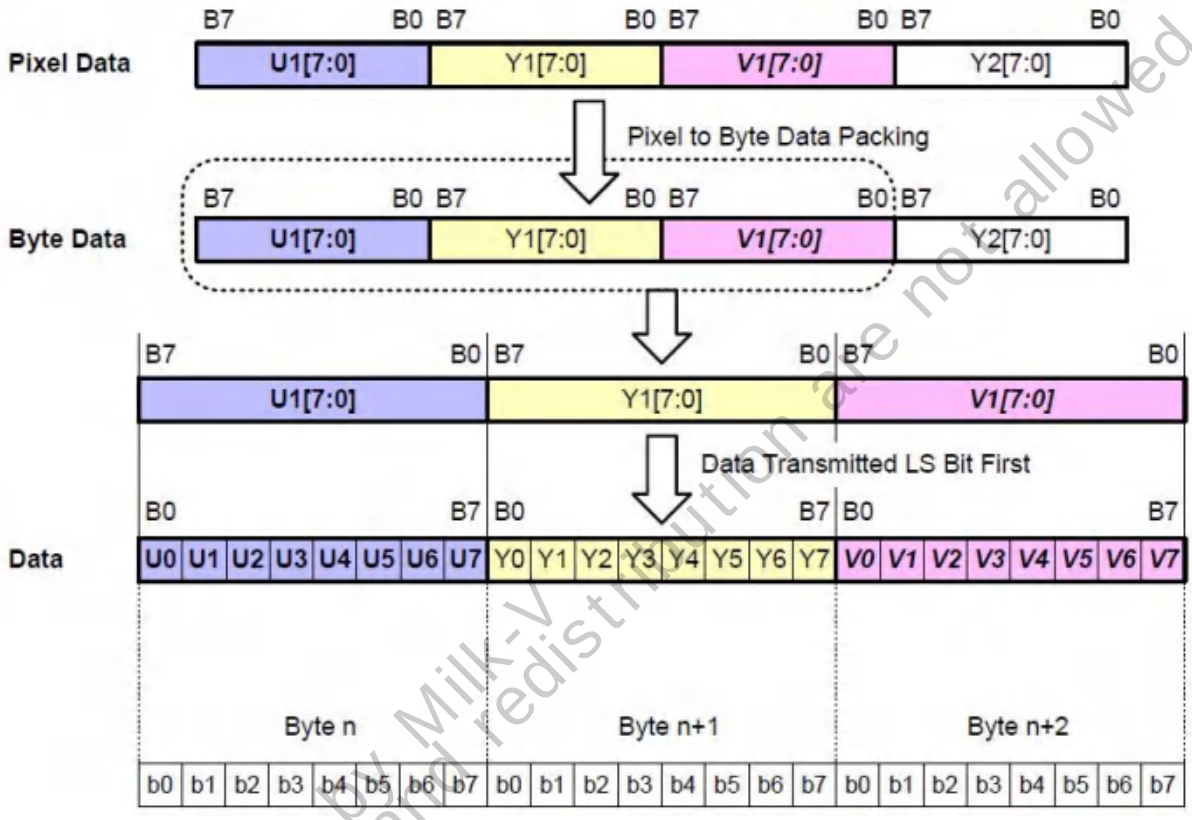


Figure 9- 19 YUV422-8bit Data Packet Transmission Correspondence

The transmission format of the whole frame is shown in Figure 9-20.

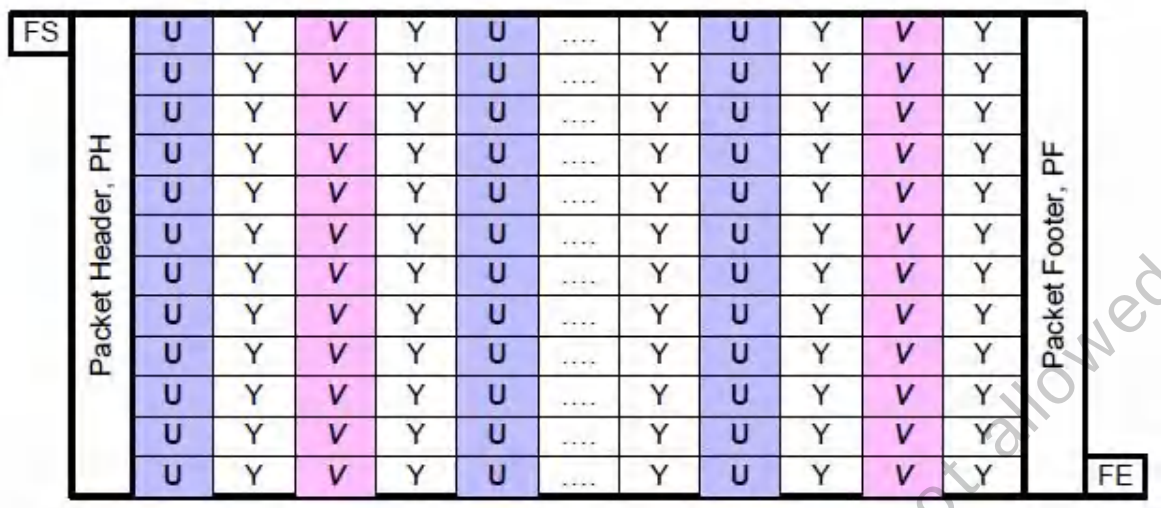


Figure 9- 9 YUV422 8-bit Frame Format

The transmission mode of YUV422-10bit is also UYVY, and the transmission sequence is shown in Figure 9-21.

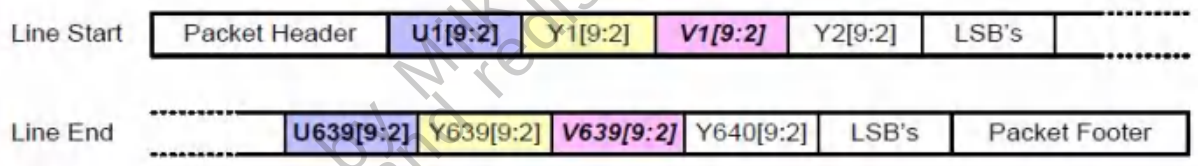


Figure 9- 10 YUV422-10bit Data Transmission Sequence

The correspondence of packets to video signals is shown in Figure 9-22.

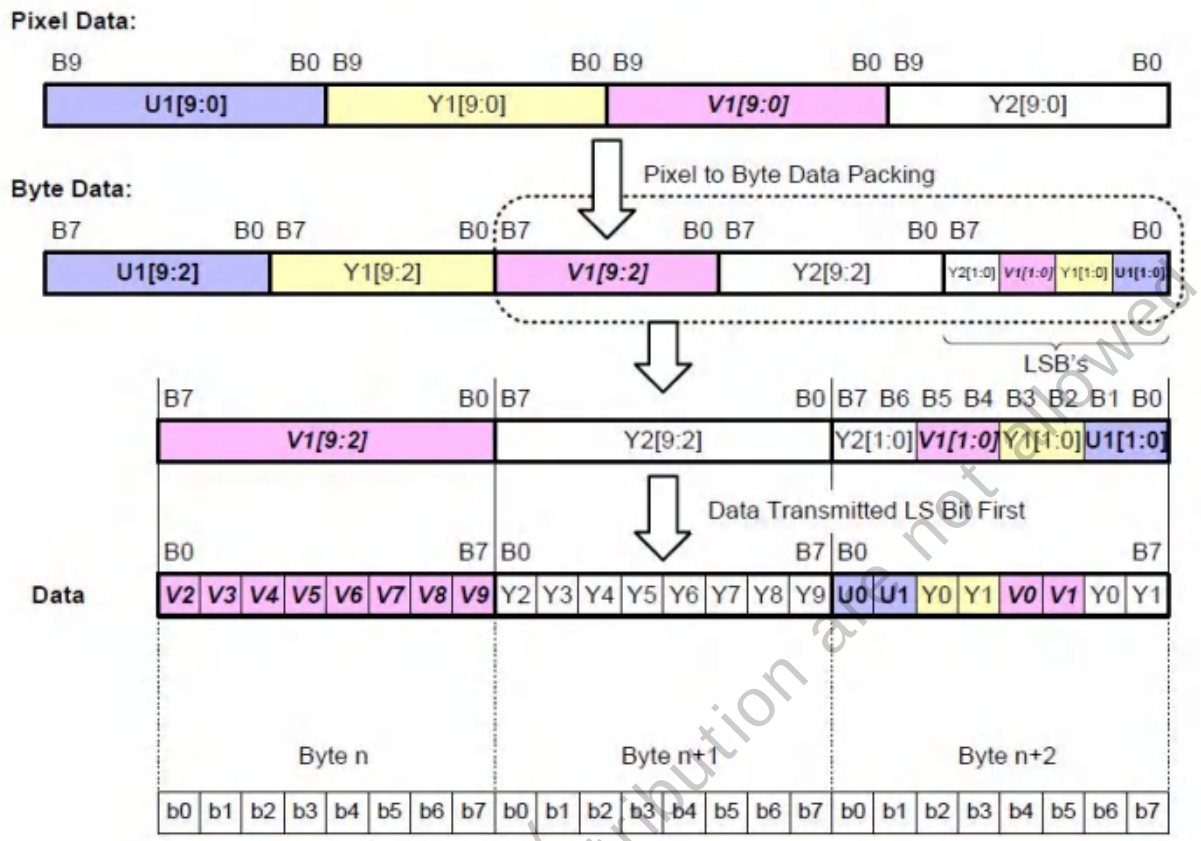


Figure 9- 11 YUV422-10bit Data Packet Transmission Correspondence

The transmission format of the whole frame is shown in Figure 9-23.

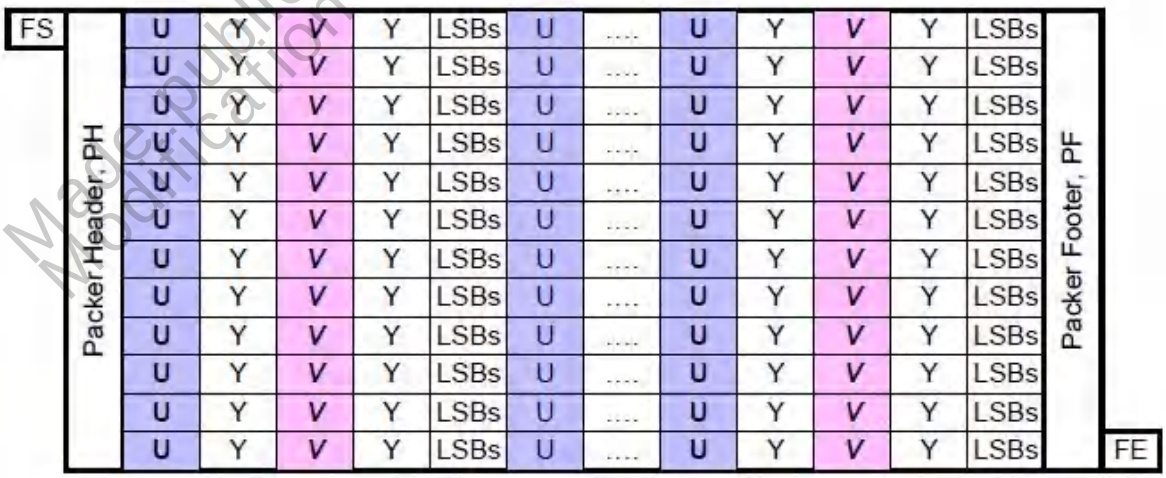


Figure 9- 12 YUV422-10bit Frame Format

The transmission sequence of RAW8 is shown in Figure 9-24.

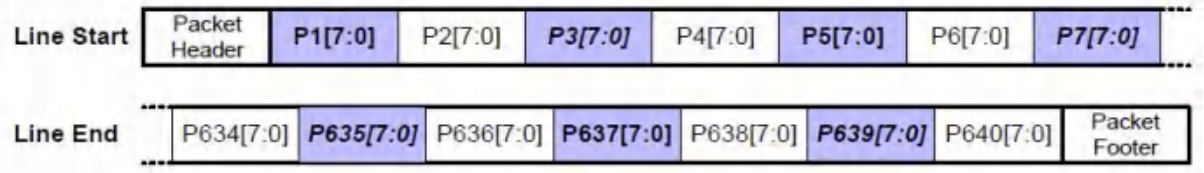


Figure 9- 13 RAW8 Data Transmission Sequence

The transmission format of the whole frame is shown in Figure 9-25.

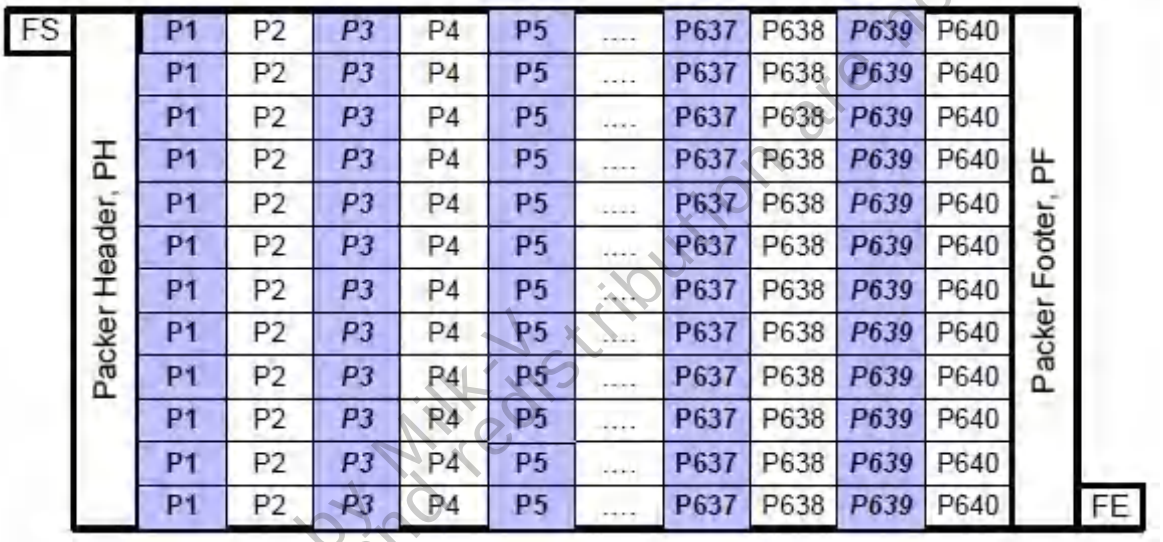


Figure 9- 14 RAW8 Frame Format

The transmission sequence of RAW10 is shown in Figure 9-26.

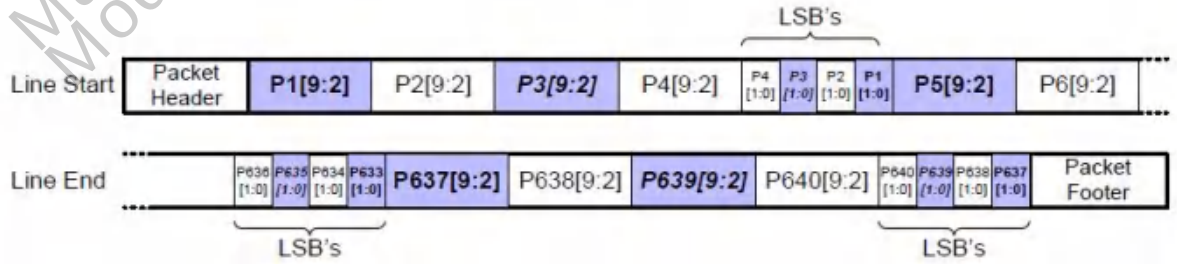


Figure 9- 15 RAW10 Data Transmission Sequence

The transmission format of the whole frame is shown in Figure 9-27.

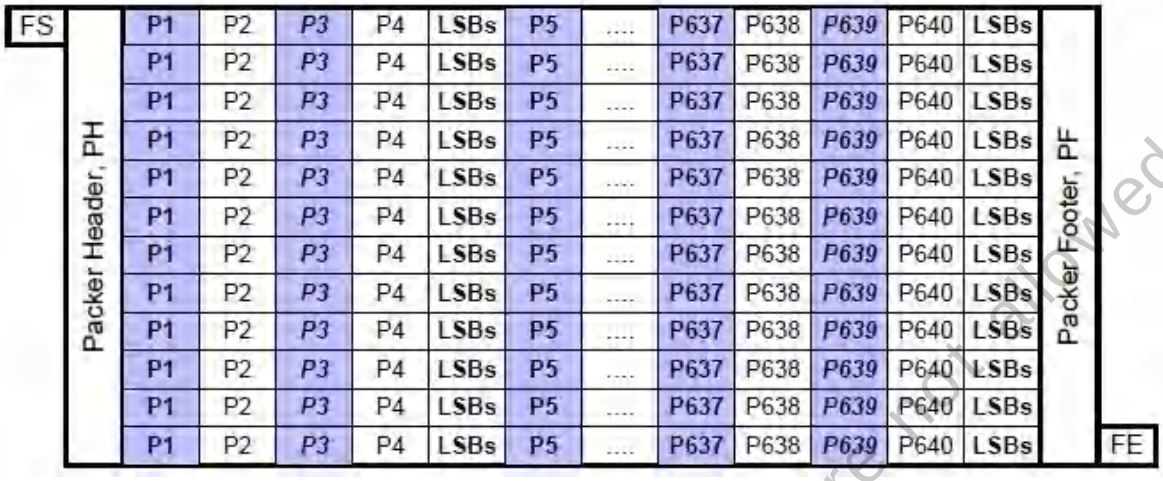


Figure 9- 16 RAW10 Frame Format

The transmission sequence of RAW12 is shown in Figure 9-28.

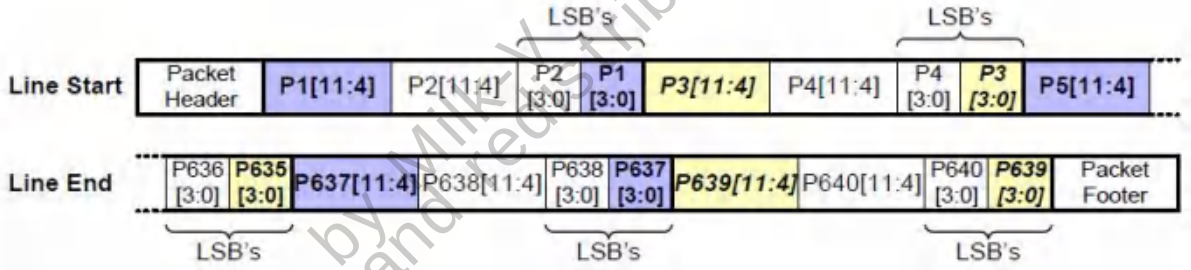


Figure 9- 17 RAW12Data Transmission Sequenc

The transmission format of the whole frame is shown in Figure 9-29.

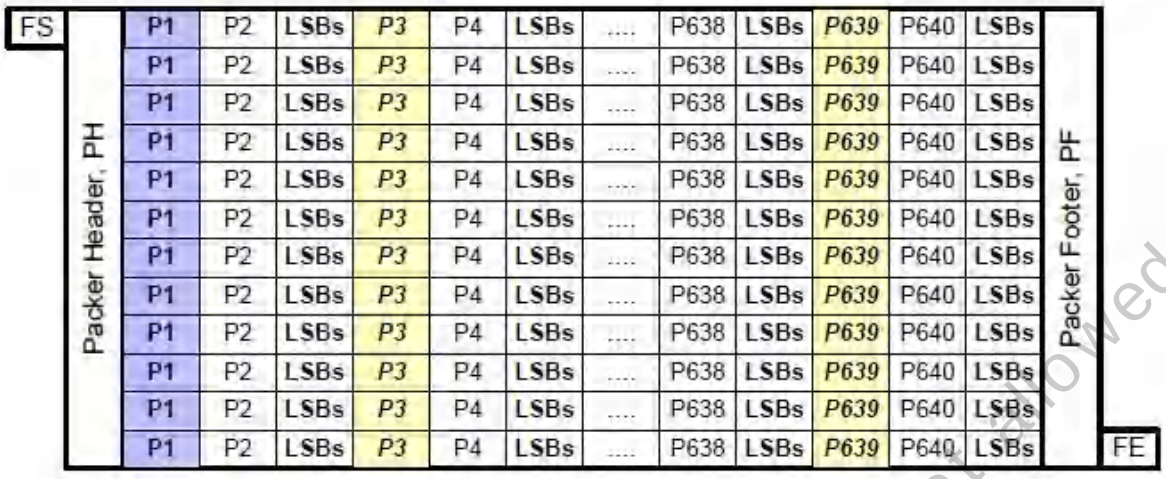
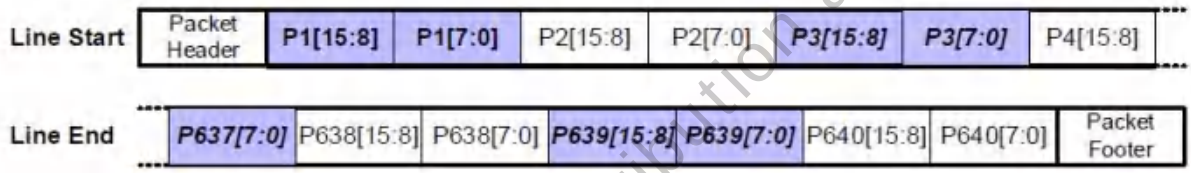


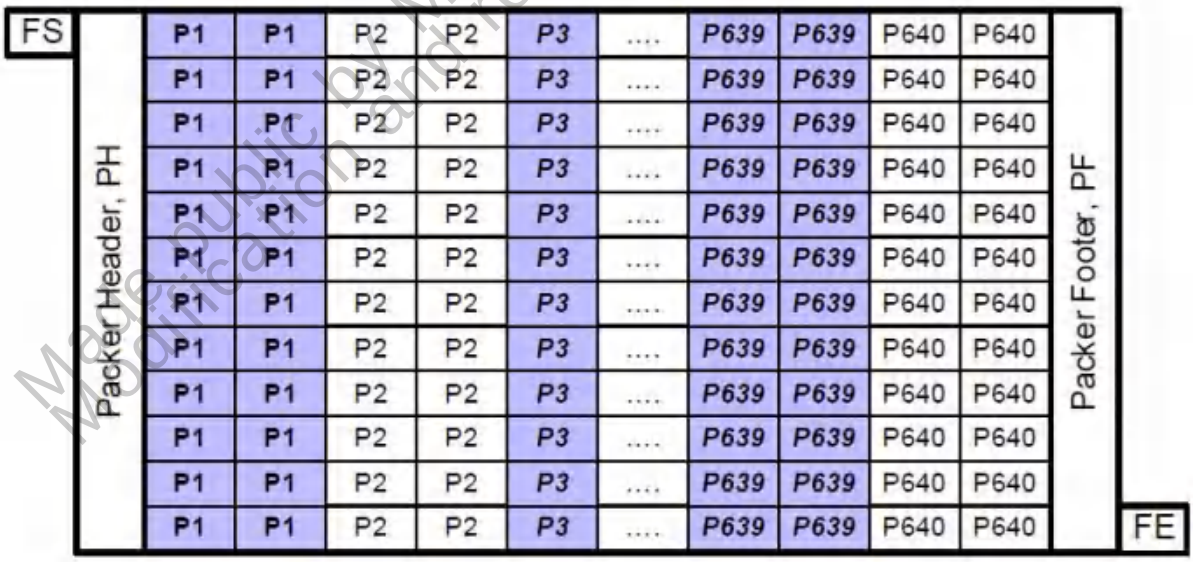
Figure 9- 29 RAW12 Frame Format

The transmission sequence of RAW16 is shown in Figure 9-30.



图表 9-18 RAW16Data Transmission Sequenc

The transmission format of the whole frame is shown in Figure 9-31.



图表 9-19 RAW16 Frame Format

9.2.3.3 MIPI Interface Linear Mode

The linear mode transmission format of MIPI interface is shown in Figure 9-32. The transmission of each graph starts with Frame Start (FS) and ends with Frame End (FE). The video content in the middle is based on the behavior unit, and each long packet transmits a complete video line. The long packet format is regulated by MIPI standard. Each row has 32bit Packet Header (PH), which contains the Virtual Channel and Data Type information of the current row.

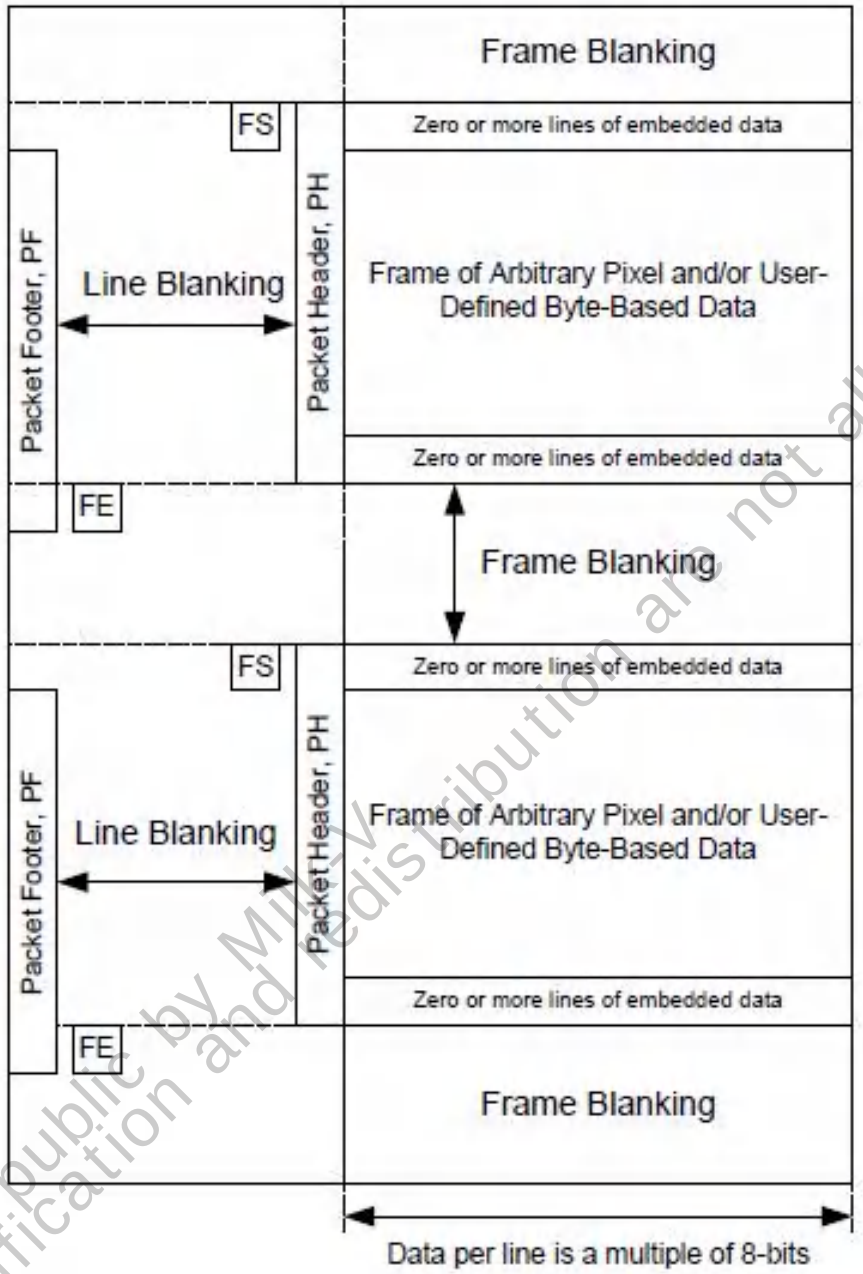


Figure 9- 20 MIPI Interface Image Format

9.2.3.4 MIPI Interface Wide Dynamic Mode

MIPI Rx supports four kinds of Wide Dynamic (WDR) modes of MIPI interface.

1. Use Data Type (DT) to distinguish long and short exposure data
2. Use the Identification Code (ID) to distinguish the long and short exposure data

3. Use the register to set the delay interval of long and short exposure data

The WDR transmission mode using DT is shown in Figure 9-33. Different exposure lengths share a group of FS / FE short packets, and the packet header of the long packet contains DT information. Different DT can be used to distinguish the long and short exposure data. The real data format DT and the two groups of DT representing the long and short exposure data can be set by registers. MIPI Rx can then analyze the correct wide dynamic timing and send it to the rear video processing module.

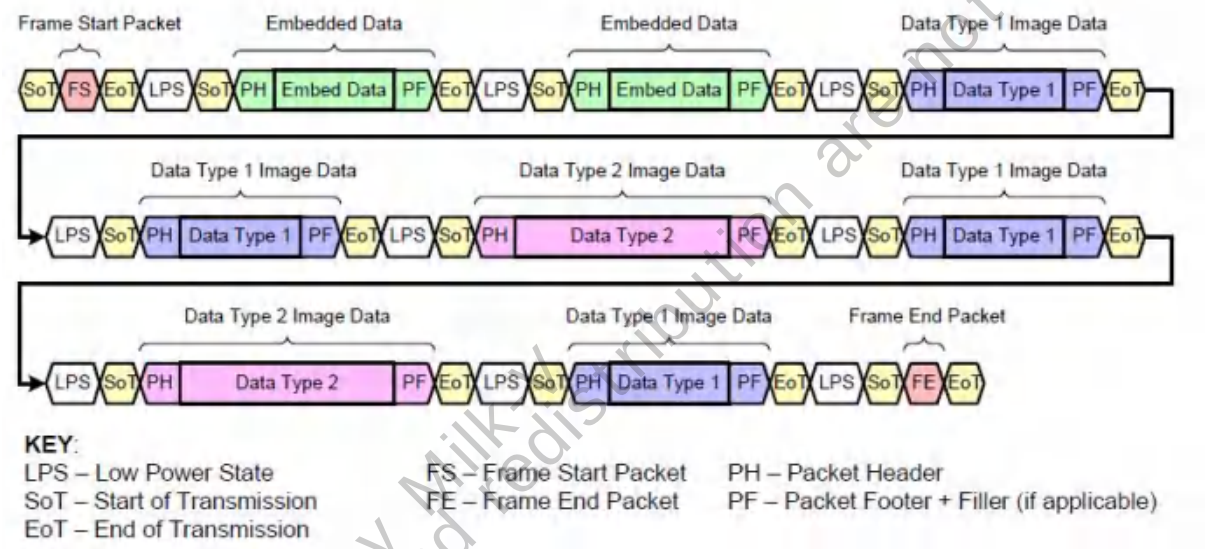


Figure 9- 21 MIPI Interface Wide Dynamic Data Transfer (using DT)

The WDR transmission mode using ID is shown in Figure 9-34. Different exposure lengths share a group of FS / FE short packets. The first four pixels of each long packet in the transmission data are used to transmit the Identification Code (ID) representing different exposure lengths. The ID representing long and short exposures can be set by the register. MIPI Rx will use ID to expose different video signals, remove the first four pixels and then send them to the video processing module.

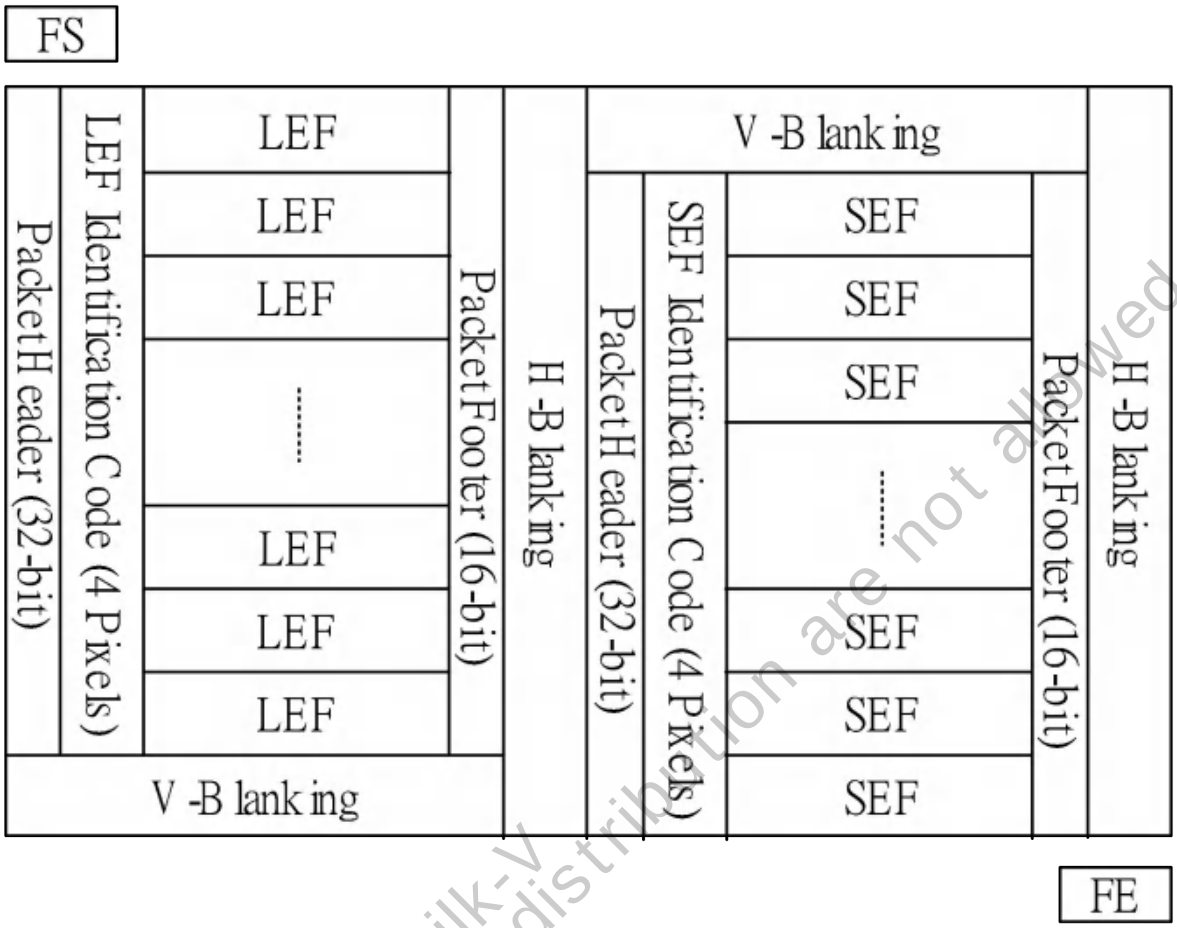


Figure 9- 22 MIPI Interface Wide Dynamic Data Transfer (using ID)

The last kind of transmission mode supported WDR is that there is no DT or ID to indicate the transmitted long packet is the content of long exposure or short exposure. Users must set their own registers to indicate the difference in the number of exposure lines between long exposure and short exposure. MIPI Rx will analyze the corresponding timing to the video processing module. The actual transmission timing is shown in Figure 9-35.

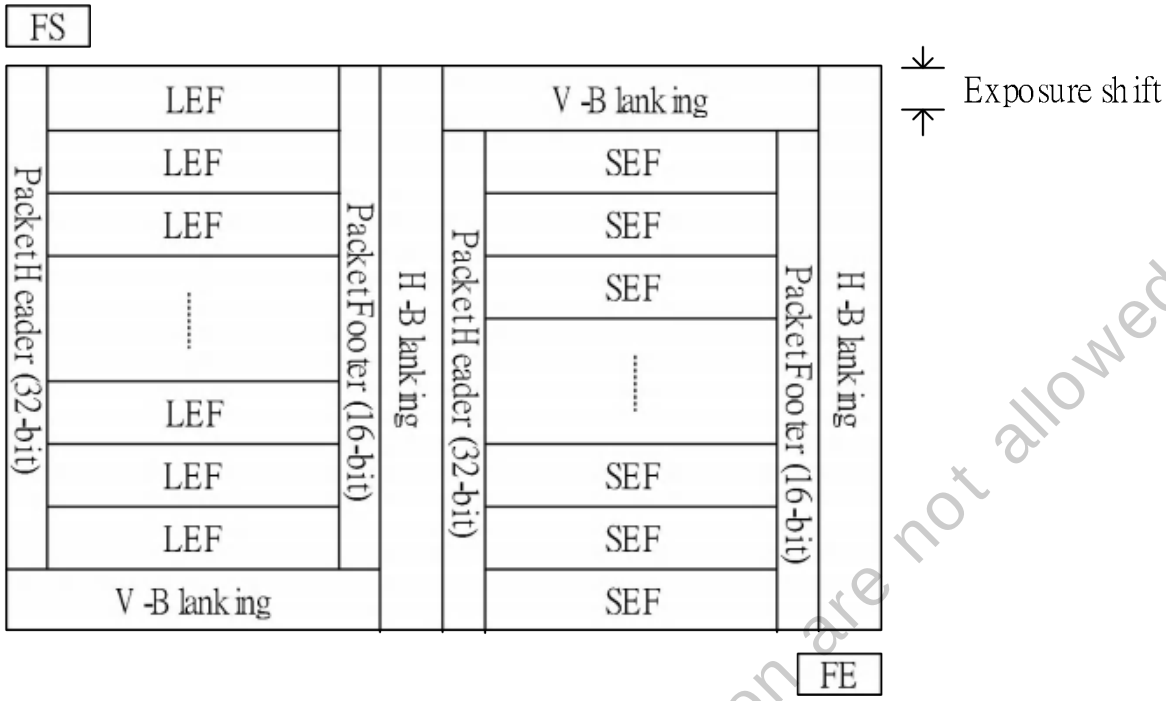


Figure 9- 23 MIPI Interface Wide Dynamic Data Transfer (register setting)

9.2.4 MIPI Rx Register Overview

Up to one set of MIPI Rx modules can be used simultaneously in the chip, which is mainly divided into three sets of registers. The first part is the register that controls the PHY module, with a base address of 0x0A0D0000. The second part is the register that controls the CSI module, with base addresses of 0x0A0C2400 and 0x0A0C4400. The third part is the register that controls the Sub-LVDS and HiSPi modules, with base addresses of 0x0A0C2200 and 0x0A0C4200.

Figure 9- 24 MIPI Rx PHY Register Overview

Name	Address Offset	Description
REG_00	0x000	PD_CTRL
REG_04	0x004	ANALOG_CTRL
REG_30	0x030	SENSOR_MODE_CTRL
REG_34	0x034	ANALOG_CAL_0
REG_38	0x038	ANALOG_CAL_1
REG_3C	0x03c	ANALOG_CAL_2
REG_40	0x040	ANALOG_CAL_3
REG_44	0x044	ANALOG_CAL_4
REG_48	0x048	ANALOG_CAL_5
REG_80	0x080	CLOCK_INVERCE_CTL

Name	Address Offset	Description
REG_A0	0x0a0	
REG_A4	0x0a4	
REG_A8	0x0a8	
REG_AC	0x0ac	

Offset = 0x0A0D0300

Name	Address Offset	Description
REG_00	0x000	SENSOR_MODE_CTRL
REG_04	0x004	LANE_SWAP_0
REG_08	0x008	LANE_SWAP_1
REG_0C	0x00c	CSI_GLB_CTL_0
REG_20	0x020	SLVDS_CTRL_0
REG_24	0x024	SLVDS_CTRL_1
REG_D0_0	0x100	D0_REG_CTRL_CALIB_0
REG_D0_1	0x104	D0_REG_CTRL_CALIB_1
REG_D0_3	0x10c	D0_CALIB_RESULT_0
REG_D0_4	0x110	D0_CALIB_RESULT_1
REG_D0_5	0x114	D0_CALIB_RESULT_2
REG_D0_6	0x118	D0_CALIB_RESULT_3
REG_D0_7	0x11c	D0_CALIB_RESULT_4
REG_D0_8	0x120	D0_CALIB_RESULT_5
REG_D0_9	0x124	D0_CALIB_RESULT_6
REG_D0_A	0x128	D0_CALIB_RESULT_7
REG_D1_0	0x140	D1_REG_CTRL_CALIB_0
REG_D1_1	0x144	D1_REG_CTRL_CALIB_1
REG_D1_3	0x14c	D1_CALIB_RESULT_0
REG_D1_4	0x150	D1_CALIB_RESULT_1
REG_D1_5	0x154	D1_CALIB_RESULT_2
REG_D1_6	0x158	D1_CALIB_RESULT_3
REG_D1_7	0x15c	D1_CALIB_RESULT_4
REG_D1_8	0x160	D1_CALIB_RESULT_5
REG_D1_9	0x164	D1_CALIB_RESULT_6
REG_D1_A	0x168	D1_CALIB_RESULT_7
REG_D2_0	0x180	D2_REG_CTRL_CALIB_0
REG_D2_1	0x184	D2_REG_CTRL_CALIB_1
REG_D2_3	0x18c	D2_CALIB_RESULT_0
REG_D2_4	0x190	D2_CALIB_RESULT_1
REG_D2_5	0x194	D2_CALIB_RESULT_2
REG_D2_6	0x198	D2_CALIB_RESULT_3
REG_D2_7	0x19c	D2_CALIB_RESULT_4
REG_D2_8	0x1a0	D2_CALIB_RESULT_5
REG_D2_9	0x1a4	D2_CALIB_RESULT_6
REG_D2_A	0x1a8	D2_CALIB_RESULT_7
REG_D3_0	0x1c0	D3_REG_CTRL_CALIB_0
REG_D3_1	0x1c4	D3_REG_CTRL_CALIB_1
REG_D3_3	0x1cc	D3_CALIB_RESULT_0
REG_D3_4	0x1d0	D3_CALIB_RESULT_1
REG_D3_5	0x1d4	D3_CALIB_RESULT_2
REG_D3_6	0x1d8	D3_CALIB_RESULT_3
REG_D3_7	0x1dc	D3_CALIB_RESULT_4
REG_D3_8	0x1e0	D3_CALIB_RESULT_5

Name	Address Offset	Description
REG_D3_9	0x1e4	D3_CALIB_RESULT_6
REG_D3_A	0x1e8	D3_CALIB_RESULT_7

Offset = 0x0A0D0600

Name	Address Offset	Description
REG_00	0x000	SENSOR_MODE_CTRL
REG_04	0x004	LANE_SWAP_0
REG_08	0x008	LANE_SWAP_1
REG_0C	0x00c	CSI_GLB_CTL_0
REG_20	0x020	SLVDS_CTRL_0
REG_D0_0	0x100	D0_REG_CTRL_CALIB_0
REG_D0_1	0x104	D0_REG_CTRL_CALIB_1
REG_D0_3	0x10c	D0_CALIB_RESULT_0
REG_D0_4	0x110	D0_CALIB_RESULT_1
REG_D0_5	0x114	D0_CALIB_RESULT_2
REG_D0_6	0x118	D0_CALIB_RESULT_3
REG_D0_7	0x11c	D0_CALIB_RESULT_4
REG_D0_8	0x120	D0_CALIB_RESULT_5
REG_D0_9	0x124	D0_CALIB_RESULT_6
REG_D0_A	0x128	D0_CALIB_RESULT_7
REG_D1_0	0x140	D1_REG_CTRL_CALIB_0
REG_D1_1	0x144	D1_REG_CTRL_CALIB_1
REG_D1_3	0x14c	D1_CALIB_RESULT_0
REG_D1_4	0x150	D1_CALIB_RESULT_1
REG_D1_5	0x154	D1_CALIB_RESULT_2
REG_D1_6	0x158	D1_CALIB_RESULT_3
REG_D1_7	0x15c	D1_CALIB_RESULT_4
REG_D1_8	0x160	D1_CALIB_RESULT_5
REG_D1_9	0x164	D1_CALIB_RESULT_6
REG_D1_A	0x168	D1_CALIB_RESULT_7

Table 9-25 MIPI Rx CSI control registers overview

Name	Address Offset	Description
REG_00	0x000	MODE_CTRL
REG_04	0x004	INTR_CTRL
REG_08	0x008	HDR_CTRL_0
REG_0C	0x00c	HDR_CTRL_1
REG_10	0x010	HDR_CTRL_2
REG_14	0x014	BLC_CTRL
REG_18	0x018	HDR_CTRL_3
REG_1C	0x01c	HDR_CTRL_4
REG_20	0x020	HDR_CTRL_5
REG_24	0x024	HDR_CTRL_6
REG_40	0x040	CSI_STATUS
REG_60	0x060	
REG_70	0x070	CSI_VS_GEN
REG_74	0x074	HDR_DT_CTRL

Table 9-26 MIPI Rx Sub-LVDS control registers overview

Name	Address Offset	Description
REG_00	0x000	MODE_CTRL
REG_04	0x004	SYNC_CODE_0
REG_08	0x008	SYNC_CODE_1
REG_0C	0x00c	SYNC_CODE_2
REG_10	0x010	SYNC_CODE_3
REG_14	0x014	SYNC_CODE_4
REG_18	0x018	SYNC_CODE_5
REG_1C	0x01c	SYNC_CODE_6
REG_20	0x020	SYNC_CODE_7
REG_24	0x024	SYNC_CODE_8
REG_28	0x028	SYNC_CODE_9
REG_2C	0x02c	VS_GEN
REG_30	0x030	LANE_MODE
REG_50	0x050	SYNC_CODE_A
REG_54	0x054	SYNC_CODE_B
REG_58	0x058	HDR_PATTEN_2
REG_60	0x060	HISPI_MODE_CTRL_0
REG_64	0x064	HISPI_MODE_CTRL_1
REG_68	0x068	HISPI_MODE_CTRL_2
REG_6C	0x06c	HISPI_MODE_CTRL_3
REG_70	0x070	HISPI_MODE_CTRL_4
REG_74	0x074	HISPI_MODE_CTRL_5
REG_80	0x080	DBG_SEL

9.2.5 MIPI RxRegister Overview

The first part is the description of MIPI Rx PHY register.

REG_00

Offset Address: 0x000

Bits	Name	Access	Description	Reset
13:0	Reserved			
14	reg_mipirx_pd_ibias	R/W	Power down analog ibias	0x1
15	Reserved			
21:16	reg_mipirx_pd_rxlp	R/W	Power down analog RXLP	0x3f
31:22	Reserved			

REG_04

Offset Address: 0x004

Bits	Name	Access	Description	Reset
15:0	Reserved			
21:16	reg_mipirx_sel_clk_channel	R/W	Analog macro clock lane select	0x0
30:22	Reserved			
31	reg_mipimpll_clk_csi_en	R/W	Gating test clock from mipimpll	0x0

REG_30

Offset Address: 0x030

Bits	Name	Access	Description	Reset
2:0	reg_sensor_phy_mode	R/W	Sensor PHY mode enable select 0: 1C4D 1: 1C2D + 1C2D else: reserved	0x0
31:3	Reserved			

REG_34

Offset Address: 0x034

Bits	Name	Access	Description	Reset
31:0	reg_mipirx_ro_cal0	RO	Analog lane 0 calibration result	

REG_38

Offset Address: 0x038

Bits	Name	Access	Description	Reset
31:0	reg_mipirx_ro_cal1	RO	Analog lane 1 calibration result	

REG_3C

Offset Address: 0x03c

Bits	Name	Access	Description	Reset
31:0	reg_mipirx_ro_cal2	RO	Analog lane 2 calibration result	

REG_40

Offset Address: 0x040

Bits	Name	Access	Description	Reset
31:0	reg_mipirx_ro_cal3	RO	Analog lane 3 calibration result	

REG_44

Offset Address: 0x044

Bits	Name	Access	Description	Reset
31:0	reg_mipirx_ro_cal4	RO	Analog lane 4 calibration result	

REG_48

Offset Address: 0x048

Bits	Name	Access	Description	Reset
31:0	reg_mipirx_ro_cal5	RO	Analog lane 5 calibration result	

REG_80

Offset Address: 0x080

Bits	Name	Access	Description	Reset
0	reg_ad_d0_clk_inv	R/W	AD clock lane0 inverse	0x0
1	reg_ad_d1_clk_inv	R/W	AD clock lane1 inverse	0x0
2	reg_ad_d2_clk_inv	R/W	AD clock lane2 inverse	0x0
3	reg_ad_d3_clk_inv	R/W	AD clock lane3 inverse	0x0
4	reg_ad_d4_clk_inv	R/W	AD clock lane4 inverse	0x0
5	reg_ad_d5_clk_inv	R/W	AD clock lane5 inverse	0x0
31:6	Reserved			

REG_A0

Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
13:0	reg_cam0_vtt	R/W		0x0
15:14	Reserved			
29:16	reg_cam0_vs_str	R/W		0x0
31:30	Reserved			

REG_A4

Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
13:0	reg_cam0_vs_stp	R/W		0x0
15:14	Reserved			
29:16	reg_cam0_htt	R/W		0x0
31:30	Reserved			

REG_A8

Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
13:0	reg_cam0_hs_str	R/W		0x0
15:14	Reserved			
29:16	reg_cam0_hs_stp	R/W		0x0
31:30	Reserved			

REG_AC

Offset Address: 0x0ac

Bits	Name	Access	Description	Reset
0	reg_cam0_vs_pol	R/W		0x0
1	reg_cam0_hs_pol	R/W		0x0
2	reg_cam0_tgen_en	R/W		0x0
31:3	Reserved			

Offset = 0x0A0D0300

REG_00

Offset Address: 0x000

Bits	Name	Access	Description	Reset
1:0	reg_sensor_mode	R/W	Sensor mode select 2'b00: CSI 2'b01: Sub-LVDS & HiSPi 2'b10: SLVSEC	0x0
31:2	Reserved			

REG_04

Offset Address: 0x004

Bits	Name	Access	Description	Reset
2:0	reg_csi_lane_d0_sel	R/W	Data lane 0 select	0x1
3	Reserved			
6:4	reg_csi_lane_d1_sel	R/W	Data lane 1 select	0x2

Bits	Name	Access	Description	Reset
7	Reserved			
10:8	reg_csi_lane_d2_sel	R/W	Data lane 2 select	0x3
11	Reserved			
14:12	reg_csi_lane_d3_sel	R/W	Data lane 3 select	0x4
31:15	Reserved			

REG_08

Offset Address: 0x008

Bits	Name	Access	Description	Reset
2:0	reg_csi_lane_ck_sel	R/W	Clock lane select	0x0
3	Reserved			
4	reg_csi_lane_ck_pnswap	R/W	Clock lane pn swap	0x0
7:5	Reserved			
8	reg_csi_lane_d0_pnswap	R/W	Data lane 0 pn swap	0x0
9	reg_csi_lane_d1_pnswap	R/W	Data lane 1 pn swap	0x0
10	reg_csi_lane_d2_pnswap	R/W	Data lane 2 pn swap	0x0
11	reg_csi_lane_d3_pnswap	R/W	Data lane 3 pn swap	0x0
15:12	Reserved			
23:16	reg_csi_ck_phase	R/W	Clock lane phase	0x0
31:24	Reserved			

REG_0C

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
3:0	reg_deskew_lane_en	R/W	Deskew lane enable 4'h0: No lane 4'h1: 1-lane 4'h3: 2-lane 8'hf: 4-lane	0x0
31:4	Reserved			

REG_20

Offset Address: 0x020

Bits	Name	Access	Description	Reset
0	reg_slvds_inv_en	R/W	Sub-LVDS bit reverse 1'b0: LSB first 1'b1: MSB first	0x1
1	Reserved			
3:2	reg_slvds_bit_mode	R/W	Sub-LVDS bit mode 2'b00: 8-bit 2'b01: 10-bit 2'b10: 12-bit	0x2
7:4	reg_slvds_lane_en	R/W	Sub-LVDS lane enable Set this register to start finding sync code	0x0
15:8	Reserved			
27:16	reg_slvds_sav_1st	R/W	Sub-LVDS sync code 1st symbol	0xfff
31:28	Reserved			

REG_24

Offset Address: 0x024

Bits	Name	Access	Description	Reset
11:0	reg_slvds_sav_2nd	R/W	Sub-LVDS sync code 2nd symbol	0x0
15:12	Reserved			
27:16	reg_slvds_sav_3rd	R/W	Sub-LVDS sync code 3rd symbol	0x0
31:28	Reserved			

REG_D0_0

Offset Address: 0x100

Bits	Name	Access	Description	Reset
0	reg_d0_prbs9_en	R/W	Manual PRBS9 enable	0x0
1	reg_d0_prbs9_clr_err	R/W	PRBS9 clear error	0x0
2	reg_d0_prbs9_source	R/W	PRBS9 source select 1'b0: after sync code shift 1'b1: direct from input	0x0
3	reg_d0_prbs9_stop_when_done	R/W	PRBS9 error count accumalation 1'b0: still count after test time done 1'b1: do not count after test time done	0x0
7:4	Reserved			
15:8	reg_d0_calib_max	R/W	Calibration max step	0x1f
23:16	reg_d0_calib_step	R/W	Calibration one step value	0x1
31:24	reg_d0_calib_pattern	R/W	Calibration golden pattern	0xaa

REG_D0_1

Offset Address: 0x104

Bits	Name	Access	Description	Reset
0	reg_d0_calib_en	R/W	Calibration software enable	0x0
1	reg_d0_calib_source	R/W	Calibration source 1'b0: normal position 1'b1: direct from analog	0x0
2	reg_d0_calib_mode	R/W	Calibration software mode 1'b0: use identical calibration pattern 1'b1: use PRBS9 pattern	0x0
3	reg_d0_calib_ignore	R/W	Ignore calibration command	0x0
31:4	Reserved			

REG_D0_3

Offset Address: 0x10c

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_0	RO	Calibration result phase 0~31	

REG_D0_4

Offset Address: 0x110

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_1	RO	Calibration result phase 32~63	

REG_D0_5

Offset Address: 0x114

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_2	RO	Calibration result phase 64~95	

REG_D0_6

Offset Address: 0x118

Bits	Name	Access	Description	Reset

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_3	RO	Calibration result phase 96~127	

REG_D0_7

Offset Address: 0x11c

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_4	RO	Calibration result phase 128~159	

REG_D0_8

Offset Address: 0x120

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_5	RO	Calibration result phase 160~191	

REG_D0_9

Offset Address: 0x124

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_6	RO	Calibration result phase 192~223	

REG_D0_A

Offset Address: 0x128

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_7	RO	Calibration result phase 224~255	

REG_D1_0

Offset Address: 0x140

Bits	Name	Access	Description	Reset
0	reg_d1_prbs9_en	R/W	Manual PRBS9 enable	0x0
1	reg_d1_prbs9_clr_err	R/W	PRBS9 clear error	0x0
2	reg_d1_prbs9_source	R/W	PRBS9 source select 1'b0: after sync code shift 1'b1: direct from input	0x0
3	reg_d1_prbs9_stop_when_done	R/W	PRBS9 error count accumalation 1'b0: still count after test time done 1'b1: do not count after test time done	0x0
7:4	Reserved			
15:8	reg_d1_calib_max	R/W	Calibration max step	0x1f
23:16	reg_d1_calib_step	R/W	Calibration one step value	0x1
31:24	reg_d1_calib_pattern	R/W	Calibration golden pattern	0xaa

REG_D1_1

Offset Address: 0x144

Bits	Name	Access	Description	Reset
0	reg_d1_calib_en	R/W	Calibration software enable	0x0
1	reg_d1_calib_source	R/W	Calibration source 1'b0: normal position 1'b1: direct from analog	0x0
2	reg_d1_calib_mode	R/W	Calibration software mode 1'b0: use identical calibration pattern 1'b1: use PRBS9 pattern	0x0
3	reg_d1_calib_ignore	R/W	Ignore calibration command	0x0
31:4	Reserved			

REG_D1_3

Offset Address: 0x14c

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_0	RO	Calibration result phase 0~31	

REG_D1_4

Offset Address: 0x150

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_1	RO	Calibration result phase 32~63	

REG_D1_5

Offset Address: 0x154

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_2	RO	Calibration result phase 64~95	

REG_D1_6

Offset Address: 0x158

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_3	RO	Calibration result phase 96~127	

REG_D1_7

Offset Address: 0x15c

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_4	RO	Calibration result phase 128~159	

REG_D1_8

Offset Address: 0x160

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_5	RO	Calibration result phase 160~191	

REG_D1_9

Offset Address: 0x164

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_6	RO	Calibration result phase 192~223	

REG_D1_A

Offset Address: 0x168

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_7	RO	Calibration result phase 224~255	

REG_D2_0

Offset Address: 0x180

Bits	Name	Access	Description	Reset
0	reg_d2_prbs9_en	R/W	Manual PRBS9 enable	0x0
1	reg_d2_prbs9_clr_err	R/W	PRBS9 clear error	0x0
2	reg_d2_prbs9_source	R/W	PRBS9 source select 1'b0: after sync code shift 1'b1: direct from input	0x0
3	reg_d2_prbs9_stop_when_done	R/W	PRBS9 error count accumalation 1'b0: still count after test time done 1'b1: do not count after test time done	0x0
7:4	Reserved			
15:8	reg_d2_calib_max	R/W	Calibration max step	0x1f

Bits	Name	Access	Description	Reset
23:16	reg_d2_calib_step	R/W	Calibration one step value	0x1
31:24	reg_d2_calib_pattern	R/W	Calibration golden pattern	0xaa

REG_D2_1

Offset Address: 0x184

Bits	Name	Access	Description	Reset
0	reg_d2_calib_en	R/W	Calibration software enable	0x0
1	reg_d2_calib_source	R/W	Calibration source 1'b0: normal position 1'b1: direct from analog	0x0
2	reg_d2_calib_mode	R/W	Calibration software mode 1'b0: use identical calibration pattern 1'b1: use PRBS9 pattern	0x0
3	reg_d2_calib_ignore	R/W	Ignore calibration command	0x0
31:4	Reserved			

REG_D2_3

Offset Address: 0x18c

Bits	Name	Access	Description	Reset
31:0	reg_d2_skew_calib_result_0	RO	Calibration result phase 0~31	

REG_D2_4

Offset Address: 0x190

Bits	Name	Access	Description	Reset
31:0	reg_d2_skew_calib_result_1	RO	Calibration result phase 32~63	

REG_D2_5

Offset Address: 0x194

Bits	Name	Access	Description	Reset
31:0	reg_d2_skew_calib_result_2	RO	Calibration result phase 64~95	

REG_D2_6

Offset Address: 0x198

Bits	Name	Access	Description	Reset
31:0	reg_d2_skew_calib_result_3	RO	Calibration result phase 96~127	

REG_D2_7

Offset Address: 0x19c

Bits	Name	Access	Description	Reset
31:0	reg_d2_skew_calib_result_4	RO	Calibration result phase 128~159	

REG_D2_8

Offset Address: 0x1a0

Bits	Name	Access	Description	Reset
31:0	reg_d2_skew_calib_result_5	RO	Calibration result phase 160~191	

REG_D2_9

Offset Address: 0x1a4

Bits	Name	Access	Description	Reset
31:0	reg_d2_skew_calib_result_6	RO	Calibration result phase 192~223	

REG_D2_A

Offset Address: 0x1a8

Bits	Name	Access	Description	Reset
31:0	reg_d2_skew_calib_result_7	RO	Calibration result phase 224~255	

REG_D3_0

Offset Address: 0x1c0

Bits	Name	Access	Description	Reset
0	reg_d3_prbs9_en	R/W	Manual PRBS9 enable	0x0
1	reg_d3_prbs9_clr_err	R/W	PRBS9 clear error	0x0
2	reg_d3_prbs9_source	R/W	PRBS9 source select 1'b0: after sync code shift 1'b1: direct from input	0x0
3	reg_d3_prbs9_stop_when_done	R/W	PRBS9 error count accumalation 1'b0: still count after test time done 1'b1: do not count after test time done	0x0
7:4	Reserved			
15:8	reg_d3_calib_max	R/W	Calibration max step	0x1f
23:16	reg_d3_calib_step	R/W	Calibration one step value	0x1
31:24	reg_d3_calib_pattern	R/W	Calibration golden pattern	0xaa

REG_D3_1

Offset Address: 0x1c4

Bits	Name	Access	Description	Reset
0	reg_d3_calib_en	R/W	Calibration software enable	0x0
1	reg_d3_calib_source	R/W	Calibration source 1'b0: normal position 1'b1: direct from analog	0x0
2	reg_d3_calib_mode	R/W	Calibration software mode 1'b0: use identical calibration pattern 1'b1: use PRBS9 pattern	0x0
3	reg_d3_calib_ignore	R/W	Ignore calibration command	0x0
31:4	Reserved			

REG_D3_3

Offset Address: 0x1cc

Bits	Name	Access	Description	Reset
31:0	reg_d3_skew_calib_result_0	RO	Calibration result phase 0~31	

REG_D3_4

Offset Address: 0x1d0

Bits	Name	Access	Description	Reset
31:0	reg_d3_skew_calib_result_1	RO	Calibration result phase 32~63	

REG_D3_5

Offset Address: 0x1d4

Bits	Name	Access	Description	Reset
31:0	reg_d3_skew_calib_result_2	RO	Calibration result phase 64~95	

REG_D3_6

Offset Address: 0x1d8

Bits	Name	Access	Description	Reset
31:0	reg_d3_skew_calib_result_3	RO	Calibration result phase 96~127	

REG_D3_7

Offset Address: 0x1dc

Bits	Name	Access	Description	Reset
31:0	reg_d3_skew_calib_result_4	RO	Calibration result phase 128~159	

REG_D3_8

Offset Address: 0x1e0

Bits	Name	Access	Description	Reset
31:0	reg_d3_skew_calib_result_5	RO	Calibration result phase 160~191	

REG_D3_9

Offset Address: 0x1e4

Bits	Name	Access	Description	Reset
31:0	reg_d3_skew_calib_result_6	RO	Calibration result phase 192~223	

REG_D3_A

Offset Address: 0x1e8

Bits	Name	Access	Description	Reset
31:0	reg_d3_skew_calib_result_7	RO	Calibration result phase 224~255	

Offset = 0x0A0D0600

REG_00

Offset Address: 0x000

Bits	Name	Access	Description	Reset
1:0	reg_sensor_mode	R/W	Sensor mode select 2'b00: CSI 2'b01: Sub-LVDS & HiSPi 2'b10: SLVSEC	0x0
31:2	Reserved			

REG_04

Offset Address: 0x004

Bits	Name	Access	Description	Reset
1:0	reg_csi_lane_d0_sel	R/W	Data lane 0 select	0x1
3:2	Reserved			
5:4	reg_csi_lane_d1_sel	R/W	Data lane 1 select	0x2
31:6	Reserved			

REG_08

Offset Address: 0x008

Bits	Name	Access	Description	Reset
1:0	reg_csi_lane_ck_sel	R/W	Clock lane select	0x0
3:2	Reserved			
4	reg_csi_lane_ck_pnswap	R/W	Clock lane pn swap	0x0
7:5	Reserved			
8	reg_csi_lane_d0_pnswap	R/W	Data lane 0 pn swap	0x0
9	reg_csi_lane_d1_pnswap	R/W	Data lane 1 pn swap	0x0
15:10	Reserved			
23:16	reg_csi_ck_phase	R/W	Clock lane phase	0x0
31:24	Reserved			

REG_0C

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
1:0	reg_deskew_lane_en	R/W	Deskew lane enable 2'h0: No lane 2'h1: 1-lane 2'h3: 2-lane	0x0
31:2	Reserved			

REG_20

Offset Address: 0x020

Bits	Name	Access	Description	Reset
0	reg_slvds_inv_en	R/W	Sub-LVDS bit reverse 1'b0: LSB first 1'b1: MSB first	0x1
1	Reserved			
3:2	reg_slvds_bit_mode	R/W	Sub-LVDS bit mode 2'b00: 8-bit 2'b01: 10-bit 2'b10: 12-bit	0x2
5:4	reg_slvds_lane_en	R/W	Sub-LVDS lane enable Set this register to start finding sync code	0x0
31:6	Reserved			

REG_D0_0

Offset Address: 0x100

Bits	Name	Access	Description	Reset
0	reg_d0_prbs9_en	R/W	Manual PRBS9 enable	0x0
1	reg_d0_prbs9_clr_err	R/W	PRBS9 clear error	0x0
2	reg_d0_prbs9_source	R/W	PRBS9 source select 1'b0: after sync code shift 1'b1: direct from input	0x0
3	reg_d0_prbs9_stop_when_done	R/W	PRBS9 error count accumalation 1'b0: still count after test time done 1'b1: do not count after test time done	0x0
7:4	Reserved			
15:8	reg_d0_calib_max	R/W	Calibration max step	0x1f
23:16	reg_d0_calib_step	R/W	Calibration one step value	0x1
31:24	reg_d0_calib_pattern	R/W	Calibration golden pattern	0xaa

REG_D0_1

Offset Address: 0x104

Bits	Name	Access	Description	Reset
0	reg_d0_calib_en	R/W	Calibration software enable	0x0
1	reg_d0_calib_source	R/W	Calibration source 1'b0: normal position 1'b1: direct from analog	0x0
2	reg_d0_calib_mode	R/W	Calibration software mode 1'b0: use identical calibration pattern 1'b1: use PRBS9 pattern	0x0
3	reg_d0_calib_ignore	R/W	Ignore calibration command	0x0
31:4	Reserved			

REG_D0_3

Offset Address: 0x10c

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_0	RO	Calibration result phase 0~31	

REG_D0_4

Offset Address: 0x110

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_1	RO	Calibration result phase 32~63	

REG_D0_5

Offset Address: 0x114

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_2	RO	Calibration result phase 64~95	

REG_D0_6

Offset Address: 0x118

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_3	RO	Calibration result phase 96~127	

REG_D0_7

Offset Address: 0x11c

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_4	RO	Calibration result phase 128~159	

REG_D0_8

Offset Address: 0x120

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_5	RO	Calibration result phase 160~191	

REG_D0_9

Offset Address: 0x124

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_6	RO	Calibration result phase 192~223	

REG_D0_A

Offset Address: 0x128

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_7	RO	Calibration result phase 224~255	

REG_D1_0

Offset Address: 0x140

Bits	Name	Access	Description	Reset
0	reg_d1_prbs9_en	R/W	Manual PRBS9 enable	0x0
1	reg_d1_prbs9_clr_err	R/W	PRBS9 clear error	0x0
2	reg_d1_prbs9_source	R/W	PRBS9 source select 1'b0: after sync code shift 1'b1: direct from input	0x0
3	reg_d1_prbs9_stop_when_done	R/W	PRBS9 error count accumalation 1'b0: still count after test time done 1'b1: do not count after test time done	0x0

Bits	Name	Access	Description	Reset
7:4	Reserved			
15:8	reg_d1_calib_max	R/W	Calibration max step	0x1f
23:16	reg_d1_calib_step	R/W	Calibration one step value	0x1
31:24	reg_d1_calib_pattern	R/W	Calibration golden pattern	0xaa

REG_D1_1

Offset Address: 0x144

Bits	Name	Access	Description	Reset
0	reg_d1_calib_en	R/W	Calibration software enable	0x0
1	reg_d1_calib_source	R/W	Calibration source 1'b0: normal position 1'b1: direct from analog	0x0
2	reg_d1_calib_mode	R/W	Calibration software mode 1'b0: use identical calibration pattern 1'b1: use PRBS9 pattern	0x0
3	reg_d1_calib_ignore	R/W	Ignore calibration command	0x0
31:4	Reserved			

REG_D1_3

Offset Address: 0x14c

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_0	RO	Calibration result phase 0~31	

REG_D1_4

Offset Address: 0x150

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_1	RO	Calibration result phase 32~63	

REG_D1_5

Offset Address: 0x154

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_2	RO	Calibration result phase 64~95	

REG_D1_6

Offset Address: 0x158

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_3	RO	Calibration result phase 96~127	

REG_D1_7

Offset Address: 0x15c

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_4	RO	Calibration result phase 128~159	

REG_D1_8

Offset Address: 0x160

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_5	RO	Calibration result phase 160~191	

REG_D1_9

Offset Address: 0x164

Bits	Name	Access	Description	Reset

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_6	RO	Calibration result phase 192~223	

REG_D1_A

Offset Address: 0x168

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_7	RO	Calibration result phase 224~255	

The second part is MIPI Rx CSI controller register.

REG_00

Offset Address: 0x000

Bits	Name	Access	Description	Reset
2:0	reg_csi_lane_mode	R/W	Lane mode 3'b000: 1-lane 3'b001: 2-lane 3'b011: 4-lane 3'b111: 8-lane	0x0
3	reg_csi_ignore_ecc	R/W	Ignore ecc result 1'b0: normal 1'b1: still processing even ecc error	0x0
4	reg_csi_vc_check	R/W	VC check enable 1'b0: do not check VC 1'b1: only process packets that meet vc_set[3:0]	0x0
7:5	Reserved			
11:8	reg_csi_vc_set	R/W	VC set only use when reg_csi_vc_check assert	0x0
12	reg_csi_line_start_sent	R/W	LS and LE packet sent 1'b0: create hsync signal by controller 1'b1: use LS and LE to create hsync signal	0x0
31:13	Reserved			

REG_04

Offset Address: 0x004

Bits	Name	Access	Description	Reset
7:0	reg_csi_intr_mask	R/W	Interrupt mask control	0x0
15:8	reg_csi_intr_clr	W1T	Interrupt clear	
16	reg_csi_hdr_en	R/W	HDR mode enable	0x0
17	reg_csi_hdr_mode	R/W	HDR mode selection 1'b0: HDR VC mode 1'b1: HDR ID mode	0x0
18	reg_csi_id_rm_else	R/W	Remove non reconized ID line 1'b0: dont remove 1'b1: remove	0x1
19	reg_csi_id_rm_ob	R/W	Remove ob line 1'b0: don't remove 1'b1: remove	0x1
31:20	Reserved			

REG_08

Offset Address: 0x008

Bits	Name	Access	Description	Reset
15:0	reg_csi_n0_ob_lef	R/W	ID for LEF ob n0	0x221
31:16	reg_csi_n0_ob_sef	R/W	ID for SEF ob n0	0x222

REG_0C

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
15:0	reg_csi_n0_lef	R/W	ID for LEF active n0	0x241
31:16	reg_csi_n1_ob_lef	R/W	ID for LEF ob n1	0x231

REG_10

Offset Address: 0x010

Bits	Name	Access	Description	Reset
15:0	reg_csi_n1_ob_sef	R/W	ID for SEF ob n1	0x232
31:16	reg_csi_n1_lef	R/W	ID for LEF active n1	0x251

REG_14

Offset Address: 0x014

Bits	Name	Access	Description	Reset
5:0	reg_csi_blc_dt	R/W	Data type for optical black line	0x37
7:6	Reserved			
8	reg_csi_blc_en	R/W	Optical black line mode enable	0x0
11:9	Reserved			
14:12	reg_csi_blc_format_set	R/W	Optical black line data format set 3'd0: YUV422 8bit 3'd1: YUV422 10bit 3'd2: RAW8 3'd3: RAW10 3'd4: RAW12 3'd5: RAW16 else: reserved	0x2
31:15	Reserved			

REG_18

Offset Address: 0x018

Bits	Name	Access	Description	Reset
3:0	reg_csi_vc_map_ch00	R/W	VC mapping to ISP channel 00	0x0
7:4	reg_csi_vc_map_ch01	R/W	VC mapping to ISP channel 01	0x1
11:8	reg_csi_vc_map_ch10	R/W	VC mapping to ISP channel 10	0x2
15:12	reg_csi_vc_map_ch11	R/W	VC mapping to ISP channel 11	0x3
31:16	Reserved			

REG_1C

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
15:0	reg_csi_n0_sef	R/W	ID for SEF active n0	0x242
31:16	reg_csi_n1_sef	R/W	ID for SEF active n1	0x252

REG_20

Offset Address: 0x020

Bits	Name	Access	Description	Reset
15:0	reg_csi_n0_sef2	R/W	ID for SEF2 active n0	0x244

Bits	Name	Access	Description	Reset
31:16	reg_csi_n1_sef2	R/W	ID for SEF2 active n1	0x254

REG_24

Offset Address: 0x024

Bits	Name	Access	Description	Reset
15:0	reg_csi_n0_ob_sef2	R/W	ID for SEF2 ob n0	0x224
31:16	reg_csi_n1_ob_sef2	R/W	ID for SEF2 ob n1	0x234

REG_40

Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	reg_csi_ecc_no_error	RO	ECC no error	
1	reg_csi_ecc_corrected_error	RO	ECC corrected error	
2	reg_csi_ecc_error	RO	ECC error	
3	Reserved			
4	reg_csi_crc_error	RO	CRC error	
5	reg_csi_wc_error	RO	WC error	
7:6	Reserved			
8	reg_csi_fifo_full	RO	CSI FIFO full	
15:9	Reserved			
21:16	reg_csi_decode_format	RO	CSI decode format from header bit[0]: YUV422 8bit bit[1]: YUV422 10bit bit[2]: RAW8 bit[3]: RAW10 bit[4]: RAW12 bit[5]: RAW16	
31:22	Reserved			

REG_60

Offset Address: 0x060

Bits	Name	Access	Description	Reset
7:0	reg_csi_intr_status	RO	Interrupt status bit[0]: ecc error bit[1]: crc error bit[2]: hdr id error bit[3]: word count error bit[4]: fifo full	
31:8	Reserved			

REG_70

Offset Address: 0x070

Bits	Name	Access	Description	Reset
1:0	reg_csi_vs_gen_mode	R/W	2'b00: vs gen by FS 2'b01: vs gen by FE else: vs gen by FS & FE	0x2
3:2	Reserved			
4	reg_csi_vs_gen_by_vcset	R/W	Vsync generation setting 1'b0: generated by all vc short packet 1'b1: only generated by indicated vc short packet	0x0

Bits	Name	Access	Description	Reset
31:5	Reserved			

REG_74

Offset Address: 0x074

Bits	Name	Access	Description	Reset
0	reg_csi_hdr_dt_mode	R/W	CSI HDR DT mode enable	0x0
3:1	Reserved			
9:4	reg_csi_hdr_dt_format	R/W	CSI HDR DT mode video format data type	0x0
11:10	Reserved			
17:12	reg_csi_hdr_dt_lef	R/W	CSI HDR DT mode LEF data type	0x0
19:18	Reserved			
25:20	reg_csi_hdr_dt_sef	R/W	CSI HDR DT mode SEF data type	0x0
31:26	Reserved			

The third part is MIPI Rx Sub-LVDS control registers

REG_00

Offset Address: 0x000

Bits	Name	Access	Description	Reset
7:0	reg_slvds_enable	R/W	Sub-LVDS lane enable for each lane	0x0
9:8	reg_slvds_bit_mode	R/W	Sub-LVDS bit mode 2'b00: 8-bit 2'b01: 10-bit 2'b10: 12-bit	0x2
10	reg_slvds_data_reverse	R/W	Sub-LVDS data packet bit inverse	0x0
11	Reserved			
12	reg_slvds_hdr_mode	R/W	Sub-LVDS HDR mode enable	0x0
13	reg_slvds_hdr_pattern	R/W	Sub-LVDS HDR pattern mode 1'b0: pattern 1 1'b1: pattern 2	0x0
31:14	Reserved			

REG_04

Offset Address: 0x004

Bits	Name	Access	Description	Reset
11:0	reg_slvds_sync_1st	R/W	Sub-LVDS SYNC code 1st word	0xFFF
15:12	Reserved			
27:16	reg_slvds_sync_2nd	R/W	Sub-LVDS SYNC code 2nd word	0x000
31:28	Reserved			

REG_08

Offset Address: 0x008

Bits	Name	Access	Description	Reset
11:0	reg_slvds_sync_3rd	R/W	Sub-LVDS SYNC code 3rd word	0x000
15:12	Reserved			
27:16	reg_slvds_norm_bk_sav	R/W	Normal mode blanking SAV	0xAB0
31:28	Reserved			

REG_0C

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
11:0	reg_slvds_norm_bk_eav	R/W	Normal mode blanking EAV	0xB60
15:12	Reserved			
27:16	reg_slvds_norm_sav	R/W	Normal mode active SAV	0x800
31:28	Reserved			

REG_10

Offset Address: 0x010

Bits	Name	Access	Description	Reset
11:0	reg_slvds_norm_eav	R/W	Normal mode active EAV	0x9D0
15:12	Reserved			
27:16	reg_slvds_n0_bk_sav	R/W	HDR mode n0 blanking SAV	0x2B0
31:28	Reserved			

REG_14

Offset Address: 0x014

Bits	Name	Access	Description	Reset
11:0	reg_slvds_n0_bk_eav	R/W	HDR mode n0 blanking EAV	0x360
15:12	Reserved			
27:16	reg_slvds_n1_bk_sav	R/W	HDR mode n1 blanking SAV	0x6B0
31:28	Reserved			

REG_18

Offset Address: 0x018

Bits	Name	Access	Description	Reset
11:0	reg_slvds_n1_bk_eav	R/W	HDR mode n1 blanking EAV	0x760
15:12	Reserved			
27:16	reg_slvds_n0_lef_sav	R/W	Sub-LVDS mode: n0 long exposure sav Sub-LVDS 12-bit LEF SAV n0 (801) Sub-LVDS 10-bit LEF SAV n0 (004) HiSPi P-SP mode: SOL T1 (800)	0x801
31:28	Reserved			

REG_1C

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
11:0	reg_slvds_n0_lef_eav	R/W	Sub-LVDS mode: n0 long exposure eav Sub-LVDS 12-bit LEF EAV n0 (9D1) Sub-LVDS 10-bit LEF EAV n0 (1D4) HiSPi P-SP mode: EOL T1 (A00)	0x9D1
15:12	Reserved			
27:16	reg_slvds_n0_sef_sav	R/W	Sub-LVDS mode: n0 short exposure sav Sub-LVDS 12-bit SEF SAV n0 (802) Sub-LVDS 10-bit SEF SAV n0 (008) HiSPi P-SP mode: SOL T2 (820)	0x802
31:28	Reserved			

REG_20

Offset Address: 0x020

Bits	Name	Access	Description	Reset

Bits	Name	Access	Description	Reset
11:0	reg_slvds_n0_sef_eav	R/W	Sub-LVDS mode: n0 short exposure eav Sub-LVDS 12-bit SEF EAV n0 (9D2) Sub-LVDS 10-bit SEF EAV n0 (1d8) HiSPi P-SP mode: EOL T2 (A20)	0x9D2
15:12	Reserved			
27:16	reg_slvds_n1_lef_sav	R/W	Sub-LVDS mode: n1 long exposure sav Sub-LVDS 12-bit LEF SAV n1 (C01) Sub-LVDS 10-bit LEF SAV n1 (404) HiSPi P-SP mode: SOF T1 (C00)	0xC01
31:28	Reserved			

REG_24

Offset Address: 0x024

Bits	Name	Access	Description	Reset
11:0	reg_slvds_n1_lef_eav	R/W	Sub-LVDS mode: n1 long exposure eav Sub-LVDS 12-bit LEF EAV n1 (DD1) Sub-LVDS 10-bit LEF EAV n1 (5D4) HiSPi P-SP mode: EOF T1 (E00)	0xDD1
15:12	Reserved			
27:16	reg_slvds_n1_sef_sav	R/W	Sub-LVDS mode: n1 short exposure sav Sub-LVDS 12-bit SEF SAV n1 (C02) Sub-LVDS 10-bit SEF SAV n1 (408) HiSPi P-SP mode: SOF T2 (C20)	0xC02
31:28	Reserved			

REG_28

Offset Address: 0x028

Bits	Name	Access	Description	Reset
11:0	reg_slvds_n1_sef_eav	R/W	Sub-LVDS mode: n1 short exposure eav Sub-LVDS 12-bit SEF EAV n1 (DD2) Sub-LVDS 10-bit SEF EAV n1 (5D8) HiSPi P-SP mode: EOF T2 (E20)	0xDD2
31:12	Reserved			

REG_2C

Offset Address: 0x02c

Bits	Name	Access	Description	Reset
11:0	reg_vs_gen_sync_code	R/W	vs generate sync code value using scenario: HiSPi P-SP HDR	0xC00
12	reg_vs_gen_by_sync_code	R/W	vs generate by identical sync code using scenario: HiSPi P-SP HDR	0x0
31:13	Reserved			

REG_30

Offset Address: 0x030

Bits	Name	Access	Description	Reset
2:0	reg_slvds_lane_mode	R/W	Sub-LVDS lane mode 2'b0: 1-lane 2'b1: 2-lane 2'b3: 4-lane 2'b7: 8-lane	0x3
3	Reserved			
11:4	reg_slvds_sync_source	R/W	Sub-LVDS output sync source select	0x1

Bits	Name	Access	Description	Reset
31:12	Reserved			

REG_50

Offset Address: 0x050

Bits	Name	Access	Description	Reset
11:0	reg_slvds_n0_lsef_sav	R/W	SAV for n0 long & short exposure both exist line only used for pattern 2	0x803
15:12	Reserved			
27:16	reg_slvds_n0_lsef_eav	R/W	EAV for n0 long & short exposure both exist line only used for pattern 2	0x9D3
31:28	Reserved			

REG_54

Offset Address: 0x054

Bits	Name	Access	Description	Reset
11:0	reg_slvds_n1_lsef_sav	R/W	SAV for n1 long & short exposure both exist line only used for pattern 2	0xC03
15:12	Reserved			
27:16	reg_slvds_n1_lsef_eav	R/W	EAV for n1 long & short exposure both exist line only used for pattern 2	0xDD3
31:28	Reserved			

REG_58

Offset Address: 0x058

Bits	Name	Access	Description	Reset
13:0	reg_slvds_hdr_p2_hsize	R/W	Hsize for pattern 2	0xF0
15:14	Reserved			
29:16	reg_slvds_hdr_p2_hblank	R/W	Hblank size for pattern 2	0x14
31:30	Reserved			

REG_60

Offset Address: 0x060

Bits	Name	Access	Description	Reset
0	reg_hispi_mode	R/W	HiSPi mode enable 1'b0: Sub-LVDS 1'b1: HiSPi	0x0
1	reg_hispi_use_hsize	R/W	HiSPi DE de-assert by register count	0x0
3:2	Reserved			
4	reg_hispi_hdr_psp_mode	R/W	HiSPi P-SP HDR mode enable	0x0
31:5	Reserved			

REG_64

Offset Address: 0x064

Bits	Name	Access	Description	Reset
11:0	reg_hispi_norm_sof	R/W	HiSPi SOF sync code	0xC00
15:12	Reserved			
27:16	reg_hispi_norm_eof	R/W	HiSPi EOF sync code	0xE00

Bits	Name	Access	Description	Reset
31:28	Reserved			

REG_68

Offset Address: 0x068

Bits	Name	Access	Description	Reset
11:0	reg_hispi_hdr_t1_sof	R/W	HiSPi HDR T1 SOF	0xC00
15:12	Reserved			
27:16	reg_hispi_hdr_t1_eof	R/W	HiSPi HDR T1 EOF	0xE00
31:28	Reserved			

REG_6C

Offset Address: 0x06c

Bits	Name	Access	Description	Reset
11:0	reg_hispi_hdr_t1_sol	R/W	HiSPi HDR T1 SOL	0x800
15:12	Reserved			
27:16	reg_hispi_hdr_t1_eol	R/W	HiSPi HDR T1 EOL	0xA00
31:28	Reserved			

REG_70

Offset Address: 0x070

Bits	Name	Access	Description	Reset
11:0	reg_hispi_hdr_t2_sof	R/W	HiSPi HDR T2 SOF	0xC20
15:12	Reserved			
27:16	reg_hispi_hdr_t2_eof	R/W	HiSPi HDR T2 EOF	0xE20
31:28	Reserved			

REG_74

Offset Address: 0x074

Bits	Name	Access	Description	Reset
11:0	reg_hispi_hdr_t2_sol	R/W	HiSPi HDR T2 SOL	0x820
15:12	Reserved			
27:16	reg_hispi_hdr_t2_eol	R/W	HiSPi HDR T2 EOL	0xA20
31:28	Reserved			

REG_80

Offset Address: 0x080

Bits	Name	Access	Description	Reset
7:0	reg_dbg_sel	R/W	Debug signal select	0x0
31:8	Reserved			

10 ISP

10.1 Function Overview

Image signal processor (ISP) optimizes the image captured by sensor, including 3A (automatic exposure (AE), automatic white balance (AWB), automatic focus (AF)), black level correction (BLC), defect pixel correction (DPC), fix pattern noise(FPN), high dynamic range image processing (HDR), Bayer domain noise reduction (BNR), and de mosaic (CFA), gamma correction, Dehaze, color space convert (CSC), image sharpen, time domain noise reduction(3DNR), brightness noise reduction(YNR), color noise reduction(CNR), hsv space conversion(HSV), etc. The specifications it supports are as follows.

- Support black level noise reduction
- Support defect pixel correction
- Support Bayer noise reduction
- Support demosaic processing
- Support purple edge correction
- Support gamma correction
- Support automatic white balance
- Support auto exposure
- Support auto focus
- Support lens shading correction
- Support automatic dehaze
- Support image sharpen
- Support time domain noise reduction
- Support 8 / 10 / 12 bits of Bayer input data (maximum 12 bits)
- The maximum image resolution is 2880x1620@20fps linear input
- The maximum image resolution is 2880x1620@20fps output
- The minimum horizontal blanking area is 72 pixels
- The minimum vertical blanking area is 48 rows
- Support IR sensor input

10.2 Overview

10.2.1 Function Block Diagram

As shown in Figure 10-1, ISP can be roughly divided into pre_raw_fe_top、pre_raw_be_top、raw_top、rgb_top, and yuv_ top according to the format of data processing. The pre_raw_fe_top has two sets that support dual-camera input, while the pre_raw_be_top is used to process statistical data of pre_raw_fe_top dual-camera input separately. The three main modules, raw_top, rgb_top, and yuv_top, can also be collectively referred to as post_raw. Figures 10-2, 10-3, 10-4, and 10-5 are detailed block diagrams of the four main modules. As the ISP supports dual-camera input, two sets of modules in pre_raw_fe_top are responsible for receiving dual-camera data. The image data from the two sensors are then processed separately in pre_raw_be_top and post_raw.

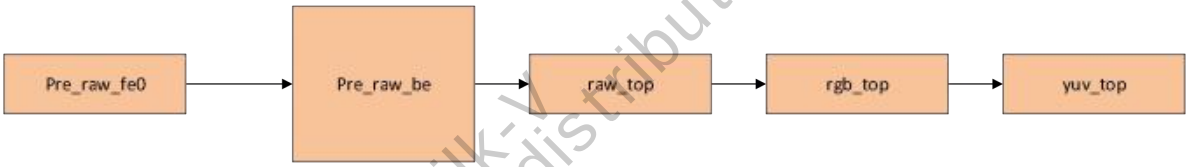


Figure 10- 1 ISP Overall Structure Diagram

The following figure shows the basic module diagram of pre_raw_fe and pre_raw_be. The CSI_BRG of pre_raw_fe receives signals from the sensor side, and the data stream is divided into two paths. One path is directly stored in DRAM or transmitted to pre_raw_be, while the other path is cropped to the desired processing size, and then RGBMAP statistics and WBG correction are performed, and the data is stored in DRAM.

The input data of pre_raw_be is the raw data sent by pre_raw_fe through DRAM or direct transmission. After being processed by crop, BLC, and DPC, one path is directly sent to post_raw for processing of RGB Bayer data, or it is first stored in DRAM and then extracted by post_raw from DRAM. The AF statistics data is directly written to DRAM.

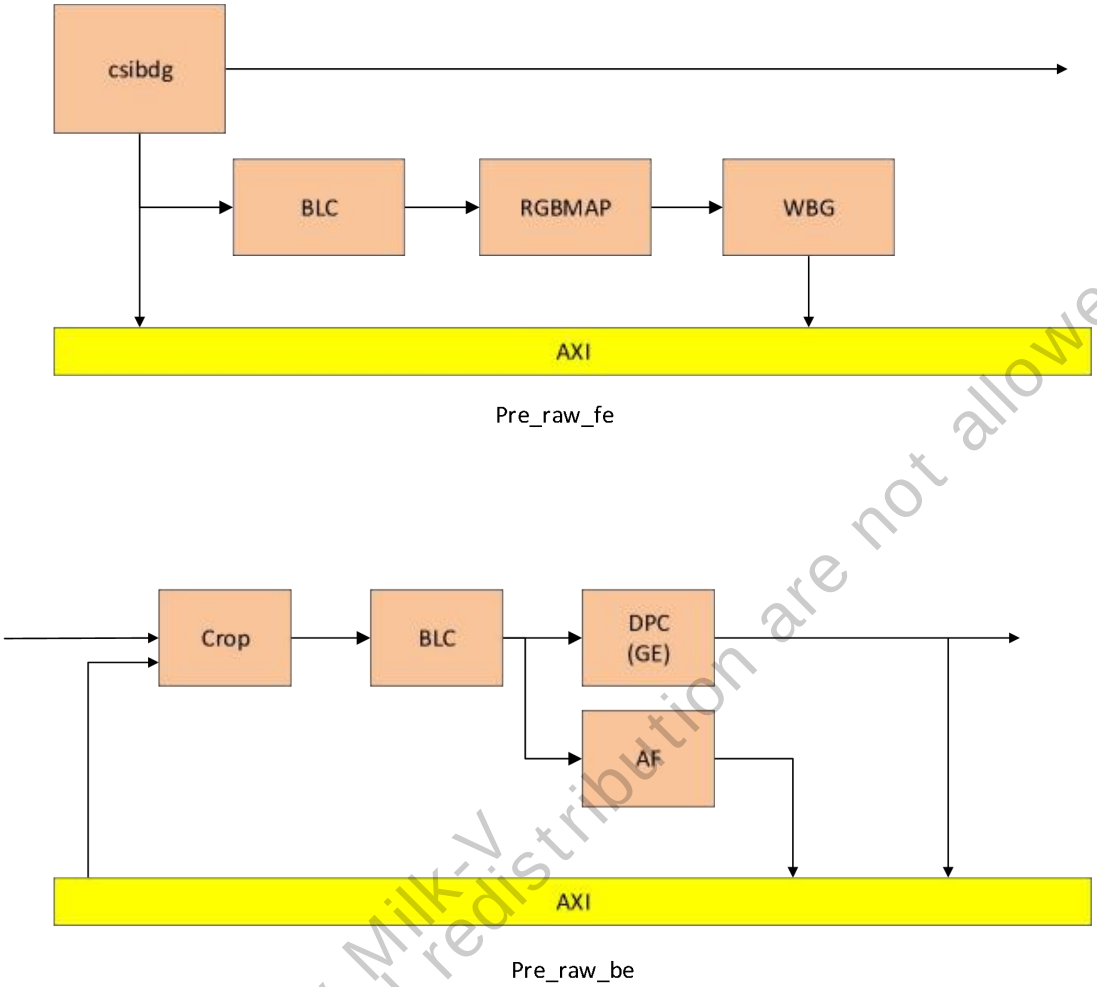


Figure 10- 2 Pre_raw_fe and Pre_raw_be Module Diagram

The following is the basic block diagram of the raw_top module. The long and short exposure data are processed by modules such as CROP, BNR, LSC, CFA, RGBCAC, and LCAC to generate RGB data, which is then sent to rgb_top. The parallel statistical processing is divided into two paths: one is processed by AE and GMS after LSC module processing and the data is stored in DRAM, and the other is processed by WBG and then processed by LMAP and stored in DRAM for reference in subsequent WDR processing.

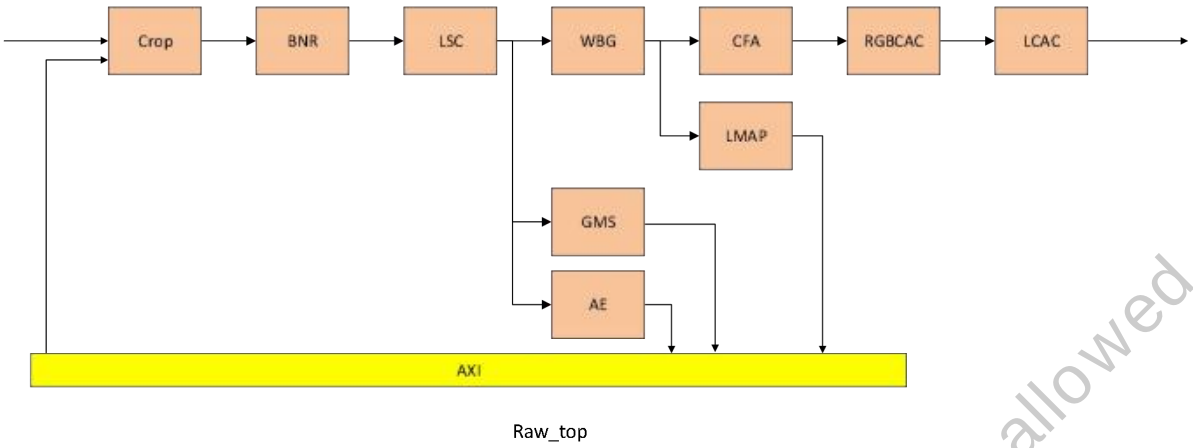


Figure 10- 3 raw_top Module Diagram

The following is the basic module diagram of rgb_top. Its input is the processed RGB data from raw_top. The data is processed through modules such as CCM, HDR (Fusion+LTM), User gamma, Gamma, Dehaze, and CLUT, and then converted to the YUV domain by the RGB2YUV (CSC) module. The YUV data is then transferred to YUV_TOP. The other path is the statistical data path, which is processed by Hist_v after CCM processing and then stored in the DRAM.

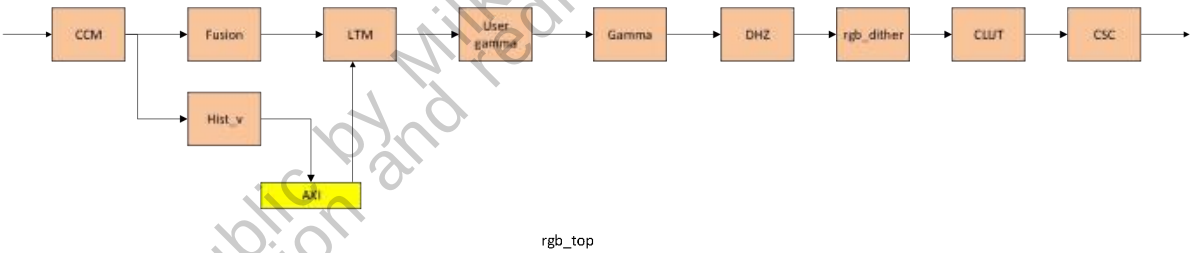


Figure 10- 4 rgb_top Module Diagram

The first module in yuv_top is PRE_EE which does the first round of edge enhancement. 3DNR then works on temporal domain for noise reduction. The brightness and color information are then separately processed in spatial domain using YUV422 format for noise reduction (YNR, CNR). After brightness noise reduction, it will go through SHARPPEN(EE) and then the brightness and chroma information will be combined for DCI, LDCI, and CA processing. The brightness and chroma information will then be split again. The brightness goes through the Ycurve module while the chroma information goes through the CA2 module. After passing through the image size cropping module (Crop) once more, the entire ISP image processing process is complete.

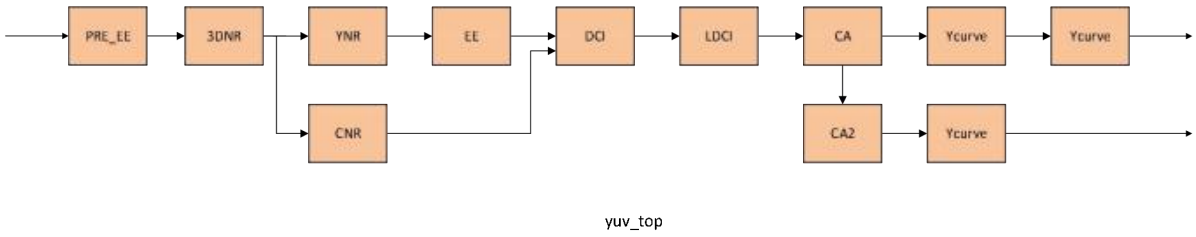


Figure 10- 5 yuv_top Module Diagram

10.2.2 Working Mode

- The maximum 12 bit Bayer input is supported. When the input is less than 12 bit, the lower bit will be 0
  Raw 8 = {data_in[7:0],4' b0}
  Raw 10 = {data_in[9:0], 2' b0}
- Support any RG, GB order interchange
- Support IR sensor
- Support two in one WDR
- Support single brightness component mode (abandon UV data, output in pure brightness mode))

10.3 ISP Interruption System

10.3.1 Function Description

Hardware interrupt events of ISP are detailed in Table 10-1.

Table 10- 1 Interrupt Indication Register

地址	状态位 1：有中断 0：无中断	清除位 (写 1 清成 0)	描述
0x0A07_0000	bit[29]	bit[29]	post_raw register update completed (shadow update done) interrupt
	bit[24]	bit[24]	pre_raw_be channel 0 update completed (shadow update done)intertupt
	bit[19]	bit[19]	pre_raw_fe0 channel 3 update completed (shadow update

地址	状态位 1 : 有中断 0 : 无中断	清除位 (写 1 清成 0)	描述
			done)interrupt
	bit[18]	bit[18]	pre_raw_fe0 channel 2 register update completed (shadow update done) interrupt
	bit[17]	bit[17]	pre_raw_fe0 channel 1 register update completed (shadow update done) interrupt
	bit[16]	bit[16]	pre_raw_fe0 channel 0 register update completed (shadow update done) interrupt
	bit[10]	bit[10]	post_raw frame done interrupt
	bit[8]	bit[8]	pre_raw_be channel 0 frame done interrupt
	bit[3]	bit[3]	pre_raw_fe0 channel 3 frame done interrupt
	bit[2]	bit[2]	pre_raw_fe0 channel 2 frame done interrupt
	bit[1]	bit[1]	pre_raw_fe0 channel 1 frame done interrupt
	bit[0]	bit[0]	pre_raw_fe0 channel 0 frame done interrupt
0x0A07_0008	bit[16]	bit[16]	dma error interrupt
	bit[14]	bit[14]	post_raw transmission specified line completed interrupt
	bit[11]	bit[11]	pre_raw_fe0 transmission specified line completed interrupt
	bit[10]	bit[10]	command queue completed interrupt
	bit[8]	bit[8]	pre_raw_fe frame error interrupt
	bit[3]	bit[3]	pre_raw_fe0 channel 3 frame start interrupt
	bit[2]	bit[2]	pre_raw_fe0 channel 2 frame start interrupt
	bit[1]	bit[1]	pre_raw_fe0 channel 1 frame start interrupt
	bit[0]	bit[0]	pre_raw_fe0 channel 0 frame start interrupt

10.3.2 Interrupt Timing

Interruption is mainly divided into several parts.

a. pre_raw_fe0/pre_raw_be/post_raw frame completion interrupt: indicates that the last pixel of a frame has been transmitted.
b. pre_raw_fe0/pre_raw_be/post_raw register completion interrupt: indicates that the register has updated the working register from the shadow register, and the user can continue to write the register settings of the next frame into the shadow register.
c. p re_raw_fe0 frame error interrupt: when an error condition (ex. drop frame or csi bridge fifo overflow) occurs in the transmission process, ISP will send this interrupt to inform the user that the error detection is in pre_raw_fe0, so post_ raw will not have this interrupt.
d. Instruction queue interrupt: in the instruction queue mode, when the last instruction is completed, this interrupt will be sent to inform the user.
e. p re_raw_fe0 frame start interrupt: this interrupt will be sent at the beginning of each frame, which makes it convenient for users to calculate the number of frames currently transmitted. Only pre_raw_fe0 detects the start of the frame, so post_ raw will not have this interrupt.
f. pre_raw_fe0/post_raw completes the specified line interrupt: in some special applications, the user can specify that the interrupt will be sent when the transmission reach a certain line, rather than when the last frame is completed.

Figure 10-6 is the timing diagram of ISP interrupt under normal transmission condition. Frame start -- > shadow update done -- > frame done will occur in sequence

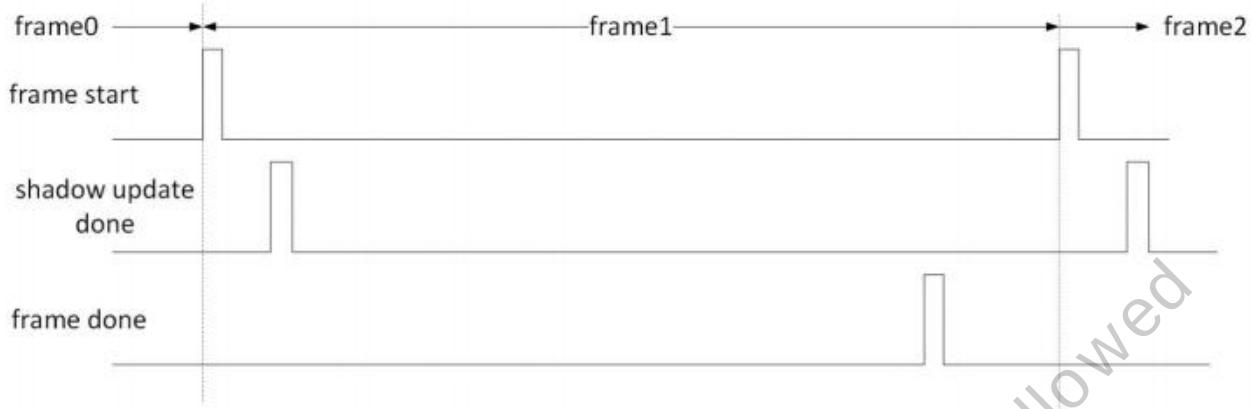


Figure 10- 6 ISP Timing Diagram when Interrupt occurs

10.4 Module Function

10.4.1 Color_bar (patgen)

The graphics generator provides three modes of images.

a. Solid color map
b. Horizontal color block
c. Vertical color block

All modes support gradient color, and the gradient degree automatically matches the image size.

10.4.2 Crop

The module can cut the input image, as shown in Figure 10-7.

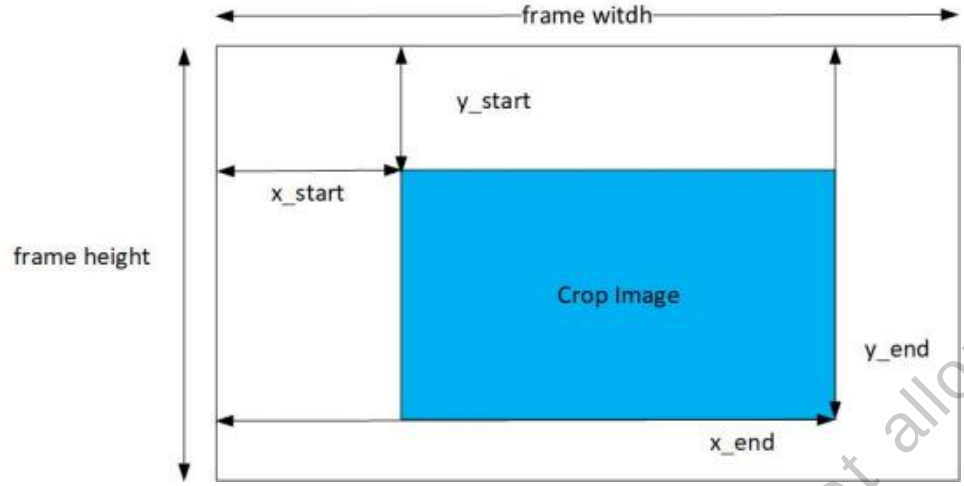


Figure 10- 7 Image Cutting Diagram

10.4.3 AE (Auto Exposure)

Auto exposure collects image data in Bayer domain, and then uses software algorithm to realize auto focus (AF).

Automatic exposure (AE) statistical information cuts the image into 32x30 blocks, accumulates the values of the pixels in each block according to the (R, G, b) three fields, counts the number of points in the R/G/B three fields at the same time, and finally outputs them to the memory (DRAM), and then uses the software algorithm in order to make further AE decision.

The AE module also includes AWB statistics. The AWB statistics information uses a 34x30 block to accumulate the values of pixels in the R/G/B three domains that fall within the specified upper and lower threshold values, and counts the number of pixels that meet the threshold values. Finally, the information is output to DRAM and further AWB decisions are made using algorithms.

10.4.4 AF (Auto Focus)

AF (auto focus) collects image data in Bayer domain, and then uses software algorithm to realize the function of AF.

AF statistics engine cuts the image into 17x15 blocks, makes a series of vertical and horizontal high-pass filters and low-pass filters for the pixels in each block with a 17x5 moving window, counts the number of highlight points in each block, and finally outputs them to DRAM, and then uses software algorithm to make further AF decision.

10.4.5 DIS (Digital image stabilization)

DIS (Digital image stabilization) realizes the function of anti-shaking for images in Bayer domain.

DIS will collect the histogram of cumulative number of green pixels in X and Y directions in 3x3 blocks of the image, and then output it to DRAM for further DIS decision.

10.4.6 BLC (Black level correction)

BLC (black level correction) provides the function of adding and subtracting color difference or multiplying digital gain in Bayer domain. R / Gr / Gb / B has corresponding registers.

10.4.7 DG (Digital Gain)

DG (Digital gain) provides the function of multiplying the image in Bayer domain by digital gain. R / Gr / Gb / B has corresponding registers.

10.4.8 DPC

DPC aims to detect and compensate the bad points, which is divided into two parts: static bad points and dynamic bad points. Static bad points can be filled in the internal SRAM by software in advance, while dynamic bad points are detected dynamically in the process of image moving to compensate for single bad point and bad point aggregation.

10.4.9 GE

GE aims to correct the phenomenon that the pixel values of Gr and Gb are not equal when the sensor leaves the factory, the phenomenon that will lead to irregular noise in the image. After GE compensates the pixels of Gr and Gb, the gap between them is narrowed and the crosstalk phenomenon is removed.

10.4.10 LSC (Lens shading correction)

LSC (lens shading correction) is used to correct the dark area of lens. Due to the optical properties of the lens, the image in the corner area may be darker than that in the central area, so we need to use gain compensation. LSC provides 37x37 gain matrix of four components (R, Gr, Gb, B) for correction. The 37x37 gain matrix is evenly distributed to the input image and uniformly compensated on each phase element by interpolation.

10.4.11 DRC (LTM)

DRC (Dynamic Range Control) adjusts the color intensity by dynamically converting the values with a larger value range to a smaller value range. It selectively enhances the image contrast in different regions while preserving the details of the edges, making the image more suitable for display and human observation.

10.4.12 WBG (White Balance Gain)

WBG (white balance gain) provides the function of multiplying the image in Bayer domain by digital gain. R / Gr / Gb / B has corresponding registers.

10.4.13 BNR (Bayer Noise Reduction)

BNR module in Bayer domain pixel data achieves image denoising. The purpose is to remove the noise, while retaining the details. The module can eliminate the sensor noise according to the noise model provided by users.

10.4.14 DEMOSAIC (CFA)

The purpose of demosaicing is to reconstruct the original pixels into complete R, G, B three-color pixels. Through the image orientation characteristics and edge detection, interpolation is performed on the surrounding pixels to produce accurate and high-resolution pixels. The RGBCAC module is introduced to eliminate the chromatic aberration (purple fringing) distortion commonly caused by lenses in chroma.

10.4.15 CCM

It is to convert the R, G, B obtained by the sensor into the standard R, G, B format through linear transformation. Using 3x3 array, the parameter is s3.10. Using the array CCM parameters corrected in advance under different color temperatures, Firmware dynamically calculates the CCM parameters according to the current image color temperature or image brightness.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} C00 & C01 & C02 \\ C10 & C11 & C12 \\ C20 & C21 & C22 \end{bmatrix} \begin{bmatrix} Rin \\ Gin \\ Bin \end{bmatrix}$$

10.4.16 Gamma

Gamma is an exponential function to adjust the pixel value according to the brightness of the scene. Here we use a 257-point table, where each element is 12bit and is used for adjustment and interpolation. R, G and B can support the dynamic adjustment of different curves.

10.4.17 Dehaze

Dehaze aims at the function of demisting. By analyzing the image scene and calculating the ambient light source, the contrast difference between the fog area and the surrounding area can be obtained. Based on the contrast difference,

the pixels in the fog area can be enhanced to achieve the effect of fog area removal.

10.4.18 CSC

CSC transforms R, G, B to Y, U, V through 3x3 matrix.

10.4.19 3DNR (3-Dimensional Noise Reduction)

Combining the noise suppression function of time domain (adjacent frame) and space domain (surrounding pixel) calculus, the picture is smoother.

10.4.20 YNR

In the luminance Y domain, the noise suppression in the spatial domain is implemented with reference to the information of the surrounding pixels and the motion of the object.

10.4.21 LDCI (DCI)

LDCI is based on the equalization of the region histogram, using enhanced region saturation to improve the details of the dark region and the performance of high-frequency parts.

10.4.22 Sharpen

Sharpen the middle frequency and high frequency edges of the image to highlight the details of the image. The image details can be divided into material area, texture and directional edge. The texture area or edge area can be adjusted separately according to the needs to enhance part of the image details, avoiding noise enhancement caused by single sharpening method and achieving better visual effect.

10.4.23 CNR

CNR implements noise suppression in spatial domain in terms of chroma information, peripheral pixel information and object motion in chroma UV domain.

10.4.24 CAC (PFC inside CNR)

CNR acts on YUV domain and can be used to reduce the noise of chroma. PFC acts at the end of CNR, and its function is to further eliminate the phenomenon of purple fringes, which is common in chroma.

10.4.25 CLUT (HSV_3D_LUT)

CLUT (Color Look-Up Table) uses a 3D LUT (Lookup Table) with a size of 17x17x17 in the RGB color space to perform three-dimensional color adjustment operations, and the adjustment direction of each color in each region can be independently controlled.

10.4.26 RGBCAC

RGBCAC corrects purple fringing in the RGB data field after CFA processing.

10.4.27 PREYEE

The module functions the same as sharpen, but is located in front of NR.

10.4.28 Hist_V

Hist_V is used to calculate the histogram of brightness as the basis for adjusting brightness weights.

10.4.29 CACP

The CA mode provides the ability to multiply the chroma by different gain values determined by the luminance (Luma) and sensitivity luminance (ISO), while the CP mode provides different color results directly corresponding to different brightness.

10.4.30 CA2

CA_LITE provides the ability to multiply the chroma by gain values determined by different saturation levels.

10.4.31 LCAC

Also known as Local Chromatic Aberration Cancellation, its function is to eliminate regional purple fringing. Purple fringing is easy to occur at the junction of high and low brightness, and this function can be turned on for purple fringing removal.

10.4.32 User Gamma

Brightness value gamma correction performed before gamma correction in the RGB domain.

11 Audio interface

11.1 AIAO

11.1.1 Overview

The Audio Input/Audio Output interface is used for connecting with the built-in Audio Codec or external Audio Codec and the digital microphone to complete the transmitting and receiving of audio data and realize the functions of recording, playing and intercom. The AIAO related modules are integrated into a subsystem. The built-in Audio Codec ADC/DAC can support stereo input and output. The AIAO integrates four sets of I2S TX/RX modules and supports two sets of I2S IO interfaces for connecting with external device. The AIAO can transmit and receive the audio data simultaneously and support multi-channel data. The AIAO block diagram is shown as below figure:

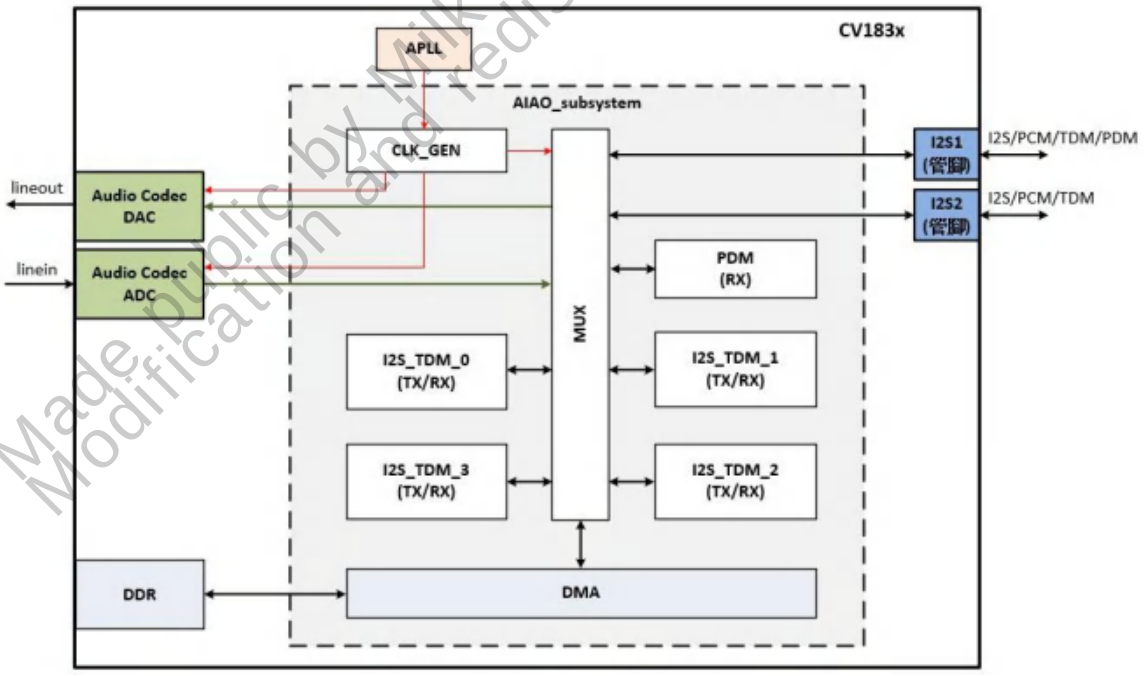


Figure 11-1 AIAO block diagram

11.1.2 Features

AIAO interface supports both Master-mode and Slave-mode of I2S, PCM, multi-channel TDM modes. The received audio data or audio data to be transmitted are transferred using System DMA. The specific features are as follows:

High flexible and configurable timing parameters, including frame period, channel period, frame sync signal active period and polarity
Configurable clock sampling edge for input and output signals
Support transmitting and receiving stereo audio data in I2S master and slave modes
Support transmitting and receiving stereo and mono audio data in PCM master and slave modes
Support transmitting and receiving multi-channel audio data in TDM master and slave modes
Transmitting and receiving operations can be enabled individually or simultaneously

11.1.2.1 PCM interface

Support 16-bit data
Frame sync signal supports short pulse (one bit clock cycle) and long pulse (configurable number of bit clock cycles)
Supports standard mode and left-justified mode

11.1.2.2 I2S interface

Support 16-bit/24-bit data
Support 8kHz ~ 192kHz sampling rate
Configurable LRCK polarity
Supports standard mode and left-justified mode

11.1.3 Function description

AIAO subsystem connects with chip built-in audio codec, I2S IO pins and transmitter/receiver modules through internal pinmux. It needs to be properly configured to achieve different connection modes.

11.1.3.1 Typical applications

The typical application and connection are described as follows:

Support I2S slave mode to connect with internal Audio Codec ADC, or I2S/PCM/TDM master or slave mode to connect with external ADC for audio recording
Support I2S master mode to connect with internal Audio Codec DAC, or I2S/PCM/TDM master or slave mode to connect with external DAC for audio playback

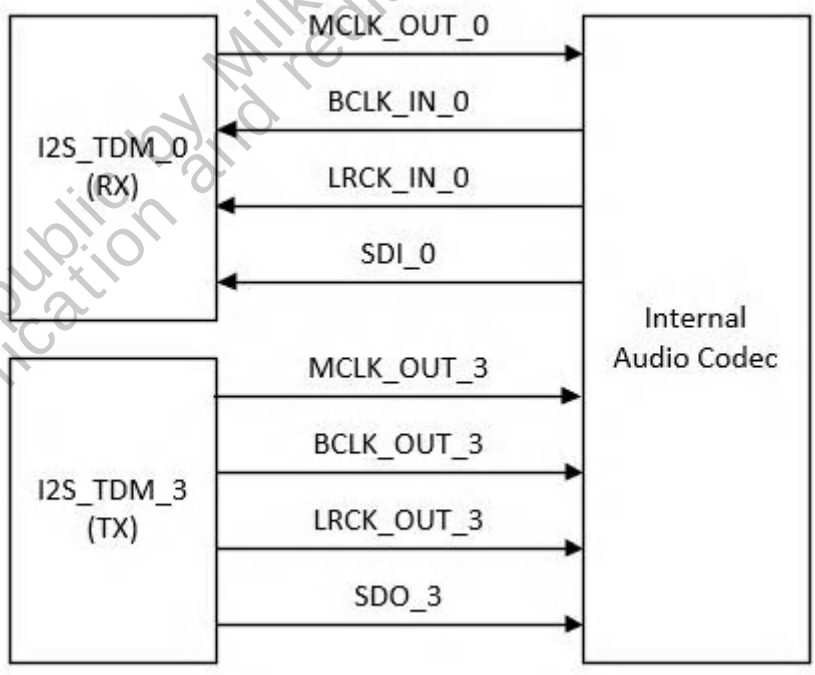


Figure 11-2 Example connection of I2S TX/RX module with internal Audio Codec

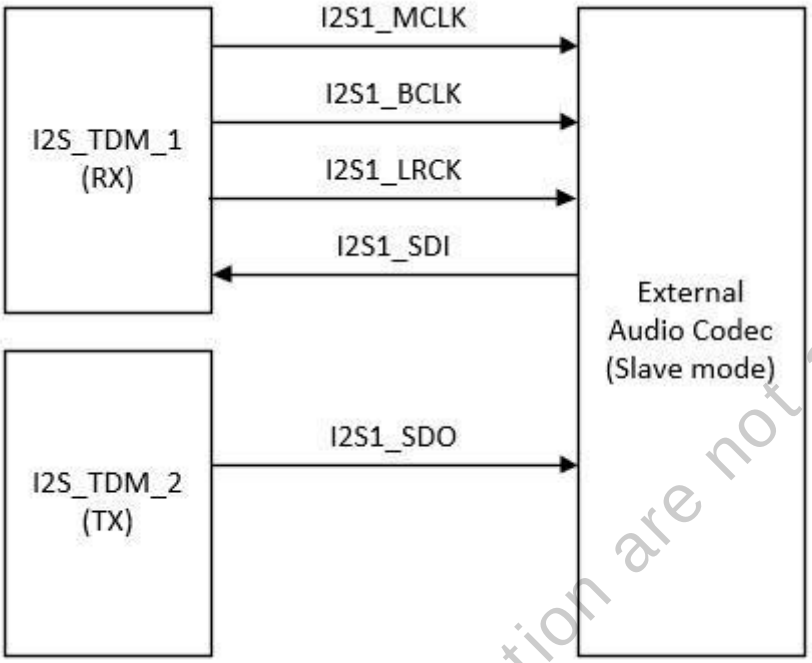


Figure 11-3 Example connection of I2S TX/RX module with external Audio Codec using slave mode

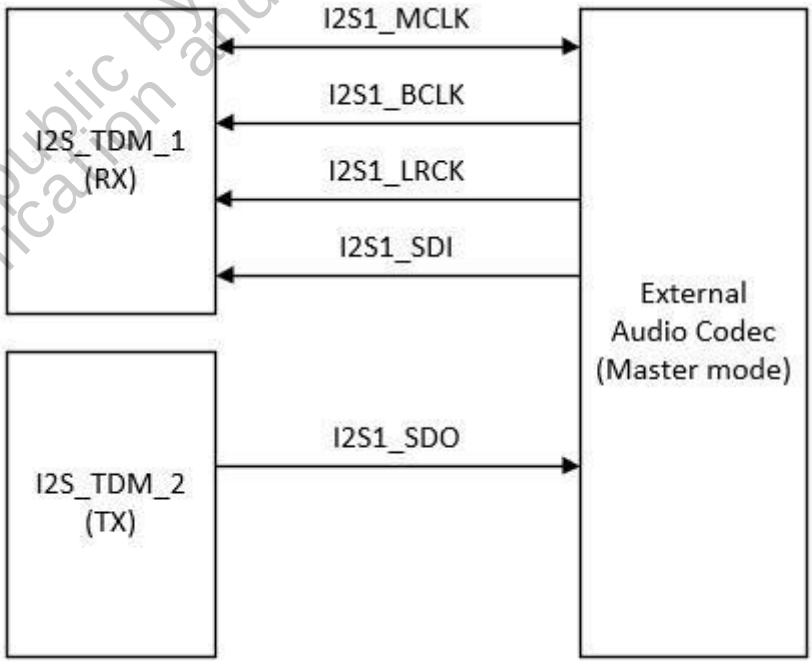


Figure 11-4 Example connection of I2S TX/RX module with external Audio Codec using master mode

11.1.3.2 Interface timing

The audio source is converted into digital data sample by internal or external Audio Codec ADC. The data sample is received by RX module through I2S or PCM interface, and stored into the circular buffer within DRAM through DMA. The data sample can be further processed and transfered to storage device to complete the recording function. The TX module reads the digital data sample from the circular buffer through DMA and transmits to internal or external Audio Codec DAC through I2S or PCM interface to complete the audio playback function.

The typical I2S interface timing is shown in Figure 11-5.

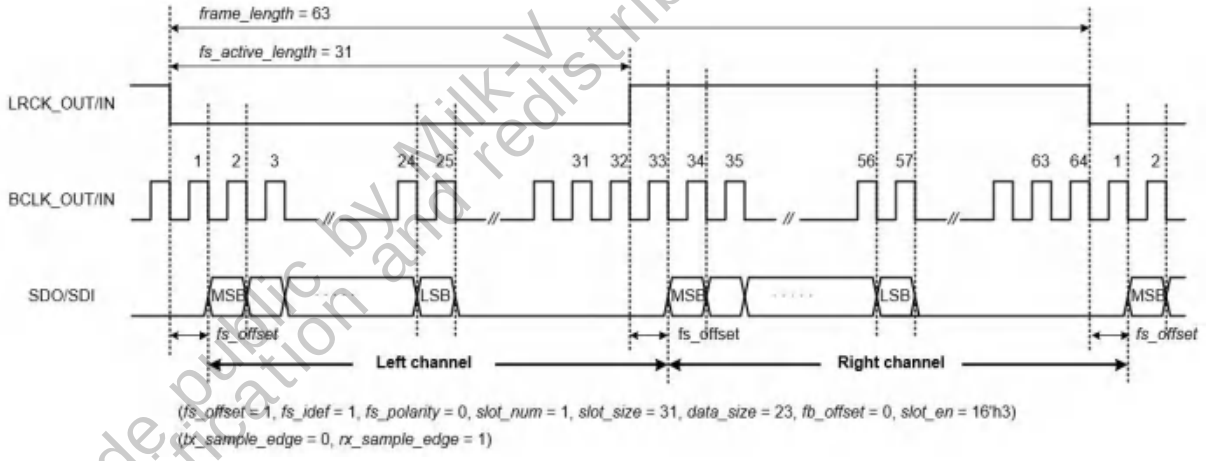


Figure 11-5 I2S interface timing

Figure 11-5 takes 24-bit data sample as an example. The data sample is transmitted in MSB first mode. The MSB delays one BCLK cycle relative to LRCK. The signals are issued at the falling edge of BCLK and latched at the rising edge of BCLK (tx_sample_edge = 0, rx_sample_edge = 1). However, it can be configured to use rising edge of BCLK issue signals and falling edge of BCLK latch signals (tx_sample_edge = 1, rx_sample_edge =0).

The typical PCM interface standard mode timing is shown in Figure 11-6, and the left-justified mode timing is shown in Figure 11-7.

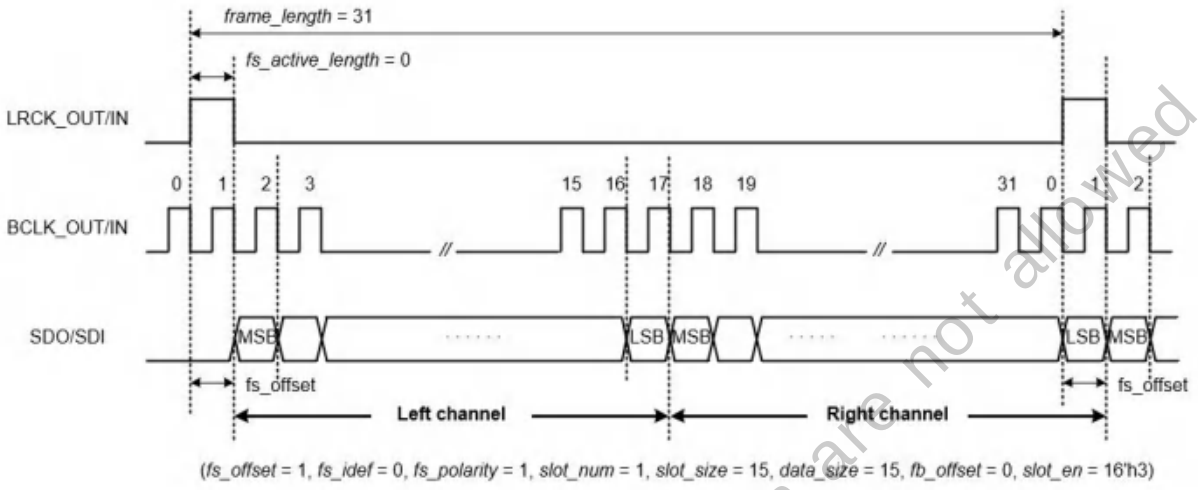


Figure 11-6 PCM interface standard mode timing

Figure 11-6 takes 16-bit data sample as an example. The data sample is transmitted in MSB first mode. The MSB delays one BCLK cycle relative to LRCK. The signals are issued at the falling edge of BCLK and latched at the rising edge of BCLK (tx_sample_edge = 0, rx_sample_edge = 1).

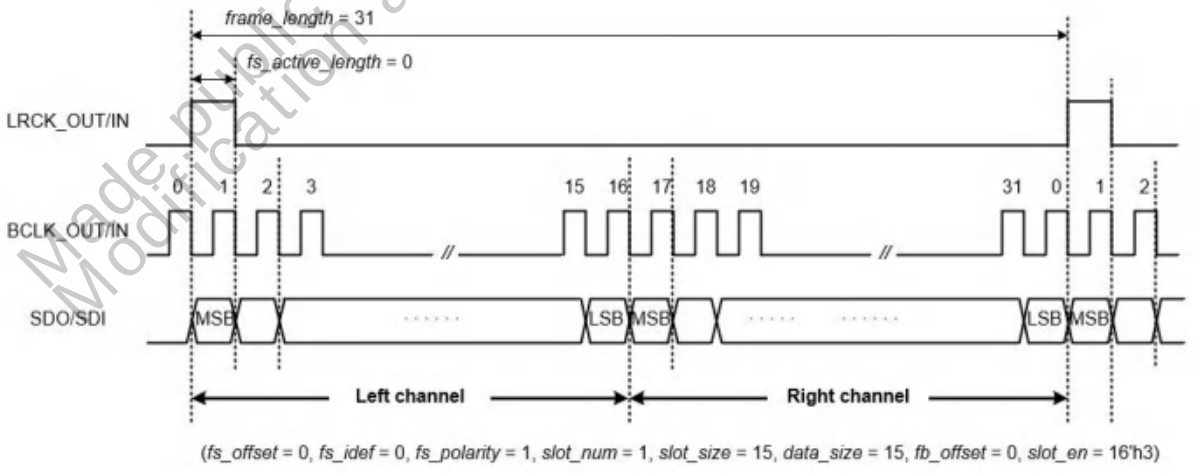


Figure 11-7 PCM interface left-justified mode timing

In left-justified mode, the MSB of data sample and LRCK signals are issued at the same BCLK clock cycle.

11.1.4 Operation control

The AIAO subsystem control registers including i2s_tdm_sclk_in_sel, i2s_tdm_fs_in_sel, i2s_tdm_sdi_in_sel, and i2s_tdm_sdo_out_sel, should be properly configured depending on the audio interface connection prior to enable data transmission.

11.1.4.1 Clock control

If AIAO operates at master mode, the register bit master_mode should be set to 1. The clock division registers I2S_CLK_CTRL1 (mclk_div, bclk_div) should be properly configured depending on data sampling rate, and then set the register aud_en to 1 to turn-on audio clock source.

11.1.4.2 Soft reset

The four TX/RX modules integrated in AIAO all have individual software reset. Each module must be applied FIFO_RESET and I2S_RESET before triggering operation.

11.1.5 AIAO register overview

An overview fo the AIAO subsystem registers is shown in Table 11-1.

Table 11- 1 AIAO subsystem register overview (base address: 0x0410_8000)

Name	Address Offset	Description
i2s_tdm_sclk_in_sel	0x000	Select the source of TX/RX module' SCLK in slave mode.
i2s_tdm_fs_in_sel	0x004	Select the source of TX/RX module's frame synchronizing signal in slave mode.
i2s_tdm_sdi_in_sel	0x008	Select the source of RX module's SDI signal.
i2s_tdm_sdo_out_sel	0x00c	Select the source of subsystem's SDO output.
i2s_bclk_oen_sel	0x030	The output control of BCLK IO.
audio_pdm_ctrl	0x040	Enable PDM mode.

Name	Address Offset	Description
i2s_sys_int_en	0x060	Enable the interrupt signal of I2S subsystem.
i2s_sys_ints	0x064	The status of I2S subsystem's interrupt signal.

I2S_ TDM_ 0~I2S_ TDM_ 3 module registers overview is shown in Table 11-2.

*Table 11- 2 I2S_TDM_0/1/2/3 register overview (address 0x0410_0000 + n*0x10000)*

Name	Address Offset	Description
BLK_MODE_SETTING	0x000	TX/RX module operation control
FRAME_SETTING	0x004	Audio frame timing control
SLOT_SETTING1	0x008	Channel and data control
SLOT_SETTING2	0x00c	Channel enable
DATA_FORMAT	0x010	Specify storage data format
BLK_CFG	0x014	TX/RX block config
I2S_ENABLE	0x018	TX/RX block enable
I2S_RESET	0x01c	TX/RX block reset
I2S_INT_EN	0x020	Interrupt enable
I2S_INT	0x024	Interrupt status
FIFO_THRESHOLD	0x028	FIFO threshold
I2S_LRCK_MASTER	0x02c	BCLK/LRCK master generator mode
FIFO_RESET	0x030	FIFO reset
RX_STATUS	0x040	RX block internal status
TX_STATUS	0x048	TX block internal status
I2S_CLK_CTRL0	0x060	Clock control
I2S_CLK_CTRL1	0x064	Clock divider
I2S_PCM_SYNTH	0x068	PCM FS synthesis mode
RX_RD_PORT	0x080	RX FIFO read port
TX_WR_PORT	0x0c0	TX FIFO write port

11.1.6 AIAO register description

11.1.6.1 AIAO subsystem register description

i2s_tdm_sclk_in_sel

Select sclk source

Offset Address: 0x000

Bits	Name	Access	Description	Reset
2:0	i2s_tdm_0_sclk_in_sel	R/W	Select SCLK input source for i2s_tdm_0 when operates at slave mode 000 = reserved 001 = From i2s_tdm_1 bclk_out 010 = From i2s_tdm_2 bclk_out 011 = From i2s_tdm_3 bclk_out 100 = From IO bclk_in_i2s0 (internal AADC) 101 = From IO bclk_in_i2s1 110 = From IO bclk_in_i2s2 111 = From IO bclk_in_i2s3	0x4
3	Reserved			
6:4	i2s_tdm_1_sclk_in_sel	R/W	Select SCLK input source for i2s_tdm_1 when operates at slave mode 000 = From i2s_tdm_0 bclk_out 001 = From internal audio_pdm i2s_sck 010 = From i2s_tdm_2 bclk_out 011 = From i2s_tdm_3 bclk_out 100 = From IO bclk_in_i2s0 (internal AADC) 101 = From IO bclk_in_i2s1 110 = From IO bclk_in_i2s2 111 = From IO bclk_in_i2s3	0x5
7	Reserved			
10:8	i2s_tdm_2_sclk_in_sel	R/W	Select SCLK input source for i2s_tdm_2 when operates at slave mode 000 = From i2s_tdm_0 bclk_out 001 = From i2s_tdm_1 bclk_out 010 = reserved 011 = From i2s_tdm_3 bclk_out 100 = From IO bclk_in_i2s0 (internal AADC) 101 = From IO bclk_in_i2s1 110 = From IO bclk_in_i2s2 111 = From IO bclk_in_i2s3	0x6
11	Reserved			
14:12	i2s_tdm_3_sclk_in_sel	R/W	Select SCLK input source for i2s_tdm_3 when operates at slave mode 000 = From i2s_tdm_0 bclk_out 001 = From i2s_tdm_1 bclk_out 010 = From i2s_tdm_2 bclk_out 011 = reserved 100 = From IO bclk_in_i2s0 (internal AADC) 101 = From IO bclk_in_i2s1 110 = From IO bclk_in_i2s2 111 = From IO bclk_in_i2s3	0x7
31:15	Reserved			

i2s_tdm_fs_in_sel

Offset Address: 0x004

Bits	Name	Access	Description	Reset
2:0	i2s_tdm_0_fs_in_sel	R/W	Select FS input source for i2s_tdm_0 when operates at slave mode 000 = reserved 001 = From i2s_tdm_1 ws_lrck_fs_out 010 = From i2s_tdm_2 ws_lrck_fs_out 011 = From i2s_tdm_3 ws_lrck_fs_out 100 = From IO ws_lrck_fs_in_i2s0 (internal AADC) 101 = From IO ws_lrck_fs_in_i2s1 110 = From IO ws_lrck_fs_in_i2s2 111 = From IO ws_lrck_fs_in_i2s3	0x4
3	Reserved			
6:4	i2s_tdm_1_fs_in_sel	R/W	Select FS input source for i2s_tdm_1 when operates at slave mode 000 = From i2s_tdm_0 ws_lrck_fs_out 001 = From internal audio_pdm i2s_lrck 010 = From i2s_tdm_2 ws_lrck_fs_out 011 = From i2s_tdm_3 ws_lrck_fs_out 100 = From IO ws_lrck_fs_in_i2s0 (internal AADC) 101 = From IO ws_lrck_fs_in_i2s1 110 = From IO ws_lrck_fs_in_i2s2 111 = From IO ws_lrck_fs_in_i2s3	0x5
7	Reserved			
10:8	i2s_tdm_2_fs_in_sel	R/W	Select FS input source for i2s_tdm_2 when operates at slave mode 000 = From i2s_tdm_0 ws_lrck_fs_out 001 = From i2s_tdm_1 ws_lrck_fs_out 010 = reserved 011 = From i2s_tdm_3 ws_lrck_fs_out 100 = From IO ws_lrck_fs_in_i2s0 (internal AADC) 101 = From IO ws_lrck_fs_in_i2s1 110 = From IO ws_lrck_fs_in_i2s2 111 = From IO ws_lrck_fs_in_i2s3	0x6
11	Reserved			
14:12	i2s_tdm_3_fs_in_sel	R/W	Select FS input source for i2s_tdm_3 when operates at slave mode 000 = From i2s_tdm_0 ws_lrck_fs_out 001 = From i2s_tdm_1 ws_lrck_fs_out 010 = From i2s_tdm_2 ws_lrck_fs_out 011 = reserved 100 = From IO ws_lrck_fs_in_i2s0 (internal AADC) 101 = From IO ws_lrck_fs_in_i2s1 110 = From IO ws_lrck_fs_in_i2s2 111 = From IO ws_lrck_fs_in_i2s3	0x7
31:15	Reserved			

i2s_tdm_sdi_in_sel

Offset Address: 0x008

Bits	Name	Access	Description	Reset
2:0	i2s_tdm_0_sdi_in_sel	R/W	Select SDI input source for i2s_tdm_0 000 = reserved 001 = From i2s_tdm_1 sdo	0x4

Bits	Name	Access	Description	Reset
			010 = From i2s_tdm_2 sdo 011 = From i2s_tdm_3 sdo 100 = From IO sdi_i2s0 (internal AADC) 101 = From IO sdi_i2s1 110 = From IO sdi_i2s2 111 = From IO sdi_i2s3	
3	Reserved			
6:4	i2s_tdm_1_sdi_in_sel	R/W	Select SDI input source for i2s_tdm_1 000 = From i2s_tdm_0 sdo 001 = From internal audio_pdm i2s_sdata 010 = From i2s_tdm_2 sdo 011 = From i2s_tdm_3 sdo 100 = From IO sdi_i2s0 (internal AADC) 101 = From IO sdi_i2s1 110 = From IO sdi_i2s2 111 = From IO sdi_i2s3	0x5
7	Reserved			
10:8	i2s_tdm_2_sdi_in_sel	R/W	Select SDI input source for i2s_tdm_2 000 = From i2s_tdm_0 sdo 001 = From i2s_tdm_1 sdo 010 = reserved 011 = From i2s_tdm_3 sdo 100 = From IO sdi_i2s0 (internal AADC) 101 = From IO sdi_i2s1 110 = From IO sdi_i2s2 111 = From IO sdi_i2s3	0x6
11	Reserved			
14:12	i2s_tdm_3_sdi_in_sel	R/W	Select SDI input source for i2s_tdm_3 000 = From i2s_tdm_0 sdo 001 = From i2s_tdm_1 sdo 010 = From i2s_tdm_2 sdo 011 = reserved 100 = From IO sdi_i2s0 (internal AADC) 101 = From IO sdi_i2s1 110 = From IO sdi_i2s2 111 = From IO sdi_i2s3	0x7
31:15	Reserved			

i2s_tdm_sdo_out_sel

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
2:0	i2s_tdm_0_sdo_out_sel	R/W	Only 0x4 is allowed.	0x4
3	Reserved			
6:4	i2s_tdm_1_sdo_out_sel	R/W	Select sdo_i2s1 output from which i2s_tdm 100 = select i2s_tdm_0 sdo to IO sdo_i2s1 101 = select i2s_tdm_1 sdo to IO sdo_i2s1 110 = select i2s_tdm_2 sdo to IO sdo_i2s1 111 = select i2s_tdm_3 sdo to IO sdo_i2s1 others = reserved	0x5

Bits	Name	Access	Description	Reset
7	Reserved			
10:8	i2s_tdm_2_sdo_out_sel	R/W	Select sdo_i2s2 output from which i2s_tdm 100 = select i2s_tdm_0 sdo to IO sdo_i2s2 101 = select i2s_tdm_1 sdo to IO sdo_i2s2 110 = select i2s_tdm_2 sdo to IO sdo_i2s2 111 = select i2s_tdm_3 sdo to IO sdo_i2s2 others = reserved	0x6
11	Reserved			
14:12	i2s_tdm_3_sdo_out_sel	R/W	Only 0x7 is allowed.	0x7
31:15	Reserved			

i2s_bclk_oen_sel

Offset Address: 0x030

Bits	Name	Access	Description	Reset
0	i2s0_bclk_oen_sel	R/W	Reserved	0x0
1	i2s1_bclk_oen_sel	R/W	Select bclk_out_i2s1 IO oen control 0 = bclk_oen from I2S1 internal 1 = bclk_oen controlled by i2s1_bclk_oen_ext	0x0
2	i2s2_bclk_oen_sel	R/W	Select bclk_out_i2s2 IO oen control 0 = bclk_oen from I2S2 internal 1 = bclk_oen controlled by i2s2_bclk_oen_ext	0x0
3	i2s3_bclk_oen_sel	R/W	Only 0x0 is allowed.	0x0
7:4	Reserved			
8	i2s0_bclk_oen_ext	R/W	External control bclk_out_i2s0 IO oen 0 = output disable 1 = output enable	0x0
9	i2s1_bclk_oen_ext	R/W	External control bclk_out_i2s1 IO oen 0 = output disable 1 = output enable	0x0
10	i2s2_bclk_oen_ext	R/W	External control bclk_out_i2s2 IO oen 0 = output disable 1 = output enable	0x0
11	i2s3_bclk_oen_ext	R/W	Only 0x0 is allowed.	0x0
15:12	Reserved			
16	i2s_bclk_oen_no_delay	R/W	Only 0x0 is allowed.	0x0
31:17	Reserved			

audio_pdm_ctrl

Offset Address: 0x040

Bits	Name	Access	Description	Reset
1:0	audio_pdm_sel_i2s_io	R/W	Enable PDM mode: 00 = General operating mode 01 = I2S1 IO operates in PDM mode 10 = I2S2 IO operates in PDM mode The i2s_tdm_1 RX module is used to receive data. When this value	0x0

Bits	Name	Access	Description	Reset
			is set to 01, the oen of I2S1_BCLK IO is fixedly controlled by the i2s1_bclk_oen_ext register. When this value is set to 10, the oen of I2S2_BCLK IO is fixedly controlled by the i2s2_bclk_oen_ext register.	
31:2	Reserved			

i2s_sys_int_en

Offset Address: 0x060

Bits	Name	Access	Description	Reset
0	i2s0_int_en	R/W	Enable I2S0 interrupt	0x1
1	i2s1_int_en	R/W	Enable I2S1 interrupt	0x1
2	i2s2_int_en	R/W	Enable I2S2 interrupt	0x1
3	i2s3_int_en	R/W	Enable I2S3 interrupt	0x1
7:4	Reserved			
8	i2s_subsys_int_en	R/W	Enable I2S_SUBSYS interrupt	0x1
31:9	Reserved			

i2s_sys_ints

Offset Address: 0x064

Bits	Name	Access	Description	Reset
0	i2s0_int	RO	I2S0 interrupt status When an I2S0 interrupt occurs, the I2S_INT value in the I2S0 register can be further read to determine the interrupt status.	
1	i2s1_int	RO	I2S1 interrupt status When an I2S1 interrupt occurs, the I2S_INT value in the I2S1 register can be further read to determine the interrupt status.	
2	i2s2_int	RO	I2S2 interrupt status When an I2S2 interrupt occurs, the I2S_INT value in the I2S2 register can be further read to determine the interrupt status.	
3	i2s3_int	RO	I2S3 interrupt status When an I2S3 interrupt occurs, the I2S_INT value in the I2S3 register can be further read to determine the interrupt status.	
31:4	Reserved			

11.1.6.2 I2S_TDM module register description

BLK_MODE_SETTING

Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	tx_mode	R/W	Transmission mode 0 = RX mode, 1 = TX mode	0x0
1	master_mode	R/W	I2S master/slave operation mode 0 = slave mode, 1 = master mode	0x0
2	rx_sample_edge	R/W	Select sampling clock edge for SDI & LRCK 0 = negative edge, 1 = positive edge	0x0
3	tx_sample_edge	R/W	Select sampling clock edge for SDO in TX mode 0 = negative edge, 1 = positive edge	0x0
6:4	Reserved			
7	dma_mode	R/W	DMA transfer mode 0 = SW mode, 1 = HW DMA mode	0x1
8	Reserved	R/W	Multiple I2S synchronous operation mode 0 = standalone operation 1 = support multi-bit I2S	0x0
9	Reserved			
10	slave_tx_fs_direct_in	R/W	WS(FS/LRCK) directly input for slave mode TX 0 = internally generate WS 1 = select external WS from master	0x0
11	Reserved			
12	pcm_synth_mode	R/W	PCM FS use synthesis mode 0 = PCM FS period controlled by frame_length 1 = PCM FS period controlled by ck_coef_n & ck_coef_m Only valid for Master mode.	0x0
15:13	Reserved			
31:16	reg_dummy	R/W	Reserved	0xff00

FRAME_SETTING

Offset Address: 0x004

Bits	Name	Access	Description	Reset
8:0	frame_length	R/W	Audio frame length bits 0~511 = 1~512 bits	0x1F
11:9	Reserved			
12	fs_polarity	R/W	Frame sync polarity 0 = active low, 1 = active high For I2S mode, set this bit to 0 means left channel first (LRCK low). For PCM mode, set this bit to 1 means active high FS pulse.	0x0
13	fs_offset	R/W	Frame offset, first bit starts: 0 = the same edge with FS transition	0x1

Bits	Name	Access	Description	Reset
			1 = 1 bit after FS transition Set this bit to 0 for Left-justified mode.	
14	fs_idef	R/W	Frame sync signal role 0 = as start of frame (FS, usually 1 bit clk) 1 = as channel side identification (WS/LRCK) Notes: If fs_offset = 1 & fs_idef = 0, one bit offset will be inserted before MSB of first data sample for each frame. If fs_offset = 1 & fs_idef = 1, one bit offset will be inserted before MSB of every sample whenever WS/LRCK toggle.	0x1
15	Reserved			
23:16	fs_active_length	R/W	Frame sync (FS/WS/LRCK) active length 0~255 = 1~256 bits. The value of fs_active_length must less than frame_length. It is usually half of frame length in I2S mode, and is 1-bit for PCM/TDM mode.	0x0F
31:24	Reserved			

SLOT_SETTING1

Offset Address: 0x008

Bits	Name	Access	Description	Reset
3:0	slot_num	R/W	Number of slot (channel) per audio frame 0~15 = 1~16 slots Set to 1 for two (left/right) channels I2S/PCM mode.	0x1
7:4	Reserved			
13:8	slot_size	R/W	Slot size 0~63 = 1~64 bits	0x0F
15:14	Reserved			
20:16	data_size	R/W	data size inside the slot (channel) 0~31 = 1~32 bits Set same value for slot_size and data_size in I2S/PCM mode.	0x0F
23:21	Reserved			
28:24	fb_offset	R/W	data offset inside the Slot 0 = no offset, 1~31 = 1~31 bits Set the first valid data bit within each slot.	0x0
31:29	Reserved			

SLOT_SETTING2

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
15:0	slot_en	R/W	Active slot (channel) slot_en[n] = 1 means slot-n is active. Otherwise, slot-n is inactive.	0x0003
31:16	Reserved			

DATA_FORMAT

Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	data_format	R/W	Storage data format 0 = packet mode 1 = reserved	0x0
2:1	word_length	R/W	Word length selection for each data sample storage 00 = 8-bit, 01 = 16-bit, 10 = 32-bit, 11 = reserved Right align each data sample if size smaller than setting.	0x1
3	pad_slot_no	R/W	Pad slot number to data sample (RX only) 0 = no 1 = pad 4-bit slot/channel number to MSB of each data sample. For example, if data sample size is 24-bit and register word_length set to 2'b10, each data will be padded to 32-bit word as {slot[3:0], 4'h0, data[23:0]}. This function is no used when data size is equal to word_length.	0x0
4	skip_rx_inactive_slot	R/W	Skip inactive Slot data for receive mode 0 = store zero instead of received data into FIFO during inactive slot state 1= skip received data during inactive slot state without store into FIFO	0x0
5	skip_tx_inactive_slot	R/W	Skip inactive Slot data for transmit mode 0 = Read data from FIFO but transmit zero to receiver during inactive slot state 1= directly transmit zero to receiver without reading FIFO during inactive slot state	0x0
6	tx_source_left_align	R/W	Select left-aligned word data for transmit 0 = select right-aligned 1 = select left-aligned When word_length is larger than data_size in TX mode, for example source data is 32-bit (word_length = 2'b10) and data sample to be transmitted is 24-bit, bits [23:0] of source data will be selected if set this bit to 0, and bits [31:8] of source data will be selected if set this bit to 1.	0x0
31:7	Reserved			

BLK_CFG

Offset Address: 0x014

Bits	Name	Access	Description	Reset
0	force_complete	R/W	force complete mode 0 = stop operation until an audio frame	0x0

Bits	Name	Access	Description	Reset
			is received or transmitted complete after i2s_enable set from 1 to 0 1 = force stop TX or RX operation immediately after i2s_enable set from 1 to 0	
1	dma_req_force_stop	R/W	dma_req force stop mode 0 = dma_req keep high until receive dma_ack after i2s_enable set from 1 to 0 1 = dma_req forced to low after i2s_enable set from 1 to 0 in dma mode	0x1
3:2	Reserved			
4	auto_disable_with_ch_en	R/W	I2S FIFO control auto disable 0 = normal operation 1 = FIFO control will automatically stop operation after DMA channel is disabled before I2S is disabled (i2s_enable set to 0). No further RX data will be received and TX zero bit will be transmitted.	0x0
5	Reserved			
6	rx_start_wait_dma_en	R/W	I2S receiver block wait DMA enable 0 = I2S receiver block start operation after i2s_enable set from 0 to 1 1 = I2S receiver block will pend operation until DMA is enabled	0x0
7	Reserved			
8	rx_blk_clk_force_en	R/W	I2S receiver block sclk force on 0 = enable clock auto gating 1 = always on	0x0
9	rx_fifo_dma_clk_force_en	R/W	RX FIFO/DMA control block pclk force on 0 = enable clock auto gating 1 = always on	0x1
15:10	Reserved			
16	tx_blk_clk_force_en	R/W	I2S transmitter block sclk force on 0 = enable clock auto gating 1 = always on	0x0
17	tx_fifo_dma_clk_force_en	R/W	TX FIFO/DMA control block pclk force on 0 = enable clock auto gating 1 = always on	0x1
31:18	Reserved			

I2S_ENABLE

block enable

Offset Address: 0x018

Bits	Name	Access	Description	Reset
0	i2s_enable	R/W	i2s_tdm engine enable Must set i2s_reset_tx or i2s_reset_rx to 1 then to 0 firstly before setting i2s_enable	0x0
31:1	Reserved			

I2S_RESET

sw reset

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
0	i2s_reset_rx	R/W	Reset receiver block 0 = nop, 1 = reset Set to 1 then set to 0 to reset receiver block. Must add some delay before set this bit back to 0 due to the signal needs to be synced from pclk domain to sclk domain.	0x0
1	i2s_reset_tx	R/W	Reset transmitter block 0 = nop, 1 = reset Set to 1 then set to 0 to reset transmit block. Must add some delay before set this bit back to 0 due to the signal needs to be synced from pclk domain to sclk domain.	0x0
31:2	Reserved			

I2S_INT_EN

interrupt enable

Offset Address: 0x020

Bits	Name	Access	Description	Reset
0	rx_fifo_avail_int_en	R/W	Enable RX FIFO data available interrupt	0x0
1	rx_fifo_overflow_int_en	R/W	Enable RX FIFO overflow interrupt	0x1
2	rx_fifo_underflow_int_en	R/W	Enable RX FIFO underflow interrupt	0x1
3	Reserved			
4	tx_fifo_avail_int_en	R/W	Enable TX FIFO data available interrupt	0x0
5	tx_fifo_overflow_int_en	R/W	Enable TX FIFO overflow interrupt	0x1
6	tx_fifo_underflow_int_en	R/W	Enable TX FIFO underflow interrupt	0x1
7	Reserved			
8	i2s_int_en	R/W	Enable I2S IP interrupt All I2S interrupt signals are merged into a 1-bit signal and reflected in the I2S subsystem register i2s_sys_ints.	0x1
31:9	Reserved			

I2S_INT

interrupt status

Offset Address: 0x024

Bits	Name	Access	Description	Reset
0	rx_fifo_avail_int	RO	RX FIFO data available interrupt status This bit will be set to 1 whenever RX FIFO fullness is larger than rx_fifo_threshold and rx_fifo_avail_int_en is 1. Write 1 to clear.	
1	rx_fifo_overflow_int	RO	RX FIFO overflow interrupt status Write 1 to clear.	

Bits	Name	Access	Description	Reset
2	rx_fifo_underflow_int	RO	RX FIFO underflow interrupt status Write 1 to clear.	
3	Reserved			
4	tx_fifo_avail_int	RO	TX FIFO data available interrupt status This bit will be set to 1 whenever TX FIFO emptyness is larger than tx_fifo_threshold and tx_fifo_avail_int_en is 1. Write 1 to clear.	
5	tx_fifo_overflow_int	RO	TX FIFO overflow interrupt status Write 1 to clear.	
6	tx_fifo_underflow_int	RO	TX FIFO underflow interrupt status Write 1 to clear.	
7	Reserved			
8	rx_fifo_avail_int_raw	RO	RX FIFO data available interrupt raw status This bit will be set to 1 whenever RX FIFO fullness is larger than rx_fifo_threshold no matter rx_fifo_avail_int_en is 1 or 0. Write 1 to clear.	
9	rx_fifo_overflow_int_raw	RO	RX FIFO overflow interrupt raw status Write 1 to clear.	
10	rx_fifo_underflow_int_raw	RO	RX FIFO underflow interrupt raw status Write 1 to clear.	
11	Reserved			
12	tx_fifo_avail_int_raw	RO	TX FIFO data available interrupt raw status This bit will be set to 1 whenever TX FIFO emptyness is larger than tx_fifo_threshold no matter tx_fifo_avail_int_en is 1 or 0. Write 1 to clear.	
13	tx_fifo_overflow_int_raw	RO	TX FIFO overflow interrupt raw status Write 1 to clear.	
14	tx_fifo_underflow_int_raw	RO	TX FIFO underflow interrupt raw status Write 1 to clear.	
31:15	Reserved			

FIFO_THRESHOLD

Offset Address: 0x028

Bits	Name	Access	Description	Reset
4:0	rx_fifo_threshold	R/W	RX FIFO threshold level Issue rx_fifo_avail_int when FIFO fullness larger than this value	0x7
15:5	Reserved			
20:16	tx_fifo_threshold	R/W	TX FIFO threshold level Issue tx_fifo_avail_int when FIFO emptyness larger than this value	0x7
23:21	Reserved			
28:24	tx_fifo_high_threshold	R/W	TX FIFO high threshold level Transmit block start when FIFO fullness larger than this value	0x1F
31:29	Reserved			

I2S_LRCK_MASTER

block enable

Offset Address: 0x02c

Bits	Name	Access	Description	Reset
0	i2s_lrck_master_enable	R/W	Enable i2s_tdm use as bclk/lrck master generator If the i2s_tdm IP need to operate as a BCLK & LRCK generator master only without TX or RX data transfer, set master_mode to 1 and BCLK will start to output after aud_en set to 1. Then LRCK will start to output after this bit set to 1. No need to set i2s_enable in this condition. In addition, apply i2s_reset_rx or i2s_reset_tx before set this bit to 1.	0x0
31:1	Reserved			

FIFO_RESET

Offset Address: 0x030

Bits	Name	Access	Description	Reset
0	rx_fifo_reset	R/W	Receive Channel FIFO Reset Write 1 then write 0 to reset RX FIFO	0x0
15:1	Reserved			
16	tx_fifo_reset	R/W	Transmit Channel FIFO Reset Write 1 then write 0 to reset TX FIFO	0x0
31:17	Reserved			

RX_STATUS

Offset Address: 0x040

Bits	Name	Access	Description	Reset
8:0	rx_frame_size_cnt	RO	Receive blk internal counter status	
9	rx_i2s_disable_req	RO	Internal signal	
10	i2s_rx_start_wait	RO	Internal signal	
16:11	rx_data_size_cnt	RO	Receive blk internal counter status	
22:17	rx_slot_size_cnt	RO	Receive blk internal counter status	
23	i2s_reset_rx_sclk	RO	i2s_reset_rx in sclk domain	
28:24	rx_slot_num_cnt	RO	Receive blk internal counter status	
29	receive_start_sclk	RO	Receive blk start in sclk domain	
30	rx_blk_active	RO	Receive blk active	
31	rx_dma_req	RO	Receive DMA request	

TX_STATUS

Offset Address: 0x048

Bits	Name	Access	Description	Reset
8:0	tx_frame_size_cnt	RO	Transmit blk internal counter status	
9	tx_i2s_disable_req	RO	Internal signal	
10	i2s_tx_start_wait	RO	Internal signal	

Bits	Name	Access	Description	Reset
16:11	tx_data_size_cnt	RO	Transmit blk internal counter status	
22:17	tx_slot_size_cnt	RO	Transmit blk internal counter status	
23	i2s_reset_tx_sclk	RO	i2s_reset_tx in sclk domain	
28:24	tx_slot_num_cnt	RO	Transmit blk internal counter status	
29	transmit_start_sclk	RO	Transmit blk start in sclk domain	
30	tx_blk_active	RO	Transmit blk active	
31	tx_dma_req	RO	Transmit DMA request	

I2S_CLK_CTRL0

Offset Address: 0x060

Bits	Name	Access	Description	Reset
0	aud_clk_sel	R/W	I2S audio clock (aud_clk) source select 0 = from audio PLL 1 = from external mclk_in	0x0
2:1	Reserved		reserved	
3	bclk_out_inv	R/W	bclk_out clock inverse 0 = bclk_out without inverse 1 = bclk_out inverted	0x0
4	bclk_in_inv	R/W	bclk_in clock inverse 0 = bclk_in without inverse 1 = bclk_in inverted	0x0
5	Reserved		bclk_in clock inverse 0 = bclk_in without inverse 1 = bclk_in inverted	
6	bclk_out_clk_force_en	R/W	bclk_out clock force enable 1 = always output 0 = bclk_out output after transmission start	0x1
7	mclk_out_en	R/W	mclk_out IO output enable 0 = disable 1 = enable	0x0
8	aud_en	R/W	I2S clock gen and master signal out enable 0 = disable clock generator and bclk_out IO 1 = enable clock generator and bclk_out IO Always set this bit to 0 when I2S operate at slave mode. When set this bit to 0, bclk generated from internal clock generator will be gated and clock dividor of bclk and mclk will be reset. Specify mclk_div or bclk_div before set aud_en to 1.	0x0
31:9	Reserved			

I2S_CLK_CTRL1

Offset Address: 0x064

Bits	Name	Access	Description	Reset
15:0	mclk_div	R/W	mclk clock divider from audio clock 1 = div 1, 2 = div 2, …	0x2
31:16	bclk_div	R/W	bclk clock divider from audio clock (for	0x2

Bits	Name	Access	Description	Reset
			master mode only) 1 = div 1, 2 = div 2, ...	

I2S_PCM_SYNTH

Offset Address: 0x068

Bits	Name	Access	Description	Reset
11:0	ck_coef_n	R/W	PCM FS period synthesis timing coefficient N The frequency of FS for PCM mode = BCLK * (N/M) when pcm_synth_mode set to 1. The value of ck_coef_n must smaller than ck_coef_m.	0x1
15:12	Reserved			
31:16	ck_coef_m	R/W	PCM FS period synthesis timing coefficient M The frequency of FS for PCM mode = BCLK * (N/M) when pcm_synth_mode set to 1.	0x40

RX_RD_PORT

Offset Address: 0x080

Bits	Name	Access	Description	Reset
31:0	rx_rd_port	RO	Receive data FIFO Read Port	

TX_WR_PORT

Offset Address: 0x0c0

Bits	Name	Access	Description	Reset
31:0	tx_wr_port	WO	Transmit data FIFO Write Port	0x0

11.2 Audio Codec

11.2.1 Overview

The chip integrates high-performance Audio Codec, including stereo playback DAC (90dB DR A-Weighted), supporting two single-end lineout; stereo recording ADC (90dB DR A-Weighted), supporting stereo single-end input.

11.2.2 Characteristics

90dB DR A-Weighted stereo DAC

Stereo single-end Lineout
Digital volume control range of DAC: - 24dB ~ 0dB;
The DAC supports 8kHz ~ 48kHz sampling rate
90dB DR A-Weighted stereo ADC
The gain range of ADC PGA is 0dB ~ 48dB
The ADC supports 8kHz~48kHz sampling rate
The ADC supports Mic in stereo single-end input or line in stereo single-end input

11.2.3 Audio Codec register overview

11.2.3.1 Audio DAC/ADC register overview

Name	Address Offset	Description
txdac_ctrl0	0x000	
txdac_ctrl1	0x004	
txdac_status	0x008	
txdac_afe0	0x00c	
txdac_afe1	0x010	
txdac_ana0	0x020	
txdac_ana1	0x024	
rxadc_ctrl0	0x100	
rxadcc_ctrl1	0x104	
rxadc_status	0x108	
rxadc_ana0	0x110	
rxadc_ana1	0x114	
rxadc_ana2	0x118	
rxadc_ana3	0x11c	
rxadc_ana4	0x120	

11.2.3.2 Audio ADC Register Description

rxadc_ctrl0

Offset Address: 0x100

Bits	Name	Access	Description	Reset
0	reg_rxadc_en	R/W		0x0
1	reg_i2s_tx_en	R/W		0x0

Bits	Name	Access	Description	Reset
31:2	Reserved			

rxadcc_ctrl1

Offset Address: 0x104

Bits	Name	Access	Description	Reset
1:0	reg_rxadc_cic_opt	R/W	CIC decimation filter option 0: downsample ratio 64 1: downsample ratio 128 2: downsample ratio 256 3: downsample ratio 512	0x0
2	reg_rxadc_chn_swap	R/W	L/R input data channel swap	0x0
3	reg_rxadc_single	R/W	Only single channel supported, used when ANALOG in differential input mode	0x0
6:4	reg_rxadc_dcb_opt	R/W	DC blocking filter option 3'b000: bypass 3'b001: 1-2^(-8) 3'b010: 1-2^(-9) 3'b011: 1-2^(-10) 3'b100: 1-2^(-11) 3'b101: 1-2^(-12) other: bypass	0x5
7	Reserved			
8	reg_rxadc_igr_init	R/W	i2s keep silence when filter with initial value	0x0
9	reg_rxadc_clk_force_en	R/W	force clock enable	0x0
10	reg_rxadc_fsm_src_sel	R/W	FSM trigger source 0: adc0_vld 1: adc1_vld	0x0
31:11	Reserved			

rxadc_status

Offset Address: 0x108

Bits	Name	Access	Description	Reset
0	reg_rxadc_cic0_init_done	RO	cic_d_0 init done	
1	reg_rxadc_fir1_0_init_done	RO	fir1_0 init done	
2	reg_rxadc_fir2_0_init_done	RO	fir2_0 init done	

Bits	Name	Access	Description	Reset
3	Reserved			
4	reg_rxadc_cic1_init_done	RO	cic_d_1 init done	
5	reg_rxadc_fir1_1_init_done	RO	fir1_1 init done	
6	reg_rxadc_fir2_1_init_done	RO	fir2_1 init done	
7	Reserved			
10:8	reg_rxadc_fsm	RO	ADC FSM state	
31:11	Reserved			

rxadc_ana0

Offset Address: 0x110

Bits	Name	Access	Description	Reset
12:0	reg_gstepl_rxpga	R/W	PGA feedback resistance selection 2dB/step default:20Kohm	0x0001
13	reg_g6dbl_rxpga	R/W	PGA input resistance selection 0:20Kohm 1:10Kohm	0x0
15:14	reg_gainl_rxadc	R/W	PGA input resistance selection 0:20Kohm(0dB) 1:10Kohm(6dB) 2: 5Kohm(12dB) 3: 2.5Kohm(18dB)	0x0
28:16	reg_gstepr_rxpga	R/W	PGA feedback resistance selection 2dB/step default:20Kohm	0x0001
29	reg_g6dbr_rxpga	R/W	PGA input resistance selection 0:20Kohm 1:10Kohm	0x0
31:30	reg_gainr_rxadc	R/W	PGA input resistance selection 0:20Kohm(0dB) 1:10Kohm(6dB) 2: 5Kohm(12dB) 3: 2.5Kohm(18dB)	0x0

rxadc_ana1

Offset Address: 0x114

Bits	Name	Access	Description	Reset
15:0	reg_gainl_status	RO	[12:0]: gstep1 [13]: g6dbl [15:14]: gainl	
31:16	reg_gainr_status	RO	[28:16]: gstepr [29]: g6dbr [31:30]: gainr	

rxadc_ana2

Offset Address: 0x118

Bits	Name	Access	Description	Reset
0	reg_mutel_rxpga	R/W	Enable pin of mute left channel, active high	0x0
1	reg_muter_rxpga	R/W	Enable pin of mute right channel, active high	0x0
15:2	Reserved			
16	reg_diff_en_rxpga	R/W	Enable pin of differential mode(Left channel only, VIN/VINB=PAD_VINL/PAD_VINR	0x0

Bits	Name	Access	Description	Reset
17	reg_tristate_rxpga	R/W	?	0x0
31:18	Reserved			

rxadc_ana3

Offset Address: 0x11c

Bits	Name	Access	Description	Reset
0	reg_addi_rxadc	R/W	ADC opamp current +50%, active High	0x0
1	reg_cksel_rxadc	R/W	PGA enable control input, active High	0x0
2	reg_en_asar_i_rxadc	R/W	Enable pin of L channel quantizer	0x1
3	reg_en_asar_q_rxadc	R/W	Enable pin of Q channel quantizer	0x1
5:4	reg_dem_type_rxadc	R/W	DEM TYPE 0:rotation 1: min cell switching 2/3: NA	0x1
7:6	Reserved			
11:8	reg_ctune_rxadc	R/W	RXADC integrator CFB selection 12*135fF + (8/4/2/1)*135fF	0xc
12	reg_en_dither_rxadc	R/W	Enable pin of dithering	0x1
13	reg_rstsdm_rxadc	R/W	Enable pin of resetting integrator	0x0
14	reg_en_vcmt_rxadc	R/W	?	0x0
15	Reserved			
17:16	reg_vldo0p9_rxadc	R/W	0.9V LDO output selection 00:0.85V 01:0.9V 10:0.95V 11:1.0V	0x1
19:18	reg_vldo12_rxadc	R/W	1.2V LDO output selection 00:1.1V 01:1.15V 10:1.2V 11:1.25V	0x1
21:20	reg_rnlvl_rxadc	R/W	Dither option	0x0
31:22	Reserved			

rxadc_ana4

Offset Address: 0x120

Bits	Name	Access	Description	Reset
0	reg_da_en_rxpga_status	RO	DA_EN_RXPGA status	
1	reg_da_end2us_rxpga_status	RO	DA_END2US_RXPGA status	
2	reg_da_en_rxadc_status	RO	DA_EN_RXADC status	
3	reg_da_en_audbias_status	RO	DA_EN_AUDBIAS status	
15:4	Reserved			
18:16	reg_ad_dol_rxadc	RO	Left channel 3-bits output	
19	Reserved			
22:20	reg_ad_dor_rxadc	RO	Right channel 3-bits output	
31:23	Reserved			

11.2.3.3 Audio DAC Register Description

txdac_ctrl0

Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	reg_txdac_en	R/W	audio dac enable	0x0
1	reg_i2s_rx_en	R/W	audio dac i2s output enable	0x0
31:2	Reserved			

txdac_ctrl1

Offset Address: 0x004

Bits	Name	Access	Description	Reset
1:0	reg_txdac_cic_opt	R/W	CIC decimation filter option 0: upsample ratio 64 1: upsample ratio 128 2: upsample ratio 256 3: upsample ratio 512	0x0
3:2	Reserved			
5:4	reg_txdac_dem_type	R/W	DEM TYPE 0:rotation 1: min cell switching 2/3: thermal code	0x1
7:6	Reserved			
8	reg_txdac_dsm_opt	R/W	DSM order option 0: order2 1: order1	0x0
9	reg_txdac_zcd_en	R/W	enable zero corssing function move from reg_spare0	0x0
10	reg_txdac_fsm_src_sel	R/W	FSM trigger source 0: i2s1_vld 1: i2s0_vld	0x1
11	Reserved			
14:12	reg_txdac_dither_opt	R/W	Dither option 0: disable others: weight = LSB/(2^(n-1))	0x0

Bits	Name	Access	Description	Reset
31:15	Reserved			

txdac_status

Offset Address: 0x008

Bits	Name	Access	Description	Reset
2:0	reg_txdac_fsm	RO	DAC main fsm state	
3	Reserved			
6:4	reg_txdac_afe_fsm	RO	DAC AFE fsm state	
7	Reserved			
16:8	reg_txdac_gain0	RO	DAC L-channel gain	
19:17	Reserved			
28:20	reg_txdac_gain1	RO	DAC R-channel gain	
31:29	Reserved			

txdac_afe0

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
5:0	reg_txdac_init_dly_cnt	R/W	DAC AFE initialize delay, min>2us	0x28
7:6	Reserved			
15:8	reg_txdac_value_tick	R/W	DAC AFE tick value for initialize/stop value ramp	0x0
27:16	reg_txdac_gain_tick	R/W	DAC AFE tick value for gain ramp	0x800
31:28	Reserved			

txdac_afe1

Offset Address: 0x010

Bits	Name	Access	Description	Reset
8:0	reg_txdac_gain_ub_0	R/W	channel 0, L-channel, DAC gain upper bound 0x00: gain=0 0xff: gain=1-2^(-8), default value to prevent DSM overflow 0x100: gain=1	0xff
15:9	Reserved			
24:16	reg_txdac_gain_ub_1	R/W	channel 1, R-channel, DAC gain upper bound 0x00: gain=0 0xff: gain=1-2^(-8), default value to prevent DSM overflow 0x100: gain=1	0xff
27:25	Reserved			
28	reg_txdac_ramp_bp	R/W	bypass initial ramp procedure	0x0
31:29	Reserved			

txdac_ana0

Offset Address: 0x020

Bits	Name	Access	Description	Reset
0	reg_addi_txdac	R/W	na	0x0
3:1	Reserved			
5:4	reg_tsel_txdac	R/W	2'b00: NA 2'b01: VCM 2'b10: VDD15A 2'b11: undefined	0x0
7:6	Reserved			
9:8	reg_vsel_txdac	R/W	1.5V LDO output selection 2'b00:1.35V 2'b01:1.4V 2'b10:1.45V 2'b11:1.5V	0x3
31:10	Reserved			

txdac_ana1

Offset Address: 0x024

Bits	Name	Access	Description	Reset
0	reg_da_en_txdac_ow_val	R/W	DA_EN_TXDAC overwrite value	0x0
1	reg_da_end2us_txdac_ow_val	R/W	DA_END2US_TXDAC overwrite value	0x0
15:2	Reserved			
16	reg_da_en_txdac_ow_en	R/W	DA_EN_TXDAC overwrite enable	0x0
17	reg_da_end2us_txdac_ow_en	R/W	DA_END2US_TXDAC overwrite enable	0x0
31:18	Reserved			

12 Peripherals equipment

12.1 I2C

12.1.1 Overview

There are five I2C controllers in the chip. Each I2C controller can be individually configured as Master / Slave. For IO configuration, please refer to chapter 2.2 pin information description.

12.1.2 Function description

I2C controller has the following features:

- Support standard address (7bit) and extended address (10bit).
- The transmission rate supports standard mode (100kbit/S) and fast mode (400kbits/S).
- Support the functions of general call and start byte.
- It does not support CBUS devices.
- Support DMA operation.
- Support 64X8bits TX FIFO and 64X8bits RX FIFO.

12.1.3 Function block diagram

Figure 12-1 shows the functional block diagram of I2C module. IIC_CLK is the module clock, and could be 25MHz or 100MHz. CPU configures registers through APB bus to select I2C modes and timing, writes TXFIFO, reads RXFIFO, and triggers FSM to send and receive SDA / SCL related IO signals. System DMA can also be used with I2C DMA_IF, and write TXFIFO and read RXFIFO through APB bus to send and receive I2C signal.

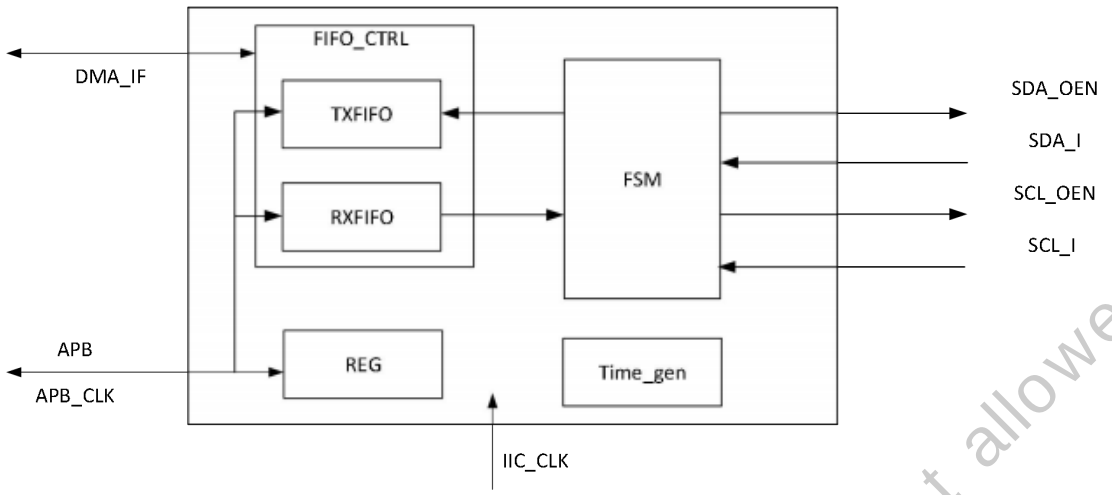


Figure 12-1 I2C function block diagram

12.1.4 I2C Agreement timing

The I2C protocol timing of chip I2C supporting general standards is shown in Figure 12-2.

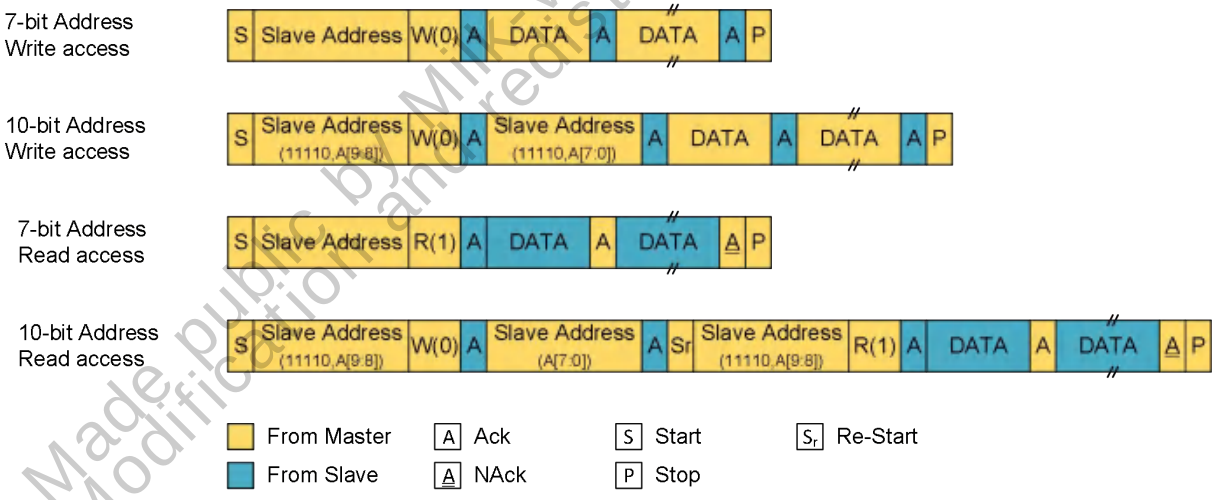


Figure 12-2 I2C protocol timing

12.1.5 Working mode

12.1.5.1 Configure I2C clock and timing parameters

1. Refer to the register chapter of CLK_DIV CRG , configure clk_byp_0_31 to select 25MHz or 100MHz to be clock source of IIC_CLK.
2. Setting related timing configuration should be in controller disable state. The IC_ENABLE needs to be set 0 and query IC_ENABLE_Status [0] confirmed to be 0.
3. Refer to table 12-1, according to the frequency of IIC_CLK to config controller registers.

Table 12-1 Relationship between I2C clock selection and related register configuration

Register	25M IIC_CLK	100M IIC_CLK	Description
IC_SS_SCL_HCNT	115	460	SCL high level time counting in standard speed mode
IC_SS_SCL_LCNT	135	540	SCL low level time counting in standard speed mode
IC_FS_SCL_HCNT	21	90	SCL high level time counting in fast speed mode
IC_FS_SCL_LCNT	42	160	SCL low level time counting in fast speed mode
IC_SDA_HOLD	1	1	SDA holding time count, relative to the negative edge of SCL
IC_SDA_SETUP	6	25	SDA set time count, relative to positive edge of SCL
IC_FS_SPKLEN	2	5	I2C burr suppression time count

12.1.5.2 Data transmission in non DMA mode

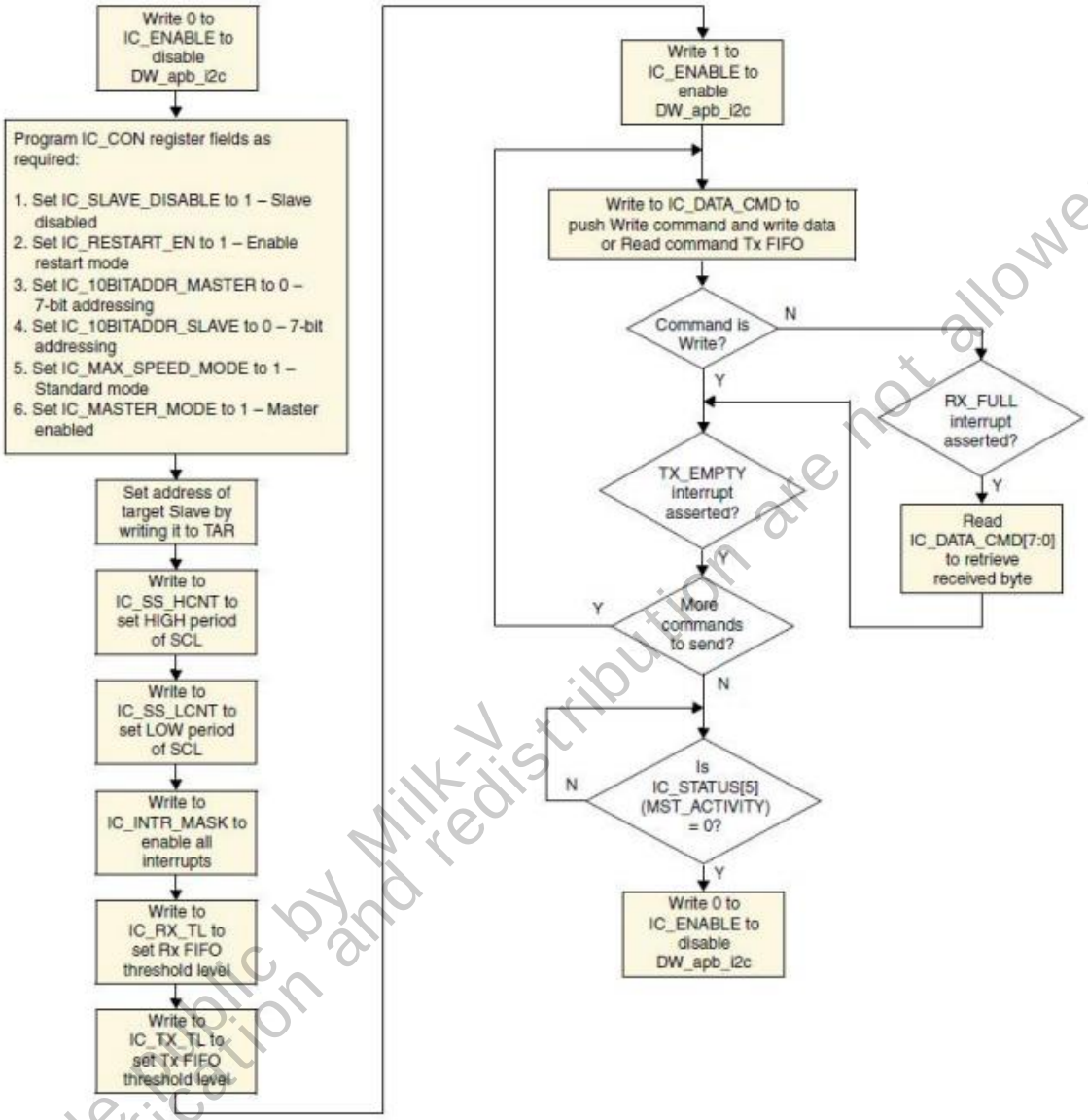


Figure 12-3 data transmission software flow in I2C non DMA mode

12.1.5.3 Data transmission in DMA mode

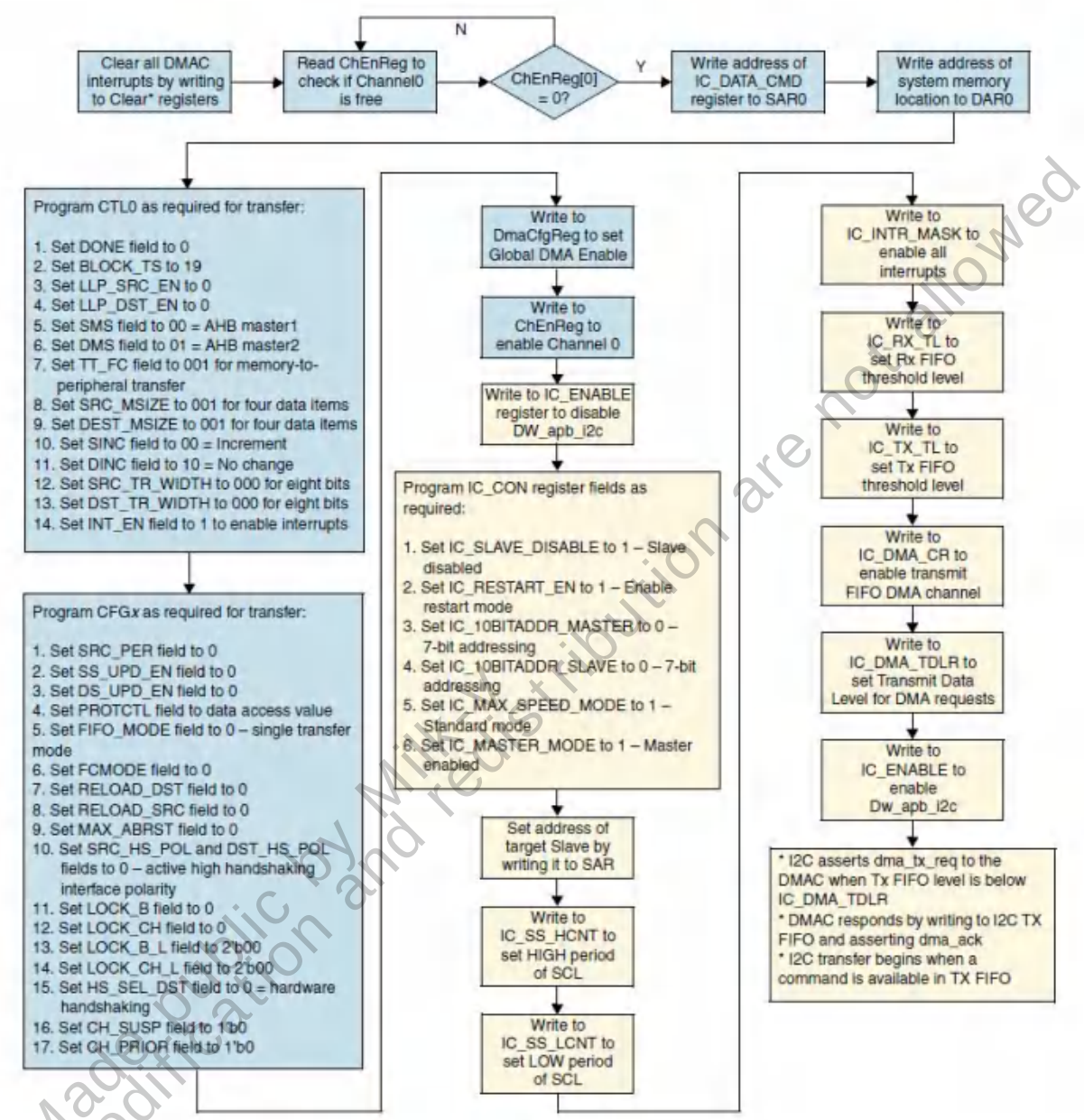


Figure 12-4 data transmission software flow under I2C DMA mode

12.1.6 I2C register overview

Base addresses of the 6 group I2C modules on the chip.

GPIO Module	Base Address
I2C0	0x04000000
I2C1	0x04010000
I2C2	0x04020000
I2C3	0x04030000
I2C4	0x04040000
RTCSYS_I2C	0x0502B000

I2C register overview of chip

Name	Address Offset	Description
IC_CON	0x000	I2C Control
IC_TAR	0x004	I2C Target Address
IC_SAR	0x008	I2C Slave Address
IC_DATA_CMD	0x010	I2C Rx/Tx Data Buffer and Command
IC_SS_SCL_HCNT	0x014	Standard speed I2C Clock SCL High Count
IC_SS_SCL_LCNT	0x018	Standard speed I2C Clock SCL Low Count
IC_FS_SCL_HCNT	0x01c	Fast speed I2C Clock SCL High Count
IC_FS_SCL_LCNT	0x020	Fast speed I2C Clock SCL Low Count
IC_INTR_STAT	0x02c	I2C Interrupt Status
IC_INTR_MASK	0x030	I2C Interrupt Mask
IC_RAW_INTR_STAT	0x034	I2C Raw Interrupt Status
IC_RX_TL	0x038	I2C Receive FIFO Threshold
IC_TX_TL	0x03c	I2C Transmit FIFO Threshold
IC_CLR_INTR	0x040	Clear Combined and Individual Interrupts
IC_CLR_RX_UNDER	0x044	Clear RX_UNDER Interrupt
IC_CLR_RX_OVER	0x048	Clear RX_OVER Interrupt
IC_CLR_TX_OVER	0x04c	Clear TX_OVER Interrupt
IC_CLR_RD_REQ	0x050	Clear RD_REQ Interrupt
IC_CLR_TX_ABRT	0x054	Clear TX_ABRT Interrupt
IC_CLR_RX_DONE	0x058	Clear RX_DONE Interrupt
IC_CLR_ACTIVITY	0x05c	Clear ACTIVITY Interrupt
IC_CLR_STOP_DET	0x060	Clear STOP_DET Interrupt
IC_CLR_START_DET	0x064	Clear START_DET Interrupt
IC_CLR_GEN_CALL	0x068	Clear GEN_CALL Interrupt
IC_ENABLE	0x06c	I2C Enable
IC_STATUS	0x070	I2C Status register
IC_TXFLR	0x074	Transmit FIFO Level Register
IC_RXFLR	0x078	Receive FIFO Level Register
IC_SDA_HOLD	0x07c	SDA hold time length register
IC_TX_ABRT_SOURCE	0x080	I2C Transmit Abort Status Register
IC_SLV_DATA_NACK_ONLY	0x084	Generate SLV_DATA_NACK Register
IC_DMA_CR	0x088	DMA Control Register for transmit and receive handshaking interface
IC_DMA_TDLR	0x08c	DMA Transmit Data Level
IC_DMA_RDLR	0x090	DMA Receive Data Level
IC_SDA_SETUP	0x094	I2C SDA Setup Register
IC_ACK_GENERAL_CALL	0x098	I2C ACK General Call Register
IC_ENABLE_STATUS	0x09c	I2C Enable Status Register
IC_FS_SPKLEN	0x0a0	ISS and FS spike suppression limit
IC_HS_SPKLEN	0x0a4	HS spike suppression limit

12.1.7 I2C register description

IC_CON

Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	MASTER_MODE	R/W	enable master mode	0x1
2:1	SPEED	R/W	1: standard mode (~100 kbit/s) 2: fast mode (~400 kbit/s)	0x3
3	IC_10BITADDR_SLAVE	R/W	enable 10bit slave address mode	0x1
4	IC_10BITADDR_MASTER	R/W	enable 10bit master address mode	0x1
5	IC_RESTART_EN	R/W	enable I2C master to be able generate restart	0x1
6	IC_SLAVE_DISABLE	R/W	0: slave is enabled 1: slave is disabled	0x1
31:7	Reserved			

IC_TAR

Offset Address: 0x004

Bits	Name	Access	Description	Reset
9:0	IC_TAR	R/W	I2C Target Address Register	0x55
10	GC_OR_START	R/W	If bit SPECIAL is set to 1, then this bit indicates whether a General Call or START byte command 0: general call 1: start byte	0x0
11	SPECIAL	R/W	Used to issue General Call or START BYTE	0x0
31:12	Reserved			

IC_SAR

Offset Address: 0x008

Bits	Name	Access	Description	Reset
9:0	IC_SAR	R/W	I2C Slave Address Register	0x55
31:10	Reserved			

IC_DATA_CMD

Offset Address: 0x010

Bits	Name	Access	Description	Reset
7:0	DAT	R/W	transmitted or received data port	0x0
8	CMD	R/W	0: Write 1: Read	0x0
9	STOP	R/W	issue stop	0x0
10	RESTART	R/W	issue restart	0x0
31:11	Reserved			

IC_SS_SCL_HCNT

Offset Address: 0x014

Bits	Name	Access	Description	Reset
15:0	IC_SS_SCL_HCNT	R/W	Standard Speed I2C Clock SCL High Count Register	0x0190
31:16	Reserved			

IC_SS_SCL_LCNT

Offset Address: 0x018

Bits	Name	Access	Description	Reset
15:0	IC_SS_SCL_LCNT	R/W	Standard Speed I2C Clock SCL Low Count Register	0x01d6
31:16	Reserved			

IC_FS_SCL_HCNT

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
15:0	IC_FS_SCL_HCNT	R/W	Fast Speed I2C Clock SCL High Count Register	0x003C
31:16	Reserved			

IC_FS_SCL_LCNT

Offset Address: 0x020

Bits	Name	Access	Description	Reset
15:0	IC_FS_SCL_LCNT	R/W	Fast Speed I2C Clock SCL Low Count Register	0x0082
31:16	Reserved			

IC_INTR_STAT

Offset Address: 0x02c

Bits	Name	Access	Description	Reset
0	R_RX_UNDER	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
1	R_RX_OVER	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
2	R_RX_FULL	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
3	R_TX_OVER	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
4	R_TX_EMPTY	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
5	R_RD_REQ	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
6	R_TX_ABRT	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
7	R_RX_DONE	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt	

Bits	Name	Access	Description	Reset
			Status	
8	R_ACTIVITY	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
9	R_STOP_DET	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
10	R_START_DET	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
11	R_GEN_CALL	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
31:12	Reserved			

IC_INTR_MASK

Offset Address: 0x030

Bits	Name	Access	Description	Reset
0	M_RX_UNDER	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x1
1	M_RX_OVER	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x1
2	M_RX_FULL	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x1
3	M_TX_OVER	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x1
4	M_TX_EMPTY	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x1
5	M_RD_REQ	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x1
6	M_TX_ABRT	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x1
7	M_RX_DONE	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x1
8	M_ACTIVITY	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x0
9	M_STOP_DET	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x0
10	M_START_DET	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x0
11	M_GEN_CALL	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x1
31:12	Reserved			

IC_RAW_INTR_STAT

Offset Address: 0x034

Bits	Name	Access	Description	Reset
0	IST_RX_UNDER	RO	when receive buffer is empty by reading from the IC_DATA_CMD register	
1	IST_RX_OVER	RO	receive buffer is oveflow (64Bytes)	
2	IST_RX_FULL	RO	receive buffer reaches or goes above the RX_TL threshold	
3	IST_TX_OVER	RO	transmit buffer is oveflow (64Bytes)	
4	IST_TX_EMPTY	RO	transmit buffer is at or below the TX_TL threshold	
5	IST_RD_REQ	RO	In slave mode, I2C hold SCL and wait for the response from processor	
6	IST_TX_ABRT	RO	In master or slave mode, when transmitter is unable to complete the action	
7	IST_RX_DONE	RO	In slave-transmitter mode, a NACK is received	
8	IST_ACTIVITY	RO	I2C activity is detected	
9	IST_STOP_DET	RO	STOP occurred	
10	IST_START_DET	RO	START or RESTART occurred	
11	IST_GEN_CALL	RO	General Call address is received	
31:12	Reserved			

IC_RX_TL

Offset Address: 0x038

Bits	Name	Access	Description	Reset
7:0	RX_TL	R/W	Receive FIFO Threshold Level	0x0
31:8	Reserved			

IC_TX_TL

Offset Address: 0x03c

Bits	Name	Access	Description	Reset
7:0	TX_TL	R/W	Transmit FIFO Threshold Level	0x0
31:8	Reserved			

IC_CLR_INTR

Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	CLR_INTR	RO	read to clear corresponding all raw staus	
31:1	Reserved			

IC_CLR_RX_UNDER

Offset Address: 0x044

Bits	Name	Access	Description	Reset
0	CLR_RX_UNDER	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	

Bits	Name	Access	Description	Reset
31:1	Reserved			

IC_CLR_RX_OVER

Offset Address: 0x048

Bits	Name	Access	Description	Reset
0	CLR_RX_OVER	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	
31:1	Reserved			

IC_CLR_TX_OVER

Offset Address: 0x04c

Bits	Name	Access	Description	Reset
0	CLR_TX_OVER	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	
31:1	Reserved			

IC_CLR_RD_REQ

Offset Address: 0x050

Bits	Name	Access	Description	Reset
0	CLR_RD_REQ	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	
31:1	Reserved			

IC_CLR_TX_ABRT

Offset Address: 0x054

Bits	Name	Access	Description	Reset
0	CLR_TX_ABRT	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	
31:1	Reserved			

IC_CLR_RX_DONE

Offset Address: 0x058

Bits	Name	Access	Description	Reset
0	CLR_RX_DONE	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	
31:1	Reserved			

IC_CLR_ACTIVITY

Offset Address: 0x05c

Bits	Name	Access	Description	Reset
0	CLR_ACTIVITY	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	

Bits	Name	Access	Description	Reset
31:1	Reserved			

IC_CLR_STOP_DET

Offset Address: 0x060

Bits	Name	Access	Description	Reset
0	CLR_STOP_DET	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	
31:1	Reserved			

IC_CLR_START_DET

Offset Address: 0x064

Bits	Name	Access	Description	Reset
0	CLR_START_DET	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	
31:1	Reserved			

IC_CLR_GEN_CALL

Offset Address: 0x068

Bits	Name	Access	Description	Reset
0	CLR_GEN_CALL	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	
31:1	Reserved			

IC_ENABLE

Offset Address: 0x06c

Bits	Name	Access	Description	Reset
0	ENABLE	R/W	Enables I2C controller	0x0
31:1	Reserved			

IC_STATUS

Offset Address: 0x070

Bits	Name	Access	Description	Reset
0	ST_ACTIVITY	RO	I2C Activity Status.	
1	ST_TFNF	RO	Transmit FIFO Not Full	
2	ST_TFE	RO	Transmit FIFO Completely Empty	
3	ST_RFNE	RO	Receive FIFO Not Empty	
4	ST_RFF	RO	Receive FIFO Completely Full	
5	ST_MST_ACTIVITY	RO	Master FSM Activity Status	
6	ST_SLV_ACTIVITY	RO	Slave FSM Activity Status	
31:7	Reserved			

IC_TXFLR

Offset Address: 0x074

Bits	Name	Access	Description	Reset
6:0	TXFLR	RO	I2C Transmit FIFO Level	
31:7	Reserved			

IC_RXFLR

Offset Address: 0x078

Bits	Name	Access	Description	Reset
6:0	RXFLR	RO	I2C Receive FIFO Level Register	
31:7	Reserved			

IC_SDA_HOLD

Offset Address: 0x07c

Bits	Name	Access	Description	Reset
15:0	IC_SDA_HOLD	R/W	Sets the required SDA hold time in units of IP clock.	0x1
31:16	Reserved			

IC_TX_ABRT_SOURCE

Offset Address: 0x080

Bits	Name	Access	Description	Reset
15:0	TX_ABRT_SOURCE	RO	I2C Transmit Abort Source Register	
31:16	Reserved			

IC_SLV_DATA_NACK_ONLY

Offset Address: 0x084

Bits	Name	Access	Description	Reset
0	NACK	R/W	generate a NACK in slave-receiver mode	0x0
31:1	Reserved			

IC_DMA_CR

Offset Address: 0x088

Bits	Name	Access	Description	Reset
0	RDMAE	R/W	Receive DMA Enable	0x0
1	TDMAE	R/W	Transmit DMA Enable	0x0
31:2	Reserved			

IC_DMA_TDLR

Offset Address: 0x08c

Bits	Name	Access	Description	Reset
5:0	DMATDL	R/W	the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value	0x0
31:6	Reserved			

IC_DMA_RDLR

Offset Address: 0x090

Bits	Name	Access	Description	Reset
5:0	DMARDL	R/W	dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or more than this field value + 1	0x0
31:6	Reserved			

IC_SDA_SETUP

Offset Address: 0x094

Bits	Name	Access	Description	Reset
0	SDA_SETUP	R/W	SDA Setup time config register	0x64
31:1	Reserved			

IC_ACK_GENERAL_CALL

Offset Address: 0x098

Bits	Name	Access	Description	Reset
0	ACK_GEN_CALL	R/W	When set to 1, DW_apb_i2c responds with a ACK when it receives a General Call. When set to 0, the IP does not generate General Call interrupts	0x1
31:1	Reserved			

IC_ENABLE_STATUS

Offset Address: 0x09c

Bits	Name	Access	Description	Reset
0	IC_EN	RO	I2C Enable Status Register	
1	SLV_DISABLED_WHILE_BUSY	RO	Slave Disabled While Busy (Transmit, Receive)	
2	SLV_RX_DATA_LOST	RO	Slave Received Data Lost.	
31:3	Reserved			

IC_FS_SPKLEN

Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
7:0	IC_FS_SPKLEN	R/W	I2C SS and FS Spike Suppression Limit Register	0x5
31:8	Reserved			

IC_HS_SPKLEN

Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
7:0	IC_HS_SPKLEN	R/W	I2C HS Spike Suppression Limit Register	0x1
31:8	Reserved			

12.2 UART

12.2.1 Overview

UART (Universal Asynchronous Receiver Transmitter) is an asynchronous serial communication interface. Its main function is to transfer data from peripheral devices into internal bus after serial parallel conversion, and output data to external devices after parallel serial conversion. The main function of UART is to connect with UART of external chip, so as to realize the communication between two chips.
This chip provides 5 UART controllers. The relevant overview is shown in the table below. For IO configuration, please refer to chapter 2.2 pin information description.

Controller	Support mode	IO pin
UART0	Two line UART	UART0_TX/UART0_RX
UART1	two/four line UART	UART1_TX/UART1_RX/UART1_CTS/UART1_RTS
		XGPIOA[20]/ XGPIOA[21]/ XGPIOA[22]/ XGPIOA[26]
UART2	two/four line UART	UART2_TX/UART2_RX/UART2_CTS/UART2_RTS
		XGPIOA[20]/ XGPIOA[21]/ XGPIOA[22]/ XGPIOA[26]
		IIC2_SDA/IIC2_SCL
UART3	two/four line UART	SPI0_CS_X/SPI0_SCK/SPI0_SDI/SPI0_SDO
		VI_DATA22/VI_DATA21/VI_DATA24/VI_DATA23
		PWM3/PWM2
UART4	Two UART	XGPIOA[22]/ XGPIOA[26]
		UART1_RTS/UART1_CTS

12.2.2 Characteristics

UART module has the following characteristics:

- UART module has the following characteristics:
- Support 32 x 8bit transmit FIFO and 32 x 8bit receive FIFO.

- Support the programmable bit width of data bit and stop bit. Data bits can be programmed to 5 / 6 / 7 / 8 bits;
- The stop bit can be set to 1 bit, 1.5 bit or 2 bit by programming.
- Support odd, even or no check.
- Support the programmable transmission rate.
- Support receiving FIFO interrupt, sending FIFO interrupt and error interrupt.
- Support initial interrupt status query and post mask interrupt status query.
- Support DMA operation.

12.2.3 Function description

12.2.3.1 Application diagram

UART is a general point-to-point physical layer transport protocol, which can be used to connect various systems, including PC and various peripheral chips, and can be used as the communication interface between chips.

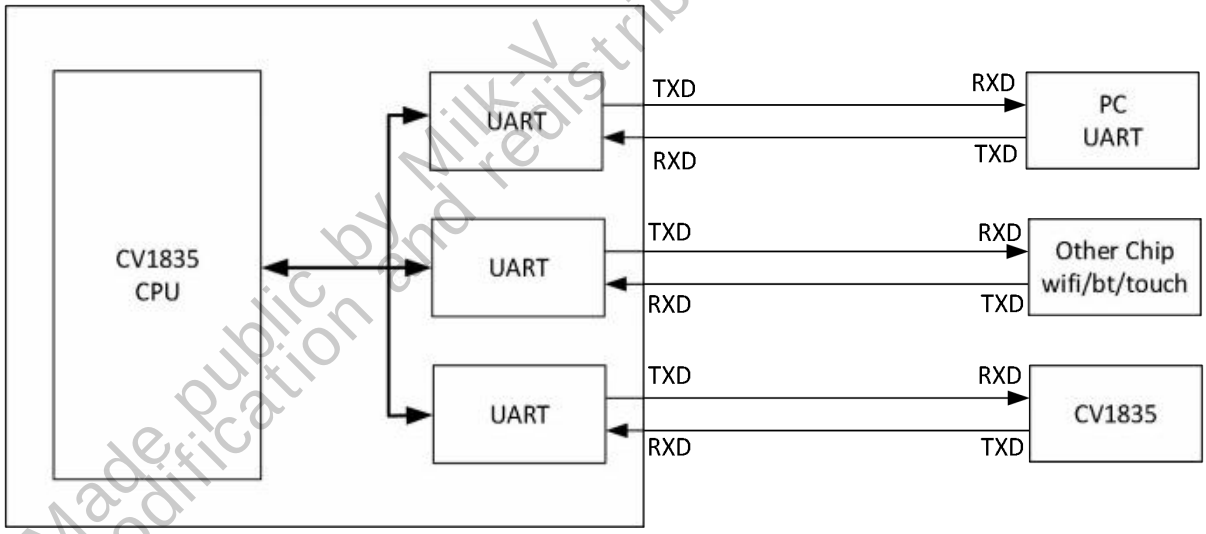


Figure 12-5 UART application block diagram

12.2.3.2 Function principle

- Baud rate

Since UART interface has no reference clock and belongs to asynchronous transmission mode, both sides need to use the same transmission speed, that is, the baud rate, to communicate. If there is any error, the error rate should be small enough to avoid

misinformation. The rate of one bit is called the baud rate. Typical baud rates are 300, 1200, 2400, 9600, 19200, 38400, 115200 bps, etc.

- Frame structure

The data structure of UART transmission is in frame. The frame structure includes start signal, data signal, check bit and end signal.

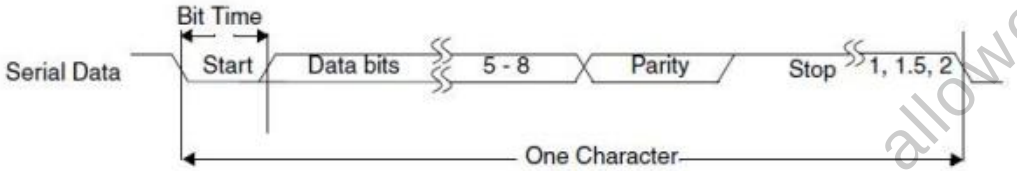


Figure 12-6 UART transmission data structure

- Start bit

The start signal is the mark of the beginning of a frame. The beginning of starting a frame transmission is to send a low level signal bit on the TXD. In RXD, if a low level signal bit is received in idle state, it is judged as the beginning of receiving a detection transmission

- Data bit

The data bit width can be adjusted according to different application requirements, which can be 5 / 6 / 7 / 8 bit data bit width. The typical data width is 8 bits

- Parity bit

The check bit is a 1-bit error correction signal. The check bit of UART includes odd parity check, even parity check and fixed check bit. At the same time, it supports the enable and disable of check bit. Please refer to LCR register for detailed description

- Stop bit

The end signal is the stop bit of the frame. It supports 1 bit, 1.5 bit and 2 bit stop bits. Sending the end signal of a frame is to send the TXD to high level to complete the transmission and enter the idle state. After counting the check bits of a received frame, the end signal needs to be received.

12.2.4 Working mode

12.2.4.1 Baud rate configuration

- UART working clock (UART_SCLK) configuration

Please refer to the CLK_DIV CRG register description to configure clk_sel_0_9~ clk_sel_0_13 and select the working clock of uart0~uart4. The default setting is 1:XTAL 25MHz , and could be configurated to 0:187.5mhz, if necessary, frequency division register div_clk_187p5m can be configured, adjust the clock to 1500/N MHz, up to 187.5 MHz.

- UART baud rate configuration

DLL and DLH are the baud rate frequency division control registers in UART controller. DLH is the high 8 bits and DLL is the low 8 bits. Before configuring DLH and DLL, LCR [7] must be set to 1. The RBR_THR_DLL(DLL) register and IER_DLH(DLH)register can be configured.

After configuration, baud rate is set. The formula is：

$$Baud\ rate = {UART\_SCLK}/{16 * (256 * DLH + DLL)}$$

- Take UART SCLK 25MHz as an example, 115200 baud rate is configured, and the formula is：

$$(256 * DLH + DLL) = {25M}/{16 * 115200} = 13.5$$

If DLL is 14 and DLH is 0, the actual baud rate is ：

$$Baud\ rate = {25M}/{16 * 14} = 111607$$

One bit time error is：

$$Bit\ Error = {(115200 - 114286)}/{115200} = 3.12\%$$

The accumulated time error of one frame is：

$$Frame\ Error = 3.12\% * 10 = 31.2\%$$

- Take UART SCLK 187.5MHz as an example, 115200 baud rate is configured, and the formula is：

$$(256 * DLH + DLL) = {187.5M}/{16 * 115200} = 101.7$$

If DLL is 102 and DLH is 0, the actual baud rate is ：

$$Baud\ rate = {187.5M}/{16 * 102} = 114890$$

One bit time error is：

$$Bit\ Error = {(115200 - 114890)}/{115200} = 0.27\%$$

The accumulated time error of one frame is：

$$Frame\ Error = 3.12\%*10 = 2.7\%$$

12.2.4.2 Data transmission flow chart

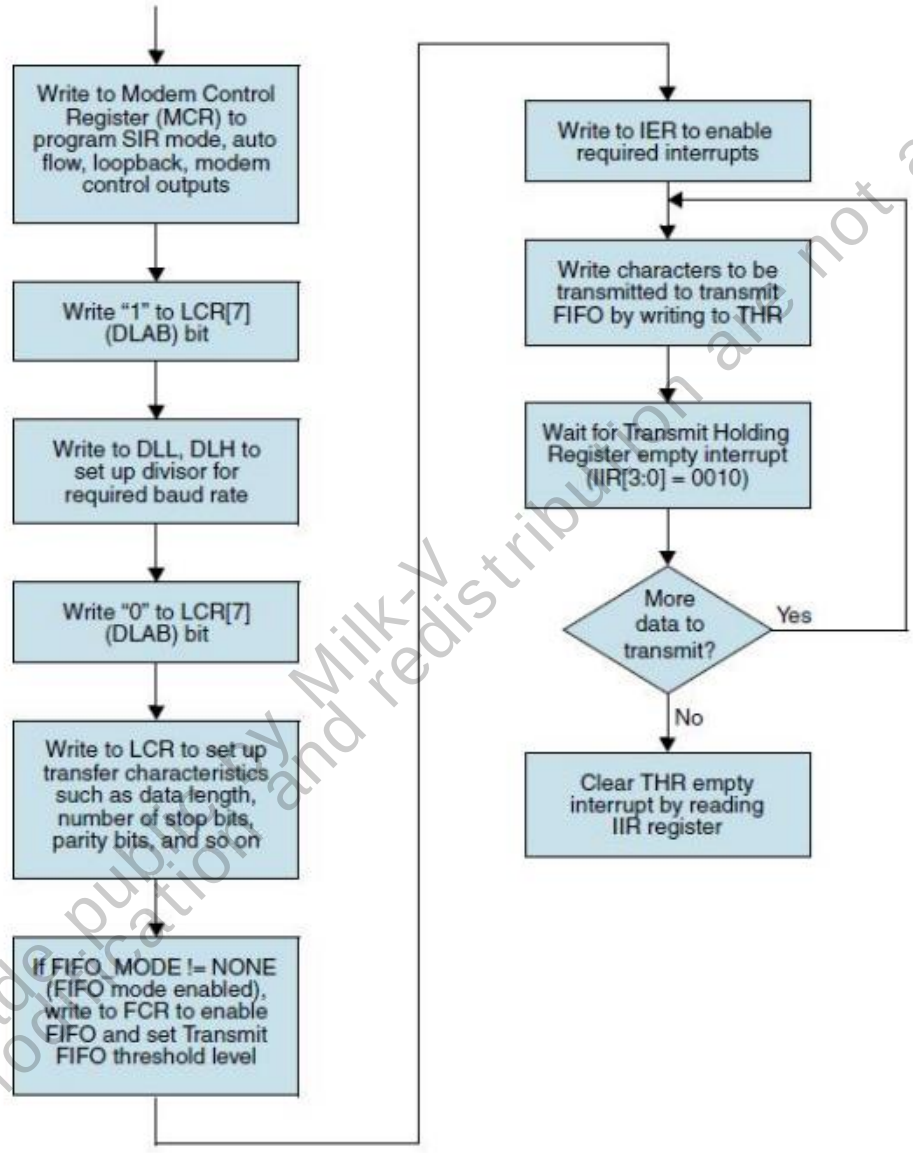


Figure 12-7 UART data transmission flow chart

12.2.4.3 Data receiving flow chart

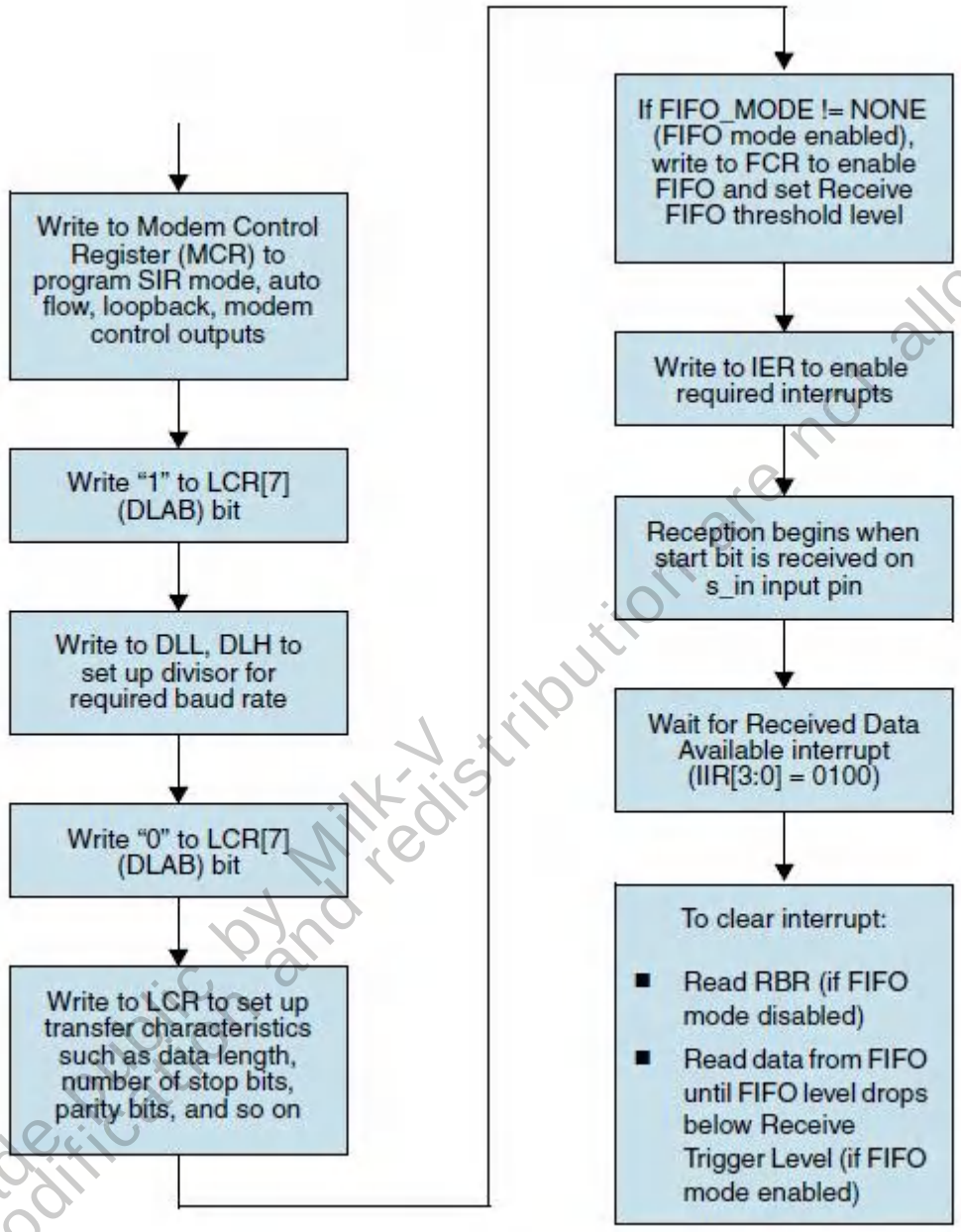


Figure 12 8 UART data receiving flow chart

12.2.4.4 Data transmission in interrupt or query mode

12.2.4.4.1 Initialization steps

1. Write 1 to LCR [7]. Enable and configure division latch access
2. Write the corresponding configuration value to RBR_THR_DLL、IER_DLH register, configure the baud rate of transmission.
3. Write 0 to LCR [7].
4. Configure LCR and set corresponding UART working mode.
5. Configure FCR and set corresponding transmit and receive FIFO threshold.
6. If interrupt mode is used, IER should be set to enable corresponding interrupt signal.

12.2.4.4.2 Data transmission

1. When LCR [7] is 0, the transmitted data could be written into RBR_THR_DLL (transmit holding register) to start data transmission.
2. If query mode is used, TX is detected by reading USR [1] (transmit FIFO not full) and TFL (transmit FIFO level) to detect the status of TX_FIFO, according to the status of TX_FIFO determines whether to continue to write data to RBR_THR_DLL.
3. If interrupt mode is used, detect according to corresponding interrupt status bit, and decide whether to continue to write data to RBR_THR_DLL.
4. By detecting USR [2] (Transmit FIFO Empty), judge whether UART completes all data transmission.

12.2.4.4.3 Data reception

1. If query mode is used, the status of RX_FIFO is detected by reading USR [3] (receive FIFO not empty) and RFL (receive FIFO level), according to the status

of RX_FIFO to determine whether to read RBR_THR_DLL (receive buffer register) to obtain data.

2. If the interrupt mode is used, whether to read RBR_THR_DLL(Receive Buffer Register) is determined according to the corresponding interrupt status bit detection to obtain data.

12.2.4.5 Data transmission in DMA mode

12.2.4.5.1 Initialization steps

1. Write 1 to LCR [7]. Enable and configure Divisor Latch Access
2. Write the corresponding configuration value to RBR_THR_DLL、IER_DLH register, configure the baud rate of transmission.
3. Write 0 to LCR [7].
4. Configure LCR and set corresponding UART working mode
5. Configure FCR and set corresponding transmit and receive FIFO threshold.
6. Turn off ETBEI/ERBFI in IER

12.2.4.5.2 Data transmission

1. Configure system DMA channel mapping. Refer to 3.5.2.4 system DMA channel mapping, configure the selected UART controller TX / RX request line number to the corresponding system DMA channel. For example, If we want mapping UART0 TX to DMA channel 3, we should configurate sdma_dma_ch_remap0[29:24] to 9 and write 1 to update_dma_remp_0_3 to make the configuration effective.
2. Configure DMA data channel, including data transmission source and destination address, number of data transmission, transmission type and other parameters. Please refer to chapter 3.6 DMA controller for specific configuration.

3. Judge whether the data is sent through DMA interrupt report.

12.2.4.5.3 Data reception

1. Configure system DMA channel mapping. Refer to 3.5.2.4 system DMA channel mapping, configure the selected UART controller TX / RX request line number to the corresponding system DMA channel. For example, If we want mapping UART0 RX to DMA channel 1, we should configurate sdma_dma_ch_remap0[13:8] to 8 and write 1 to update_dma_remp_0_3 to make the configuration effective.
2. Configure DMA data channel, including data transmission source and destination address, data receiving area address, data transmission number, transmission type and other parameters. Please refer to chapter 3.6 DMA controller for specific configuration.
3. Judge whether the data is received through DMA interrupt report.

12.2.5 UART register overview

Base addresses of the 6 group UART modules on the chip

GPIO module	Base Address
UART0	0x04140000
UART1	0x04150000
UART2	0x04160000
UART3	0x04170000
UART4	0x041C0000
RTCSYS_UART	0x05022000

UART register overview of the chip

Name	Address Offset	Description
RBR_THR_DLL	0x000	Receive Buffer,Transmit Holding or Divisor Latch Low byte Register
IER_DLH	0x004	Interrupt Enable or Divisor Latch high byte Register
FCR_IIR	0x008	FIFO Control or Interrupt Identification Register
LCR	0x00c	Line Control Register
MCR	0x010	Modem Control Register

Name	Address Offset	Description
LSR	0x014	Line Status Register
MSR	0x018	Modem Status Register
LPDLL	0x020	Low Power Divisor Latch (Low) Register
LPDLH	0x024	Low Power Divisor Latch (High) Register
SRBR_STHR	0x030	Shadow Receive/Trasnmit Buffer Register
FAR	0x070	FIFO Access Register
TFR	0x074	Transmit FIFO Read
RFW	0x078	Receive FIFO Write
USR	0x07c	UART Status Register
TFL	0x080	Transmit FIFO Level
RFL	0x084	Receive FIFO Level
SRR	0x088	Software Reset Register
SRTS	0x08c	Shadow Request to Send
SBCR	0x090	Shadow Break Control Register
SDMAM	0x094	Shadow DMA Mode
SFE	0x098	Shadow FIFO Enable
SRT	0x09c	Shadow RCVR Trigger
STET	0x0a0	Shadow TX Empty Trigger
HTX	0x0a4	Halt TX
DMASA	0x0a8	DMA Software Acknowledge

12.2.6 UART register description

RBR_THR_DLL

Offset Address: 0x000

Bits	Name	Access	Description	Reset
7:0	RBR_THR_DLL	R/W	LCR[7] bit = 0 ： (R) Receive Buffer Register ,Data byte received on the serial input port (W)Transmit Holding Register,Data to be transmitted on the serial output port LCR[7] bit = 1 ： Lower 8 bits of a 16-bit Divisor Latch register that contains the baud rate divisor for the UART	0x0
31:8	Reserved			

IER_DLH

Offset Address: 0x004

Bits	Name	Access	Description	Reset
7:0	IER_DLH	R/W	LCR[7] bit = 0 ： IER[0] : Enable Received Data Available Interrupt. IER[1] : Enable Transmit Holding Register Empty Interrupt. IER[2] : Enable Receiver Line	0x0

Bits	Name	Access	Description	Reset
			Status Interrupt. IER[3] : Enable Modem Status Interrupt. IER[7] : Programmable THRE Interrupt Mode Enable LCR[7] bit = 1 : Upper 8-bits of a 16-bit Divisor Latch register that contains the baud rate divisor for the UART.	
31:8	Reserved			

FCR_IIR

Offset Address: 0x008

Bits	Name	Access	Description	Reset
7:0	FCR_IIR	R/W	(R) interrupt Identification Register [3:0] Interrupt ID 0000 : modem status 0001 : no interrupt pending 0010 : THR empty 0100 : received data available 0110 : receiver line status 0111 : busy detect 1100 : character timeout [7:6] FIFOs Enabled 00 – disabled 11 – enable (W) FIFO Control Register [0] FIFO Enable [1] RCVR FIFO Reset [2] XMIT FIFO Reset [3] DMA Mode 0 – mode 0, single DMA data transfers at a time 1 – mode 1, multi DMA data transfers are made continuously [5:4] TX Empty 00 – FIFO empty 01 – 2 characters in the FIFO 10 – FIFO ¼ full 11 – FIFO ½ full [7:6] RCVR Trigger 00 – 1 character in the FIFO 01 – FIFO ¼ full 10 – FIFO ½ full 11 – FIFO 2 less than full	0x1
31:8	Reserved			

LCR

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
7:0	LCR	R/W	Line Control Register [1:0] Data Length Select. (00:5 bits,01:6 bits,10:7 bits,11:8 bits) [2] Number of stop bits. (0:1 stop	0x0

Bits	Name	Access	Description	Reset
			bit,1:1.5 stop bits when Data Length Select is 0, else 2 stop bits) [3] Parity Enable [4] Even Parity Select [5] Stick Parity [6] Break Control Bit [7] Divisor Latch Access Bit	
31:8	Reserved			

MCR

Offset Address: 0x010

Bits	Name	Access	Description	Reset
7:0	MCR	R/W	Modem Control Register [0] reserved [1] Request to Send. This is used to directly control the Request to Send (rts_n) output [2] reserved [3] reserved [4] reserved [5] Auto Flow Control Enable. [6] reserved	0x0
31:8	Reserved			

LSR

Offset Address: 0x014

Bits	Name	Access	Description	Reset
7:0	LSR	RO	Line Status Register [0] Data Ready bit. there is at least one character in the RBR or the receiver FIFO. [1] Overrun error bit. This is used to indicate the occurrence of an overrun error. [2] Parity Error bit. [3] Framing Error bit.. [4] Break Interrupt bit. [5] Transmit Holding Register Empty bit. [6] Transmitter Empty bit. [7] Receiver FIFO Error bit.	
31:8	Reserved			

MSR

Offset Address: 0x018

Bits	Name	Access	Description	Reset
7:0	MSR	RO	Modem Status Register [0] Delta Clear to Send. [1] reserved [2] reserved [3] reserved [4] CTS	

Bits	Name	Access	Description	Reset
			[5] reserved [6] reserved [7] reserved	
31:8	Reserved			

LPDLL

Offset Address: 0x020

Bits	Name	Access	Description	Reset
7:0	LPDLL	R/W	LCR[7] bit = 1 : Low Power Divisor Latch (Low) Register	0x0
31:8	Reserved			

LPDLH

Offset Address: 0x024

Bits	Name	Access	Description	Reset
7:0	LPDLH	R/W	LCR[7] bit = 1 : Low Power Divisor Latch (High) Register	0x0
31:8	Reserved			

SRBR_STHR

Offset Address: 0x030

Bits	Name	Access	Description	Reset
7:0	SRBR_STHR	R/W	LCR[7] bit = 0 : (R) Shadow Receive Buffer Register (W) Shadow Transmit Holding Register	0x0
31:8	Reserved			

FAR

Offset Address: 0x070

Bits	Name	Access	Description	Reset
0	FAR	R/W	FIFO Access Register,This register is use to enable a FIFO access mode for testing	0x0
31:1	Reserved			

TFR

Offset Address: 0x074

Bits	Name	Access	Description	Reset
7:0	TFR	R/W	Transmit FIFO Read. These bits are only valid when FIFO access mode is enabled	0x0
31:8	Reserved			

RFW

Offset Address: 0x078

Bits	Name	Access	Description	Reset
9:0	RFW	R/W	Receive FIFO Write. These bits are only valid when FIFO access mode is enabled	0x0

Bits	Name	Access	Description	Reset
			[7:0] Receive FIFO Write Data. [8] Receive FIFO Parity Error. [9] Receive FIFO Framing Error.	
31:10	Reserved			

USR

Offset Address: 0x07c

Bits	Name	Access	Description	Reset
4:0	USR	RO	UART Status Register [0] UART Busy. [1] Transmit FIFO Not Full. [2] Transmit FIFO Empty. [3] Receive FIFO Not Empty. [4] Receive FIFO Full.	
31:5	Reserved			

TFL

Offset Address: 0x080

Bits	Name	Access	Description	Reset
5:0	TFL	RO	Transmit FIFO Level. This is indicates the number of data entries in the transmit FIFO.	
31:6	Reserved			

RFL

Offset Address: 0x084

Bits	Name	Access	Description	Reset
5:0	RFL	RO	Receive FIFO Level. This is indicates the number of data entries in the receive FIFO.	
31:6	Reserved			

SRR

Offset Address: 0x088

Bits	Name	Access	Description	Reset
2:0	SRR	R/W	Software Reset Register [0] UART Reset. [1] RCVR FIFO Reset. [2] XMIT FIFO Reset.	0x0
31:3	Reserved			

SRTS

Offset Address: 0x08c

Bits	Name	Access	Description	Reset
0	SRTS	R/W	Shadow Request to Send. This is a shadow register for the RTS bit (MCR[1])	0x0
31:1	Reserved			

SBCR

Offset Address: 0x090

Bits	Name	Access	Description	Reset
0	SBCR	R/W	Shadow Break Control Bit. This is a shadow register for the Break bit (LCR[6]).	0x0
31:1	Reserved			

SDMAM

Offset Address: 0x094

Bits	Name	Access	Description	Reset
0	SDMAM	R/W	Shadow DMA Mode. This is a shadow register for the DMA mode bit (FCR[3]).	0x0
31:1	Reserved			

SFE

Offset Address: 0x098

Bits	Name	Access	Description	Reset
0	SFE	R/W	Shadow FIFO Enable. This is a shadow register for the FIFO enable bit (FCR[0]).	0x0
31:1	Reserved			

SRT

Offset Address: 0x09c

Bits	Name	Access	Description	Reset
1:0	SRT	R/W	Shadow RCVR Trigger. This is a shadow register for the RCVR trigger bits (FCR[7:6]).	0x0
31:2	Reserved			

STET

Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
1:0	STET	R/W	Shadow TX Empty Trigger. This is a shadow register for the TX empty trigger bits (FCR[5:4]).	0x0
31:2	Reserved			

HTX

Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
0	HTX	R/W	This register is use to halt transmissions for testing,	0x0
31:1	Reserved			

DMASA

Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
0	DMASA	R/W	This register is use to perform a DMA software acknowledge if a transfer needs to be terminated due to an error condition.	0x0
31:1	Reserved			

12.3 SPI

12.3.1 Overview

The system is equipped with four SPI controller modules, which can be used as a Master for synchronous serial communication with external devices to realize serial/parallel conversion of data.

12.3.2 Characteristics

The characteristics of SPI controller module are as follows:

- Support Motorola SPI (full duplex), TI SSP (full duplex) and NS MicroWire (half duplex) serial peripheral interface protocols.
- Independent receive / transmit FIFO
- Programmable data frame length: 4-16 bits
- The clock frequency of SPI interface is programmable
- Support DMA operation mode
- Support internal loopback test mode
- Working reference clock can be set to 187.5MHz or 100MHz, output SPI_SCK supports a maximum of 46.875MHz.

12.3.3 Function description

12.3.3.1 Tpyical application

The application block diagram of SPI master docking with external slave is shown in Figure 12-9.

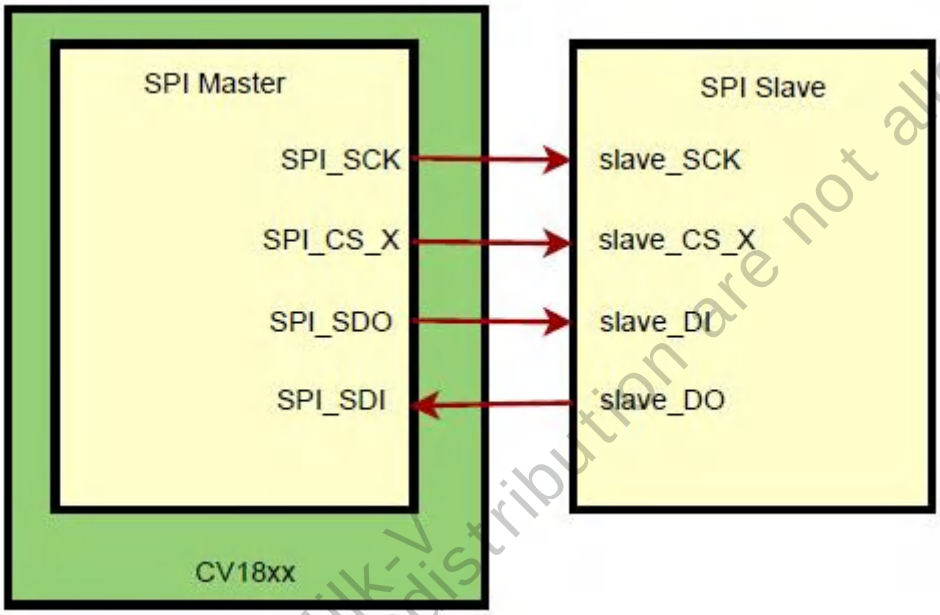


Figure 12-9 SPI application block diagram

12.3.4 Working mode

12.3.4.1 Work mode

SPI's working modes are as follows:

Data transmission in interrupt or query mode

Data transmission in DMA mode

12.3.4.2 Clock

The reference clock of SPI controller module can be set to 187.5MHz or 100MHz. SPI_SCK output supports a maximum of 46.875MHz.

The calculation method is as follows:

SPI_SCK output = SPI reference clock / BAUDR

SPI reference clock: 187.5MHz or 100MHz.

BAUDR register: set an even number between 2 and 65534

Calculation example:

SPI reference clock = 187.5MHz and BAUDR = 4

SPI_SCK output = 187.5MHz/4 = 46.875MHz

12.3.4.3 Interrupt processing

SPI controller module has six interrupts, the first five of which are level interrupts(active high) and can be masked independently.

RXFINTR

Receive FIFO interrupt request. The interrupt is set when RXFTLR+1 or more valid data are in the receive FIFO

RXOINTR

When the receive FIFO is full and new data needs to be written into FIFO, FIFO overflow will be caused and the interrupt is set. At this time, the data is written to the receive shift register instead of FIFO.

RXUINTR

When the receive FIFO is read empty and no new data is written into the receive FIFO, a new read request occurs, which will cause FIFO underflow and the interrupt is set. At this time, all the values read are 0. The interrupt can be cleared by reading register RXUICR.

TXOINTR

When the transmit FIFO is full and new data needs to be written into FIFO, FIFO overflow will be caused and the interrupt is set.

TXEINTR

Send FIFO interrupt request. This interrupt is set when there are TXFTLR or less valid data in the transmit FIFO.

SPI_INTR

The combined interrupt is the "OR" operation result of the above five interrupts.

To mask this interrupt, register IMR must be set to mask the above five interrupts.
If any of the above five independent interrupts is set and enabled, the interrupt is set.

12.3.4.4 Initialization

The initialization steps of SPI controller module are as follows:
Step 1: set "0" in register SPIENR to stop SPI module.
Step 2: configure the register BAUDR and set the divisor of output clock frequency division. The set value must be even number.
Step 3: set register CTRLR0, and configure parameters such as bit width and frame format of transmission data.
Step 4: In DMA operation mode, configure register DMACR to enable DMA function of SPI. When operating in DMA mode, the interrupt related register should be set to prevent the generation of interrupt signal.
Step 5: In interrupt operation mode, set register IMR to generate corresponding interrupt signal
Step 6: set "1" in register SPIENR to enable SPI module

12.3.4.5 Data transmission process of SPI

The process of SPI master docking with external SPI / SSP slave is shown in Figure 12-10.

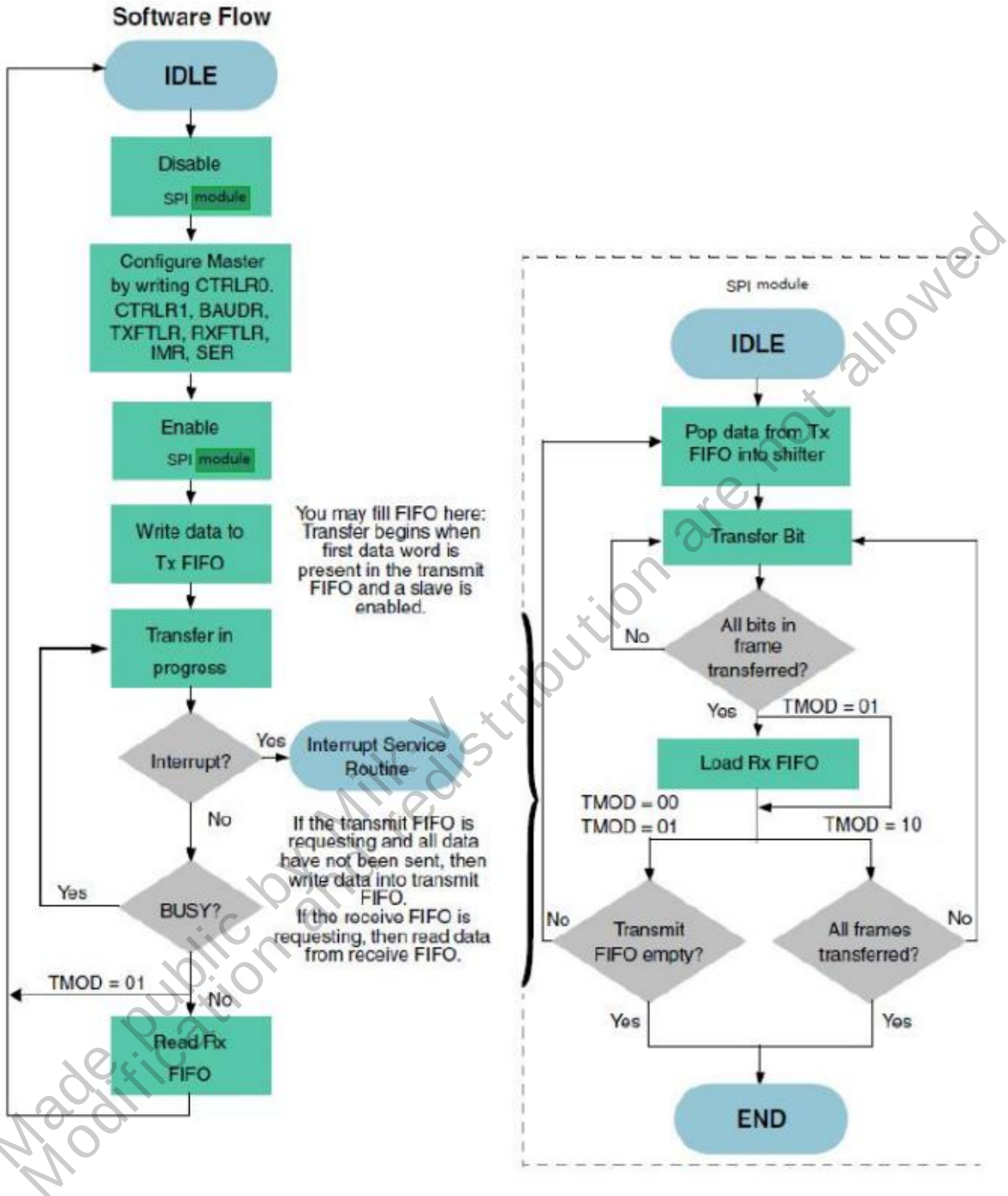


Figure 12-10 data transmission process when docking with external SPI / SSP slave

- The process of SPI master docking with external microwire slave is shown in Figure 12-11.

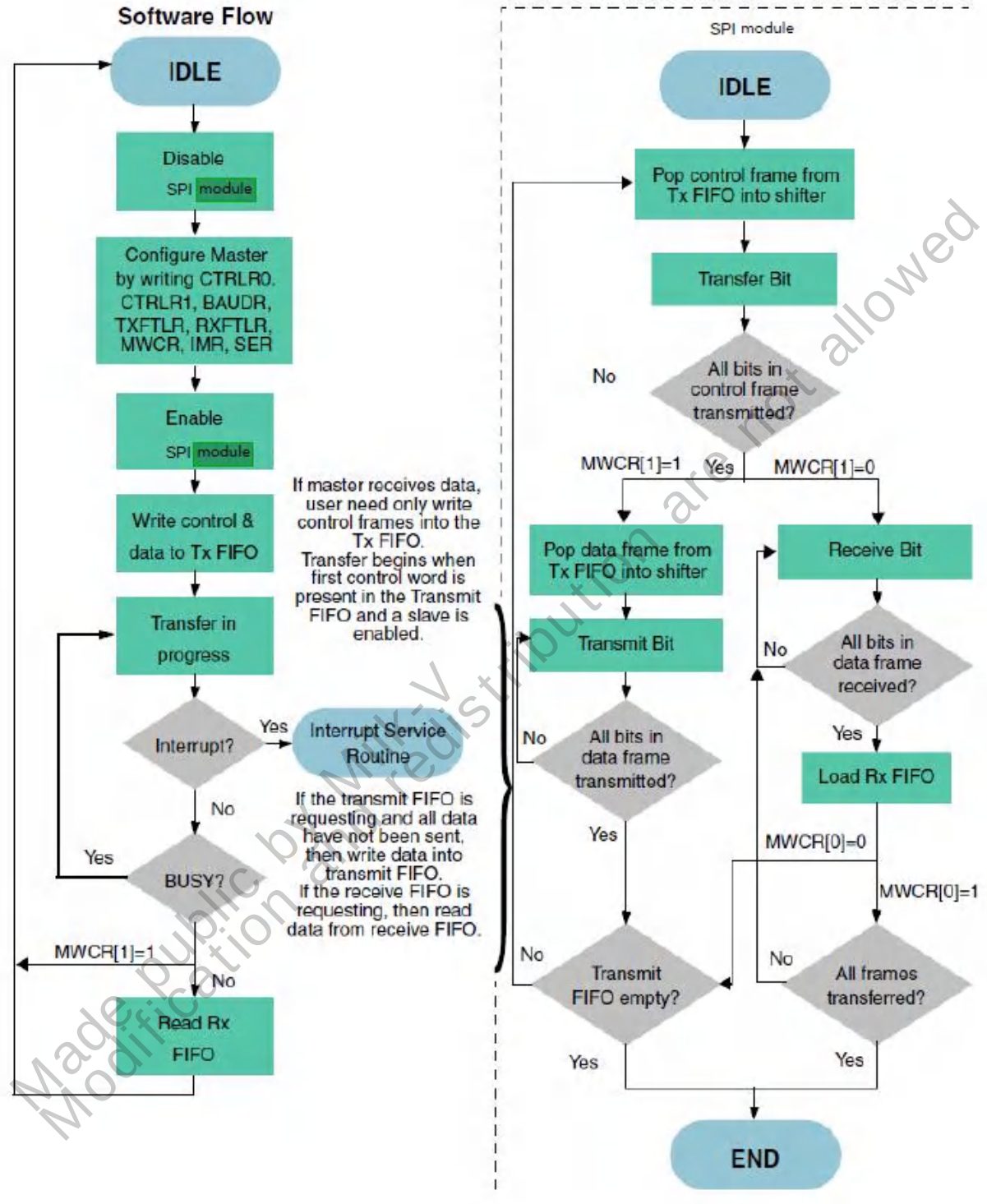


Figure 12-11 data transmission process when connecting to external Microwire slave

12.3.4.6 Data transmission in DMA mode

SPI module uses two DMA channels, one for transmitting and one for receiving. The registers of SPI DMA mode are DMACR, DMATDLR and DMARDLR.

The steps to enable SPI DMA mode are as follows:

Step 1: allocate two DMA channels for SPI.

Step 2: set the register DMACR[1:0] to enable SPI DMA transmission.

Step 3: set "1" in register SPIENR to enable SPI.

Step 4: send data

1. Configure the control registers related to the transmit DMA channel.
2. Start DMA controller to respond to the request of SPI sending FIFO.
3. Judge whether the transmission is completed through DMA Controller Interrupt report, and close DMA function of SPI if the transmission is completed.

Step 5: receive data

1. Configure control registers related to receive DMA channel.
2. Start DMA controller to respond to the request of SPI receiving FIFO.
3. Through DMA Controller Interrupt report, judge whether the data is received completely, if it is finished, close the receiving DMA function of SPI.

Step 6: set "0" to SPIENR register to stop SPI.

12.3.5 Three kinds of serial peripheral bus sequence diagram

12.3.5.1 Motorola SPI interface

The following figures show various data transmission formats of Motorola SPI. SCPH stands for SPI_ SCK phase, SCPOL for SPI_ SCK polarity is set by register CTRLR0[7:6].

(A) SCPH = 0

In this mode, SPI_CS_X is set to high level when it is idle and low level when it is transmitting. SPI_SCK is different through SCPOL setting, SCPOL = 0, set to low level when in idle state, capture data by rising edge of clock when

transmitting, SCPOL = 1, set to high level when in idle state, capture data by falling edge of clock when transmitting.

The single frame transmission format is shown in Figure 12-12.

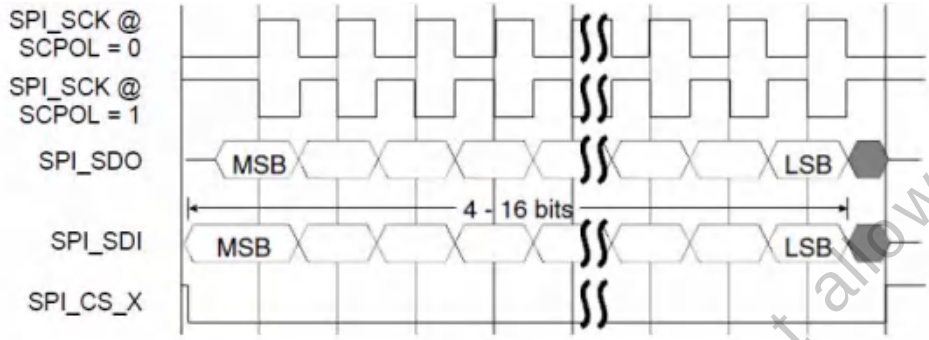


Figure 12-12Motorola SPI single frame transmission format (SCPH = 0)

The format of continuous frame transmission is shown in Figure 12-13.

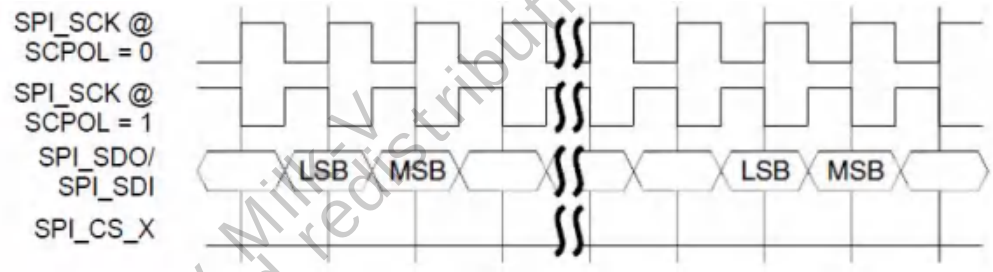


Figure 12-13 Motorola SPI continuous frame transmission format (SCPH = 0)

(B) SCPH = 1

In this mode, SPI_CS_X is set to high level when it is idle and low level when it is transmitting. SPI_SCK is different through scpol setting, SCPOL = 0, set to low level when in idle state, capture data by falling edge of clock when transmitting, SCPOL = 1, set to high level when in idle state, capture data by rising edge of clock when transmitting.

The single frame transmission format is shown in Figure 12-14.

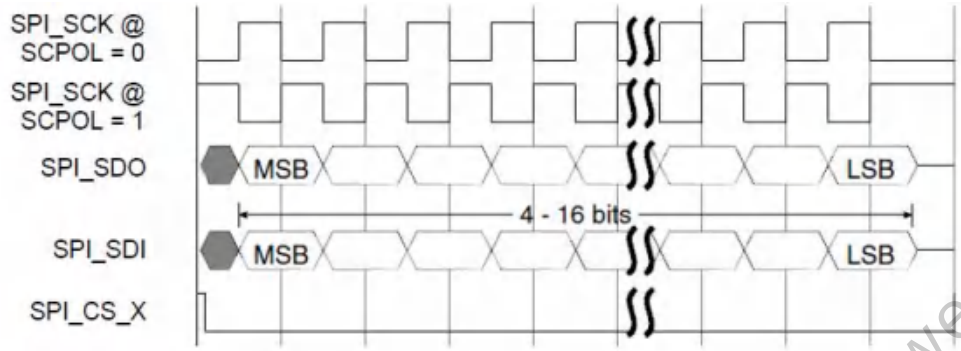


Figure 12-14 Motorola SPI single frame transmission format (SCPH = 1)

The continuous frame transmission format is shown in Figure 12-15.

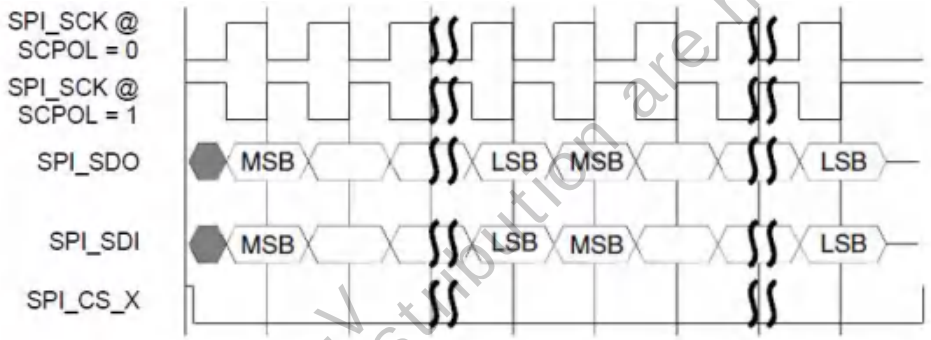


Figure 12-15 Motorola SPI continuous frame transmission format (SCPH = 1)

12.3.5.2 TI Synchronous Serial Interface

In SSP mode, SPI_CS_X is set to high level when in idle state; SPI_CS_X is set to low level when in transmitting state. SPI_ SCK is set to low level when in idle state, and it captures data by falling edge of clock when transmitting.

The following figures show the TI SSP data transmission format.

The single frame transmission format is shown in Figure 12-16.

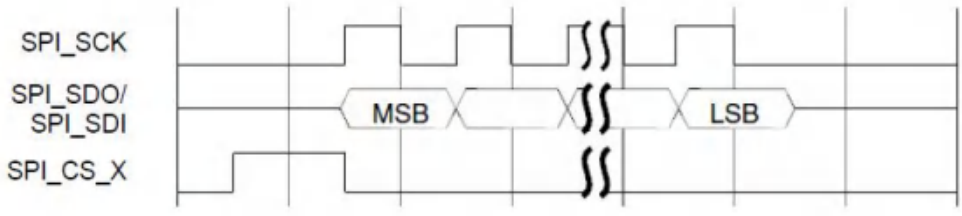


Figure 12- 1 TI SSP Single Frame Transmission Format

The format of continuous frame transmission is shown in Figure 12-17.

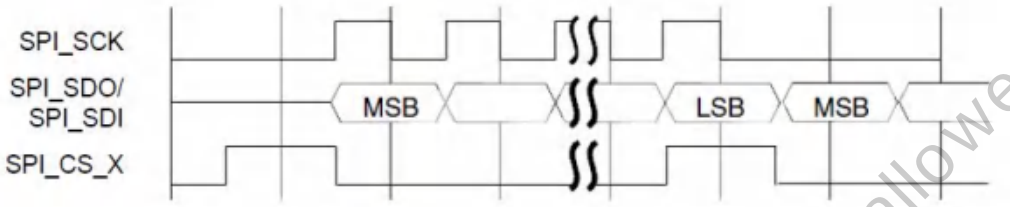


Figure 12- 2 TI SSP Continuous Frame Transmission Format

12.3.5.3 National Semiconductor Microwire Interface

In Microwire mode, SPI_CS_X is set to high level when in idle state; SPI_CS_X is set to low level when in transmitting state. SPI_ SCK is set to low level when in idle state, and it captures data by rising edge of clock when transmitting.
In this mode, the control word must be added before data transmission, and then the external chip responds to the data word required by the Master according to the control word. The length of the control word can be set through the register CTRLR0[15:12], and other related parameters can be set through the register MWCR.

The following figures show the NS Microwire data transmission format.
The single frame transmission format is shown in Figure 12-18.

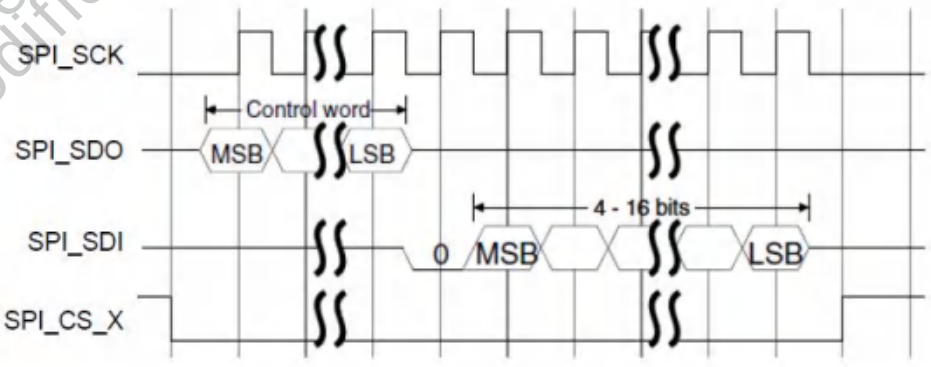


Figure 12- 3 NS Microwire Single Frame Transmission Format

The format of continuous frame transmission is shown in Figure 12-19.

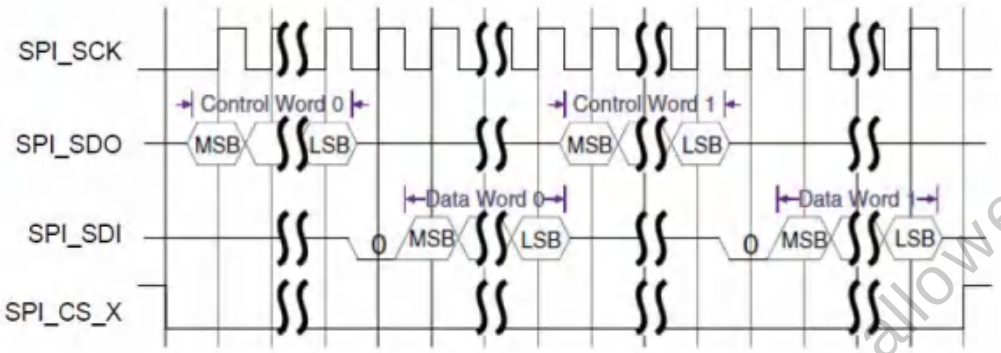


Figure 12- 4 NS Microwire Continuous Frame Transmission Format

12.3.6 Register Overview

The base addresses of four groups of SPI modules are shown in Table 12-2.

Table 12- 2 Four Groups of SPI Module Base Address of the Chip

GPIO Module	Base Address
SPI0	0x04180000
SPI1	0x04190000
SPI2	0x041A0000
SPI3	0x041B0000

Table 12-3 is the offset address and definition of the first group of SPI module (SPI0) registers. SPI0 to SPI3 have the same register definition.

Table 12- 3 SPI Register Overview

Name	Address Offset	Description
CTRLR0	0x000	Control Register 0
CTRLR1	0x004	Control Register 1
SPIENR	0x008	SPI Enable Register
MWCR	0x00c	Microwire Control Register
SER	0x010	Slave Enable Register
BAUDR	0x014	Baud Rate Select
TXFTLR	0x018	Transmit FIFO Threshold Level
RXFTLR	0x01c	Receive FIFO Threshold Level

Name	Address Offset	Description
TXFLR	0x020	Transmit FIFO Level Register
RXFLR	0x024	Receive FIFO Level Register
SR	0x028	Status Register
IMR	0x02c	Interrupt Mask Register
ISR	0x030	Interrupt Status Register
RISR	0x034	Raw Interrupt Status Register
TXOICR	0x038	Transmit FIFO Overflow Interrupt Clear Register
RXOICR	0x03c	Receive FIFO Overflow Interrupt Clear Register
RXUICR	0x040	Receive FIFO Underflow Interrupt Clear Register
MSTICR	0x044	Multi-Master Interrupt Clear Register
ICR	0x048	Interrupt Clear Register
DMACR	0x04c	DMA Control Register
DMATDLR	0x050	DMA Transmit Data Level
DMARDLR	0x054	DMA Receive Data Level
DR (36 组)	0x060	Data Register
RX_SAMPLE_DLY	0x0f0	Rx Sample Delay Register

12.3.7 Register Description

CTRLR0

Offset Address: 0x000

Bits	Name	Access	Description	Reset
15:0	CTRLR0	R/W	[15:12] Control Frame Size. Selects the length of the control word for the Microwire frame format. 0000 1-bit control word 0001 2-bit control word 0010 3-bit control word 0011 4-bit control word 0100 5-bit control word 0101 6-bit control word 0110 7-bit control word 0111 8-bit control word 1000 9-bit control word 1001 10-bit control word 1010 11-bit control word 1011 12-bit control word 1100 13-bit control word 1101 14-bit control word 1110 15-bit control word 1111 16-bit control word [11] Shift Register Loop. Used for testing purposes only. When internally active, connects the transmit shift register output to the receive shift register input. Can be used in both serialslave and serial-master modes.	0x7

Bits	Name	Access	Description	Reset
			0 – Normal Mode Operation 1 – Test Mode Operation [10] only for slave mode. [9:8] Transfer Mode. Selects the mode of transfer for serial communication. This field does not affect the transfer duplicity. Only indicates whether the receive or transmit data are valid. In transmit-only mode, data received from the external device is not valid and is not stored in the receive FIFO memory; it is overwritten on the next transfer. In receive-only mode, transmitted data are not valid. After the first write to the transmit FIFO, the same word is retransmitted for the duration of the transfer. In transmit-and-receive mode, both transmit and receive data are valid. The transfer continues until the transmit FIFO is empty. Data received from the external device are stored into the receive FIFO memory, where it can be accessed by the host processor. In eeprom-read mode, receive data is not valid while control data is being transmitted. When all control data is sent to the EEPROM, receive data becomes valid and transmit data becomes invalid. All data in the transmit FIFO is considered control data in this mode. 00 –- Transmit & Receive 01 –- Transmit Only 10 –- Receive Only 11 –- EEPROM Read [7] Serial Clock Polarity. Valid when the frame format (FRF) is set to Motorola SPI. Used to select the polarity of the inactive serial clock, which is held inactive when the SPI master is not actively transferring data on the serial bus. 0 – Inactive state of serial clock is low 1 – Inactive state of serial clock is high [6] Serial Clock Phase. Valid when the frame format (FRF) is set to Motorola SPI. The serial clock phase selects the relationship of the serial clock with the slave select signal. When SCPH = 0, data are captured on the first	

Bits	Name	Access	Description	Reset
			edge of the serial clock. When SCPH = 1, the serial clock starts toggling one cycle after the slave select line is activated, and data are captured on the second edge of the serial clock. 0: Serial clock toggles in middle of first data bit 1: Serial clock toggles at start of first data bit [5:4] Frame Format. Selects which serial protocol transfers the data. 00 –- Motorola SPI 01 –- Texas Instruments SSP 10 –- National Semiconductors Microwire 11 –- Reserved [3:0] Data Frame Size. Selects the data frame length. When the data frame size is programmed to be less than 16 bits, the receive data are automatically right-justified by the receive logic, with the upper bits of the receive FIFO zero-padded. You must right-justify transmit data before writing into the transmit FIFO. The transmit logic ignores the upper unused bits when transmitting the data. 0000 Reserved – undefined operation 0001 Reserved – undefined operation 0010 Reserved – undefined operation 0011 4-bit serial data transfer 0100 5-bit serial data transfer 0101 6-bit serial data transfer 0110 7-bit serial data transfer 0111 8-bit serial data transfer 1000 9-bit serial data transfer 1001 10-bit serial data transfer 1010 11-bit serial data transfer 1011 12-bit serial data transfer 1100 13-bit serial data transfer 1101 14-bit serial data transfer 1110 15-bit serial data transfer 1111 16-bit serial data transfer	
31:16	Reserved			

CTRLR1

Offset Address: 0x004

Bits	Name	Access	Description	Reset
15:0	CTRLR1	R/W	Number of Data Frames. When TMOD = 10 or TMOD = 11, this register field sets the number of data frames to be continuously received by	0x0

Bits	Name	Access	Description	Reset
			the SPI. The SPI continues to receive serial data until the number of data frames received is equal to this register value plus 1, which enables you to receive up to 64 KB of data in a continuous transfer.	
31:16	Reserved			

SPIENR

Offset Address: 0x008

Bits	Name	Access	Description	Reset
0	SPIENR	R/W	SPI Enable. Enables and disables all SPI operations. When disabled, all serial transfers are halted immediately. Transmit and receive FIFO buffers are cleared when the device is disabled. It is impossible to program some of the SPI control registers when enabled. When disabled, the spi_sleep output is set (after delay) to inform the system that it is safe to remove the spi_clk, thus saving power consumption in the system.	0x0
31:1	Reserved			

MWCR

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
2:0	MWCR	R/W	[2] Microwire Handshaking. Relevant only when the SPI is configured as a serial-master device. When configured as a serial slave, this bit field has no functionality. Used to enable and disable the “busy/ready” handshaking interface for the Microwire protocol. When enabled, the SPI checks for a ready status from the target slave, after the transfer of the last data/control bit, before clearing the BUSY status in the SR register. 0: handshaking interface is disabled 1: handshaking interface is enabled [1] Microwire Control. Defines the direction of the data word when the Microwire serial protocol is used. When this bit is set to 0, the data word is received by the SPI from the external serial device. When this bit is set to 1, the data word is transmitted from the SPI to the external serial device. [0] Microwire Transfer Mode.	0x0

Bits	Name	Access	Description	Reset
			Defines whether the Microwire transfer is sequential or non-sequential. When sequential mode is used, only one control word is needed to transmit or receive a block of data words. When non-sequential mode is used, there must be a control word for each data word that is transmitted or received. 0 – non-sequential transfer 1 – sequential transfer	
31:3	Reserved			

SER

Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	SER	R/W	Slave Select Enable Flag. This register corresponds to a slave select line (ss_x_n]) from the SPI master. When this register is set (1), the slave select line from the master is activated when a serial transfer begins. It should be noted that setting or clearing this register have no effect on the corresponding slave select outputs until a transfer is started. Before beginning a transfer, you should enable this register that corresponds to the slave device with which the master wants to communicate. 1: Selected 0: Not Selected	0x0
31:1	Reserved			

BAUDR

Offset Address: 0x014

Bits	Name	Access	Description	Reset
15:0	BAUDR	R/W	SPI Clock Divider(SCKDV). The LSB for this field is always set to 0 and is unaffected by a write operation, which ensures an even value is held in this register. If the value is 0, the serial output clock (sclk_out) is disabled. The frequency of the sclk_out is derived from the following equation: Fsclk_out/SCKDV = Fssi_clk where SCKDV is any even value between 2 and 65534. For example: for Fssi_clk = 3.6864MHz and SCKDV =2 Fsclk_out = 3.6864/2 = 1.8432MHz	0x0
31:16	Reserved			

TXFTLR

Offset Address: 0x018

Bits	Name	Access	Description	Reset
2:0	TXFTLR	R/W	Transmit FIFO Threshold. Controls the level of entries (or below) at which the transmit FIFO controller triggers an interrupt. The FIFO depth is 8; If you attempt to set this value greater than or equal to the depth of the FIFO, this field is not written and retains its current value. When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered.	0x0
31:3	Reserved			

RXFTLR

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
2:0	RXFTLR	R/W	Receive FIFO Threshold. Controls the level of entries (or above) at which the receive FIFO controller triggers an interrupt. The FIFO depth is 8. If you attempt to set this value greater than the depth of the FIFO, this field is not written and retains its current value. When the number of receive FIFO entries is greater than or equal to this value + 1, the receive FIFO full interrupt is triggered.	0x0
31:3	Reserved			

TXFLR

Offset Address: 0x020

Bits	Name	Access	Description	Reset
3:0	TXFLR	RO	Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO.	
31:4	Reserved			

RXFLR

Offset Address: 0x024

Bits	Name	Access	Description	Reset
3:0	RXFLR	RO	Receive FIFO Level. Contains the number of valid data entries in the receive FIFO.	
31:4	Reserved			

SR

Offset Address: 0x028

Bits	Name	Access	Description	Reset
6:0	SR	RO	[6] Data Collision Error. This bit is set if the SPI master is actively transmitting when another master	

Bits	Name	Access	Description	Reset
			selects this device as a slave. This informs the processor that the last data transfer was halted before completion. This bit is cleared when read. 0 – No error 1 – Transmit data collision error [5] Transmission Error. Set if the transmit FIFO is empty when a transfer is started. Data from the previous transmission is resent on the txd line. This bit is cleared when read. 0 – No error 1 – Transmission error [4] Receive FIFO Full. When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared. 0 – Receive FIFO is not full 1 – Receive FIFO is full [3] Receive FIFO Not Empty. Set when the receive FIFO contains one or more entries and is cleared when the receive FIFO is empty. This bit can be polled by software to completely empty the receive FIFO. 0 – Receive FIFO is empty 1 – Receive FIFO is not empty [2] Transmit FIFO Empty. When the transmit FIFO is completely empty, this bit is set. When the transmit FIFO contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt. 0 – Transmit FIFO is not empty 1 – Transmit FIFO is empty [1] Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. 0 – Transmit FIFO is full 1 – Transmit FIFO is not full [0] SPI Busy Flag. When set, indicates that a serial transfer is in progress; when cleared indicates that the SPI is idle or disabled. 0 – SPI is idle or disabled 1 – SPI is actively transferring data	
31:7	Reserved			

IMR

Offset Address: 0x02c

Bits	Name	Access	Description	Reset
5:0	IMR	R/W	[5] Multi-Master Contention Interrupt Mask. 0 – spi_mst_intr interrupt is masked 1 – spi_mst_intr interrupt is not masked [4] Receive FIFO Full Interrupt Mask 0 – spi_rxf_intr interrupt is masked 1 – spi_rxf_intr interrupt is not masked [3] Receive FIFO Overflow Interrupt Mask 0 – spi_rxo_intr interrupt is masked 1 – spi_rxo_intr interrupt is not masked [2] Receive FIFO Underflow Interrupt Mask 0 – spi_rxu_intr interrupt is masked 1 – spi_rxu_intr interrupt is not masked [1] Transmit FIFO Overflow Interrupt Mask 0 – spi_txo_intr interrupt is masked 1 – spi_txo_intr interrupt is not masked [0] Transmit FIFO Empty Interrupt Mask 0 – spi_txe_intr interrupt is masked 1 – spi_txe_intr interrupt is not masked	0x3F
31:6	Reserved			

ISR

Offset Address: 0x030

Bits	Name	Access	Description	Reset
5:0	ISR	RO	[5] Multi-Master Contention Interrupt Status. 0 = spi_mst_intr interrupt not active after masking 1 = spi_mst_intr interrupt is active after masking [4] Receive FIFO Full Interrupt Status 0 = spi_rxf_intr interrupt is not active after masking 1 = spi_rxf_intr interrupt is full after masking [3] Receive FIFO Overflow Interrupt Status 0 = spi_rxo_intr interrupt is not active after masking 1 = spi_rxo_intr interrupt is active after masking	

Bits	Name	Access	Description	Reset
			[2] Receive FIFO Underflow Interrupt Status 0 = spi_rxu_intr interrupt is not active after masking 1 = spi_rxu_intr interrupt is active after masking [1] Transmit FIFO Overflow Interrupt Status 0 = spi_txo_intr interrupt is not active after masking 1 = spi_txo_intr interrupt is active after masking [0] Transmit FIFO Empty Interrupt Status 0 = spi_txe_intr interrupt is not active after masking 1 = spi_txe_intr interrupt is active after masking	
31:6	Reserved			

RISR

Offset Address: 0x034

Bits	Name	Access	Description	Reset
5:0	RISR	RO	[5] Multi-Master Contention Raw Interrupt Status. 0 = spi_mst_intr interrupt is not active prior to masking 1 = spi_mst_intr interrupt is active prior masking [4] Receive FIFO Full Raw Interrupt Status 0 = spi_rxf_intr interrupt is not active prior to masking 1 = spi_rxf_intr interrupt is active prior to masking [3] Receive FIFO Overflow Raw Interrupt Status 0 = spi_rxo_intr interrupt is not active prior to masking 1 = spi_rxo_intr interrupt is active prior masking [2] Receive FIFO Underflow Raw Interrupt Status 0 = spi_rxu_intr interrupt is not active prior to masking 1 = spi_rxu_intr interrupt is active prior to masking [1] Transmit FIFO Overflow Raw	

Bits	Name	Access	Description	Reset
			Interrupt Status 0 = spi_txo_intr interrupt is not active prior to masking 1 = spi_txo_intr interrupt is active prior masking [0] Transmit FIFO Empty Raw Interrupt Status 0 = spi_txe_intr interrupt is not active prior to masking 1 = spi_txe_intr interrupt is active prior masking	
31:6	Reserved			

TXOICR

Offset Address: 0x038

Bits	Name	Access	Description	Reset
0	TXOICR	RO	Clear Transmit FIFO Overflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the spi_txo_intr interrupt; writing has no effect.	
31:1	Reserved			

RXOICR

Offset Address: 0x03c

Bits	Name	Access	Description	Reset
0	RXOICR	RO	Clear Receive FIFO Overflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the spi_rxo_intr interrupt; writing has no effect.	
31:1	Reserved			

RXUICR

Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	RXUICR	RO	Clear Receive FIFO Underflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the spi_rxu_intr interrupt; writing has no effect.	
31:1	Reserved			

MSTICR

Offset Address: 0x044

Bits	Name	Access	Description	Reset
0	MSTICR	RO	Clear Multi-Master Contention Interrupt.	

Bits	Name	Access	Description	Reset
			This register reflects the status of the interrupt. A read from this register clears the spi_mst_intr interrupt; writing has no effect.	
31:1	Reserved			

ICR

Offset Address: 0x048

Bits	Name	Access	Description	Reset
0	ICR	RO	Clear Interrupts. This register is set if any of the interrupts below are active. A read clears the spi_txo_intr, spi_rxu_intr, spi_rxo_intr, and the spi_mst_intr interrupts. Writing to this register has no effect.	
31:1	Reserved			

DMACR

Offset Address: 0x04c

Bits	Name	Access	Description	Reset
1:0	DMACR	R/W	[1] Transmit DMA Enable. This bit enables/disables the transmit FIFO DMA channel. 0 = Transmit DMA disabled 1 = Transmit DMA enabled [0] Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel 0 = Receive DMA disabled 1 = Receive DMA enabled	0x0
31:2	Reserved			

DMATDLR

Offset Address: 0x050

Bits	Name	Access	Description	Reset
2:0	DMATDLR	R/W	Transmit Data Level. This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and TDMAE = 1.	0x0
31:3	Reserved			

DMARDLR

Offset Address: 0x054

Bits	Name	Access	Description	Reset
2:0	DMARDLR	R/W	Receive Data Level. This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1, and RDMAE=1.	0x0
31:3	Reserved			

DR

Offset Address: 0x060

Bits	Name	Access	Description	Reset
15:0	DR	R/W	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. Read = Receive FIFO buffer Write = Transmit FIFO buffer Note : The DR register in the SPI occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus.	0x0
31:16	Reserved			

RX_SAMPLE_DLY

Offset Address: 0x0f0

Bits	Name	Access	Description	Reset
7:0	RX_SAMPLE_DLY	R/W	Receive Data (rxd) Sample Delay. This register is used to delay the sample of the rxd input signal. Each value represents a single ssi_clk delay on the sample of the rxd signal. NOTE: If this register is programmed with a value that exceeds the depth of the internal shift registers (DEPTH = 8), a zero (0) delay will be applied to the rxd sample.	0x0
31:8	Reserved			

12.4 SD/SDIO Controller

12.4.1 Function Description

12.4.1.1 Functional Block Diagram

SD / SDIO controller (SD controller for short) is used to handle the operation of data reading and writing of SD card, as well as the external devices (such as Bluetooth, WiFi, etc.) supported by SDIO protocol. The chip provides two sets of SD controllers, including:

SDIO0 supports devices compliant to the Secure Digital Memory (SD 3.0) protocol.
SDIO1 supports devices complian to the Secure Digital I/O(SDIO 3.0) protocol.

The function signals and pins corresponding to the SD controllers in the chip are shown in the table below.

Table 12- 4 Corresponding Function Signal and Pin of SD Controller

SDMMC Controller	Function Signal	Pin Name
SDIO0	SD_CLK	SD0_CLK
	SD_CMD	SD0_CMD
	SD_DATA0	SD0_D0
	SD_DATA1	SD0_D1
	SD_DATA2	SD0_D2
	SD_DATA3	SD0_D3
	SD_CARD_DETECT	SD0_CD
	SD_POWER_EN	SD0_PWR_EN
SDIO1	SDIO_CLK	SD1_CLK
	SDIO_CMD	SD1_CMD
	SDIO_DATA0	SD1_D0
	SDIO_DATA1	SD1_D1
	SDIO_DATA2	SD1_D2
	SDIO_DATA3	SD1_D3

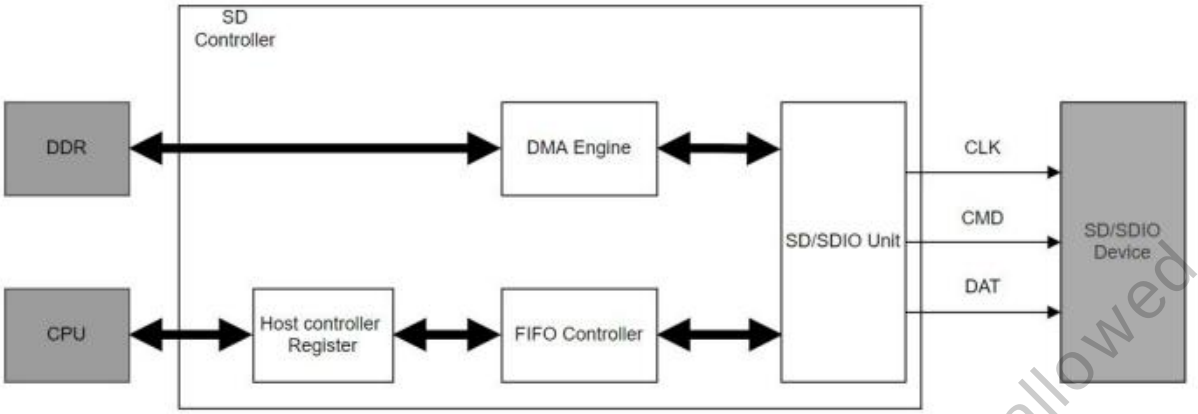


Figure 12- 20 SD Controller Function Block Diagram

Functions of SDMMC:

1. Support SD card, SDIO device.
2. Transfer data between SD / SDIO and system memory through internal DMA controller.
3. Support CRC generation and check of command and data.
4. The required frequency between different modes can be generated through the internal frequency divider.
5. Provide a mechanism to turn off the internal clock and the clock on the interface to save power.
6. Provide 1-bit and 4-bit data transmission interface to communicate with the device.
7. Support block_size read and write operations with size equal to 1-2048byte.
8. Support SDIO protocol, including interrupt interval, suspend, resume and read wait.
9. Support AXI/AHB interface and access system memory through internal DMA.
10. Support AHB interface and access internal registers through CPU.

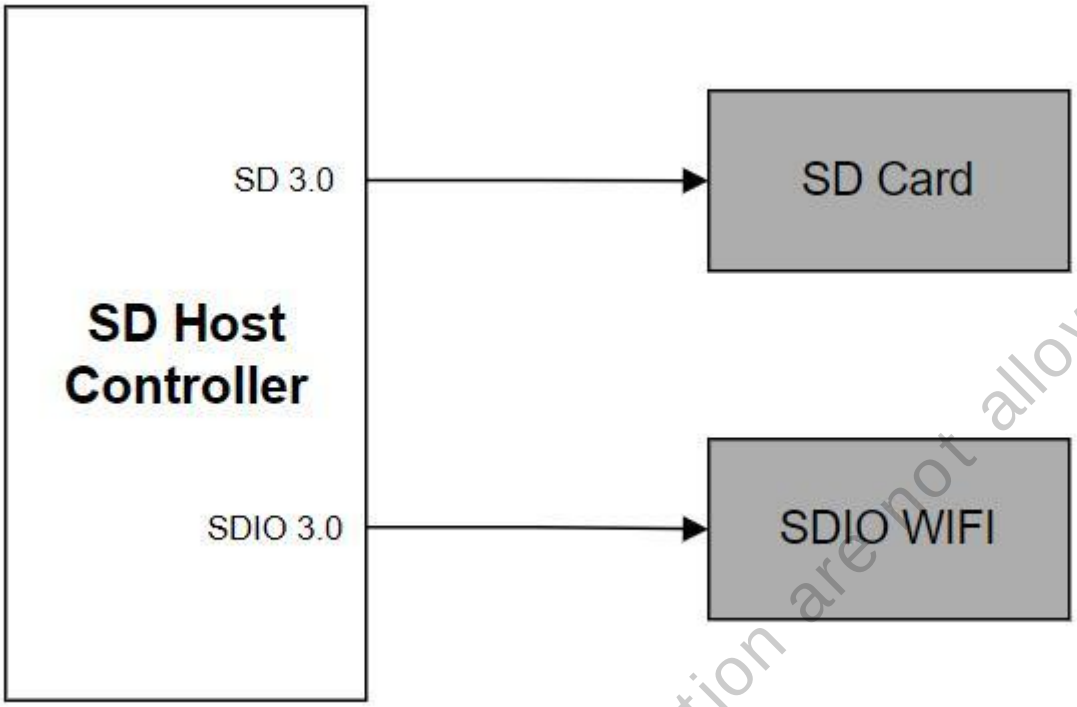


Figure 12- 21 Typical Application

12.4.1.2 Command and Response

The bus packet of SD consists of three parts: commmand, response and data.

The commmand and response packets are transmitted through the CMD signal line.

Command Packet

The command packet is sent to the device by the host to indicate the start of an operation. The packet format consists of 48 bits including start bit, transmit bit, command index, command argument, CRC verification code and end bit. It is shown in Figure 12-22.

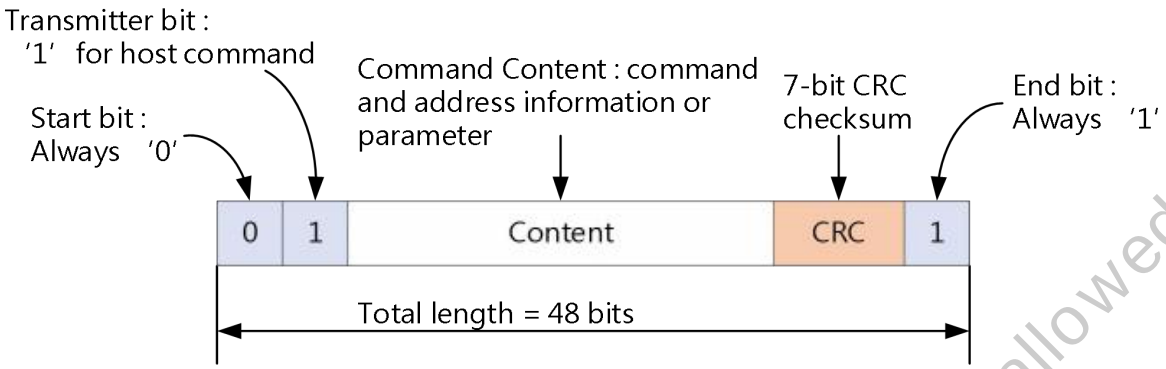


Figure 12- 22 SD/SDIO Command Format

Response Packet

After receiving the command, the device will return the response according to different command types, which is used to show the status or parameters of the device. Its length is 48 bits or 136 bits. It is shown in Figure 12-23.

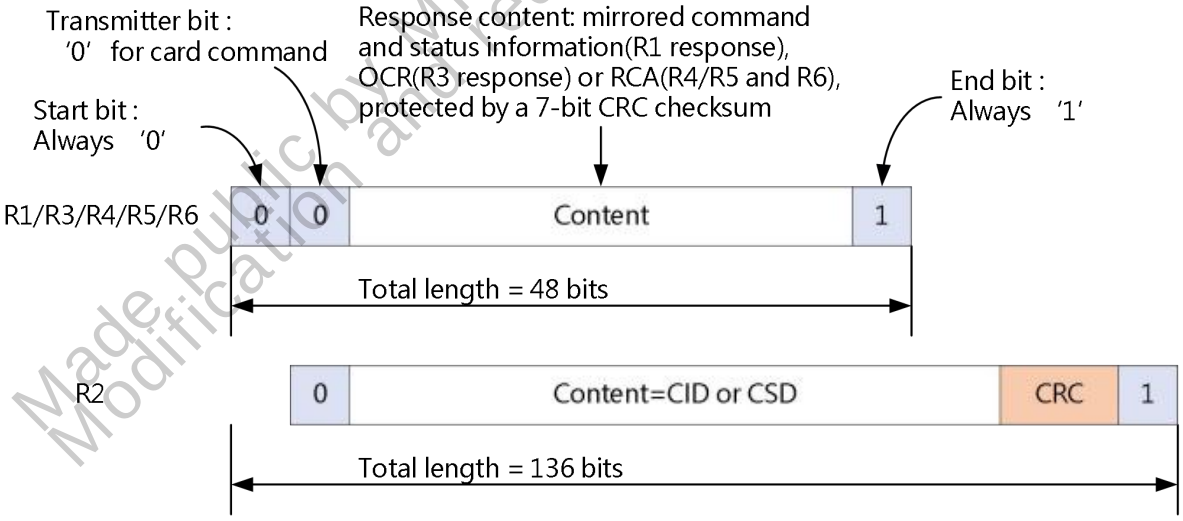


Figure 12- 23 SD/SDIO Response Format

Data Packet

Data packets are used to exchange data between the host and the device. According to different requirements, 1-bit (DATA0), 4-bit (DATA 0- DATA 3) or 7-bit (DATA 0- DATA 7) can be selected. In each clock cycle, each data signal line can choose to transmit 1-bit (single data rate) or 2-bit (dual data rate). The packet formats are shown in Figure 12-24 to 12-26.

Figure 12- 24 SD/SDIO 1-bit Data Packet Format

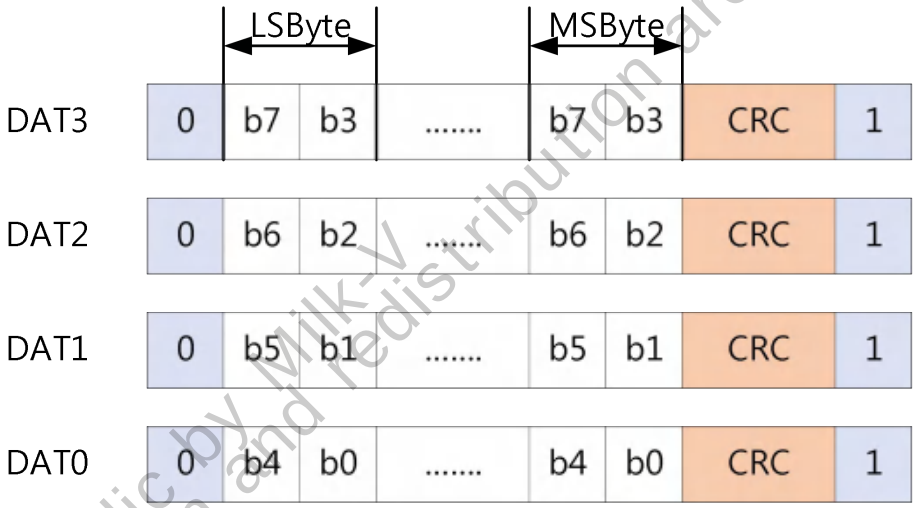


Figure 12- 25 SD/SDIO 4-bit Data Packet Format

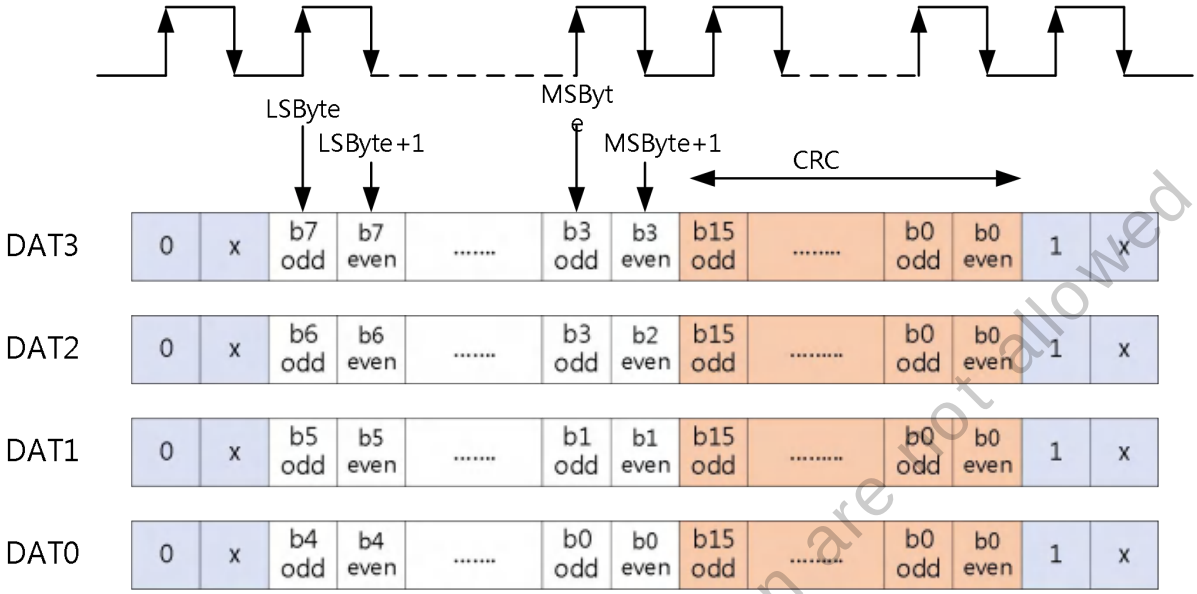


Figure 12- 26 SD/SDIO 4-bit Data Packet Format

According to whether there is data transmission, commands can be further divided into the following two types.

Non data transmission command: through the signal line CMD to complete the command transmission and receiving response.

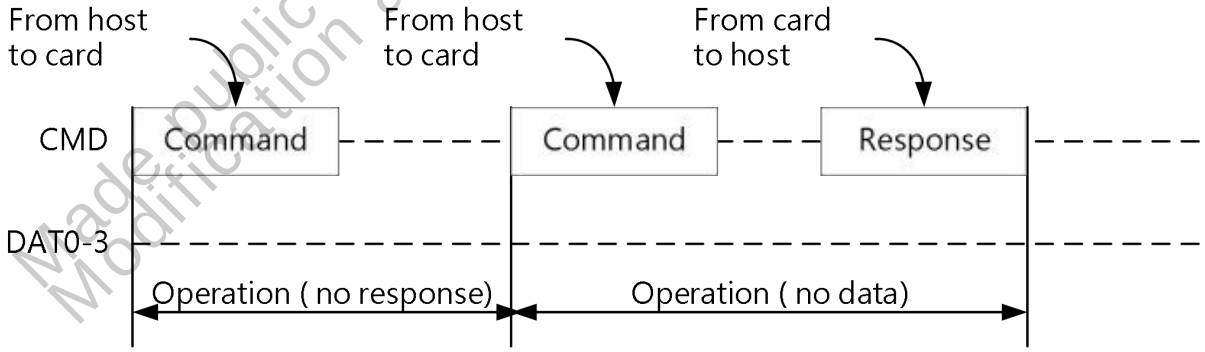


Figure 12- 27 Non data transmission command: Complete Transmission and Receiving Responses through CMD

Data transmission command: in addition to the transmission on the signal line CMD, there is also data transmission on the data lines DAT0~DAT3.

12.4.1.3 Data Transmission

The data transmission between the host and the device is mainly based on the block. In addition to the data, CRC check bits are also included to verify the correctness of the data. Common methods of data reading and writing include single-block and multi-block. Compared with single block data transmission, multi block data transmission has higher efficiency. Among them, the block size of SD card is 512byte. SDIO is special. It can support 1-2048byte block size. Users can define the block size value according to different devices.

(1) Single block and multi block read operations are shown in Figure 12-30. Single block transmission consists of command, response, data and CRC. Multi block transmission terminates transmission by STOP CMD.

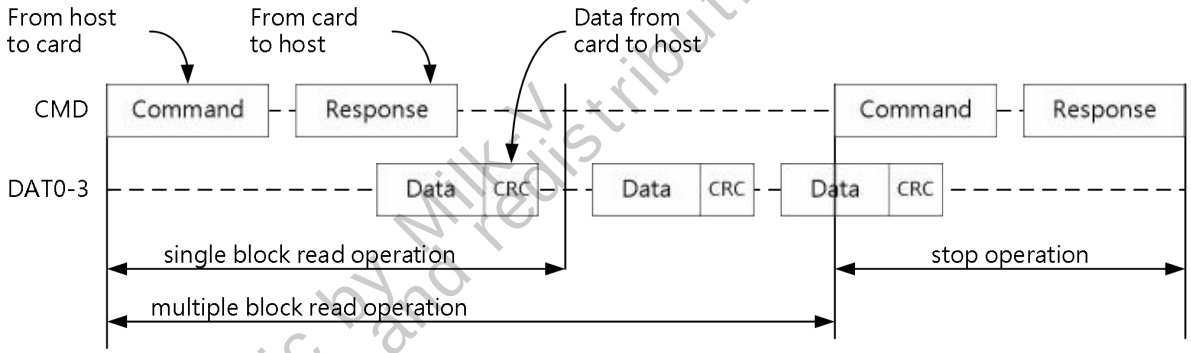


Figure 12- 5 Single Block and Multi Block Read Operation

(2) Single block and multi block write operations are shown in Figure 12-31. In the transmission process, a BUSY signal will be sent through the DAT0 signal line to inform the host that the write device is in progress.

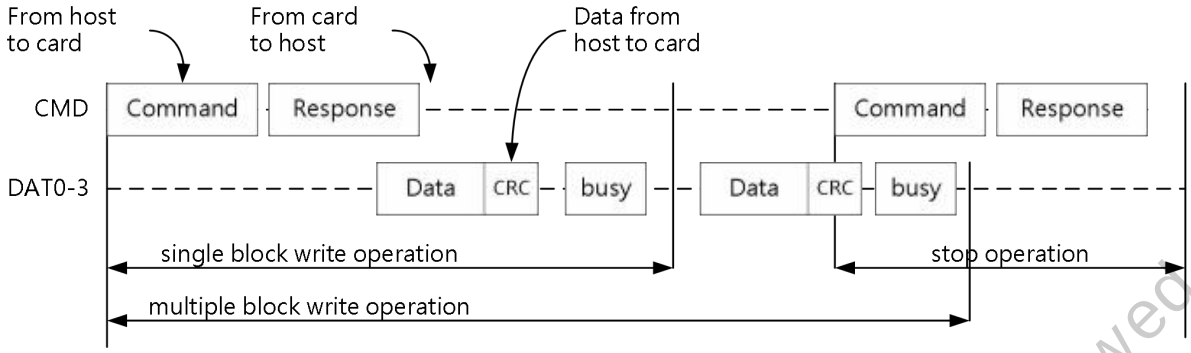


Figure 12- 29 Single Block and Multi Block Write Operation

12.4.1.4 Speed Mode and Voltage Switching supported by SD3.0

Voltage Switching Procedure (1.8V → 3.3V)

Step 1: Set PWRSW to 3.0V mode
=> sd_pwrsw_ctrl (0x030001F4) = 0x00000009

(reg_pwrsw_auto=1, reg_pwrsw_disc=0, reg_pwrsw_vsel=0(3.0v), reg_en_pwrsw=1)

Step 2: Wait 1ms to complete the voltage switching

Voltage Switching Procedure (3.3V → 1.8V)

Step 1: Set PWRSW to 1.8V
=> sd_pwrsw_ctrl (0x030001F4) = 0x0000000B
(reg_pwrsw_auto=1, reg_pwrsw_disc=0, reg_pwrsw_vsel=1(1.8v), reg_en_pwrsw=1)

Step 2: Wait 1ms to complete the voltage switching

Supported speed mode and voltage

The speed mode and voltage supported by SD3.0 are shown in the table below.

Table 12- 1 SD3.0 Supported Speed and Voltage

Supported Mode	Supported Speed	Voltage
DS (default speed)	25Mhz	1.8V/3.3V
HS (high speed)	50Mhz	1.8V/3.3V

Supported Mode	Supported Speed	Voltage
SDR12	25Mhz	1.8V
SDR25	50Mhz	1.8V
DDR50	50Mhz	1.8V
SDR50	100Mhz	1.8V
SDR104	187.5Mhz	1.8V

12.4.2 Application Explanation

12.4.2.1 Clock off Control

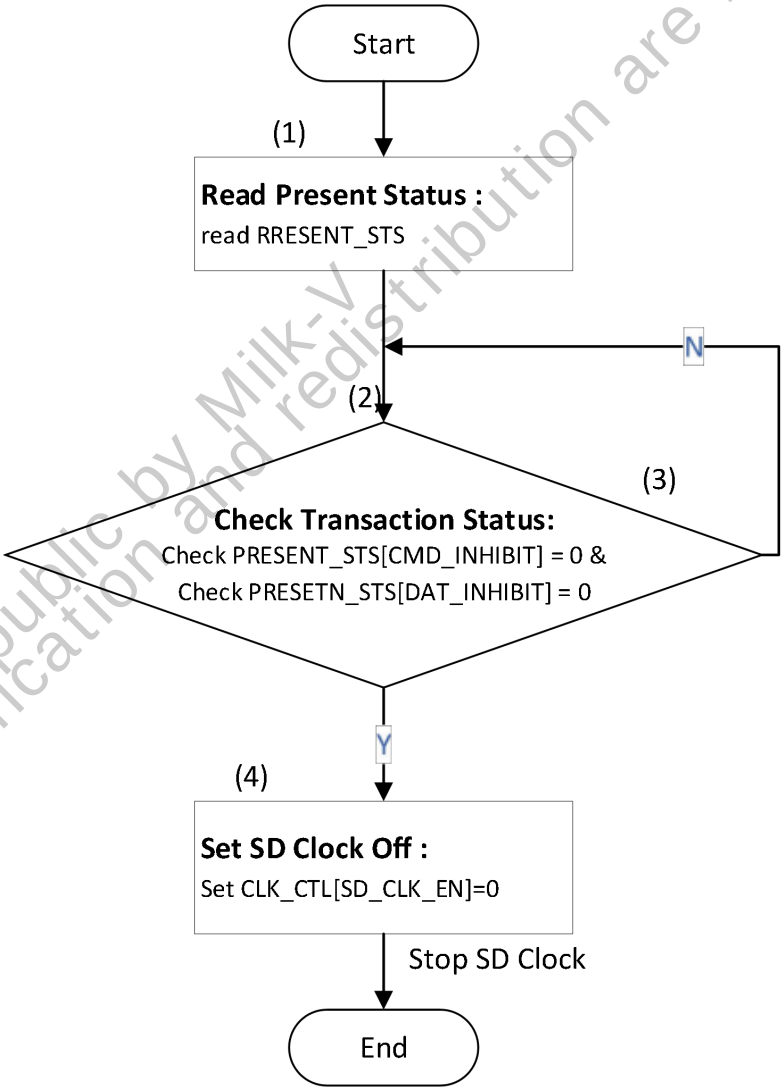


Figure 12- 30 Clock off Control Procedure

As shown in Figure 12-30, is a clock off control program. The host must ensure that there is no transmission on the bus before the clock can be turned off.

(1) Read register PRESENT_STS
(2) Check bit CMD_ INHIBIT and DAT_INHIBIT are both 0
(3) If any bit is not 0, it means that the transmission is still in progress and needs to be delayed.
(4) If all are 0, it can be set_ CTL[SD_ CLK_ EN] = 0 to turn off the clock

12.4.2.2 Soft Reset

When the controller operation is abnormal, reset the configuration register (base address = 0x0300_ 3000) for soft reset. The register addresses used are as follows.

6. SDIO0 : SOFT_RSTN_0[reg_soft_reset_x_sd0] (address offset : 0x000, Bit16)
7. SDIO1 : SOFT_RSTN_0[reg_soft_reset_x_sd1] (address offset : 0x000, Bit17)

12.4.2.3 Interface Clock Configuration

Figure 12-31 shows the flow chart of interface clock configuration. SD controller provides a frequency divider, which allows users to adjust the required clock frequency according to different protocols and speed modes. The relationship is as follows.

$$F_{SD\_CLK\_OUT} = F_{INT\_CARD\_CLK} / (2 \times clk\_divisor)$$

When SD changes the frequency, in addition to ensuring that no commands and data are still in transmission, it must also be set according to the steps of the interface clock configuration flow chart to avoid the clock glitch output to the SD device.

(1) Turn off the interface clock.
(2) Calculate the frequency division factor.
(3) Set the frequency division factor. Fill the parameters calculated in (2) into CLK_ CTL[FREQ_ SEL] and start to turn on the internal clock switch (CLK_CTL[INT_CLK_EN]=1)
(4) Check CLK_CTL[INT_CLK_STABLE] to confirm whether the frequency switching is completed.
(5) If not completed (CLK_CTL[INT_CLK_STABLE]=0), delay to wait.

(6) If the clock frequency switching is completed, turn on the interface clock.

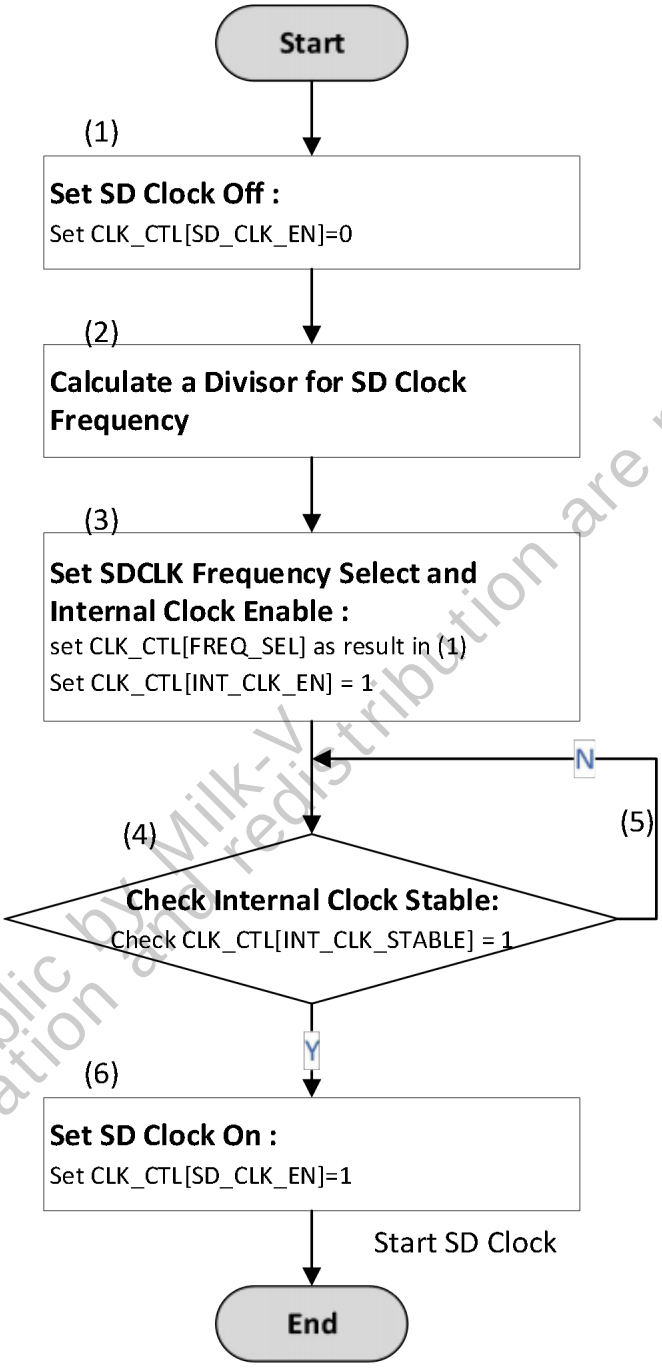


Figure 12- 6 Clock Configuration Flow Chart

12.4.2.4 Non Data Transfer Command

Command transfer sequence

The command transfer procedure is shown in Figure 12-32.

(1) The register bit PRESENT[CMD_INHIBIT] must be 0 to confirm whether the CMD line is still in use.

(2) If the CMD line is idle, further confirm whether it is an command with busy. If it is not a busy command, you do not need to check the status on the DATA Line, and directly execute step (5). Otherwise, if it is a busy command, you must execute step (3) to confirm whether it is an Abort command.

(3) If it is an Abort command, it means that when the CMD line completes the transmission, the DATA line is also idle, so you can directly go to step (5); otherwise, if it is not an Abort command, you must go to step (4) to confirm whether the busy on the DATA line has been released.

(4) Check if register bit PRESENT[DAT_INHIBIT] is 0 to confirm whether the DATA line is still in use or not. If it is still in use, wait until the end of transmission, and then perform step (5).

(5) Set the value of ARGUMENT register and CMD register according to the command requirements.

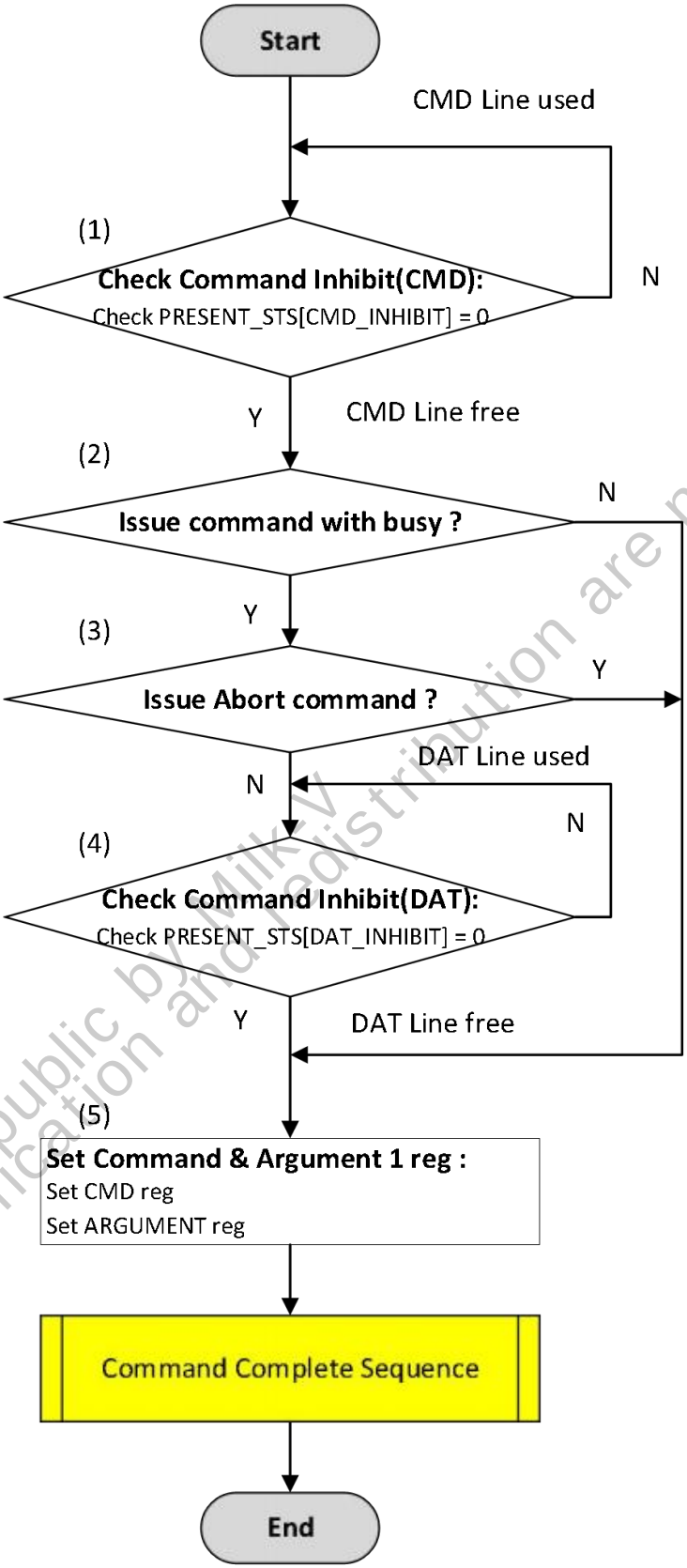


Figure 12- 32 Command Transfer Sequence

Command Complete Sequence

The command completion sequence is shown in Figure 12-33.

(1) First wait for Interrupt NORM_INT_STS[CMD_CMPL] completed by Command.
(2) Set NORM_INT_STS[CMD_CMPL]=1 to clear the CMD_ Cmpl interrupt status after receiving interrupt.
(3) Next read RESP1_0, RESP3_ 2, RESP5_ 4, and RESP7_ 6 to get the response value.
(4) If it is an command containing data transmission, step (5) will be executed; otherwise, skip to step (8).
(5) Wait for data transmission interrupt NORM_INT_STS[XFER_CMPL].
(6) Set NORM_INT_STS[XFER_CMPL]=1 to clear XFER_CMPL interrupt status after receiving interrupt.
(7) Check RESP1_0, RESP3_ 2, RESP5_ 4, and RESP7_ 6 registers to confirm whether there is an error state. If there is no error status, go to step (8) and return that there is no error. If there is an error, perform step (9) to report the error.

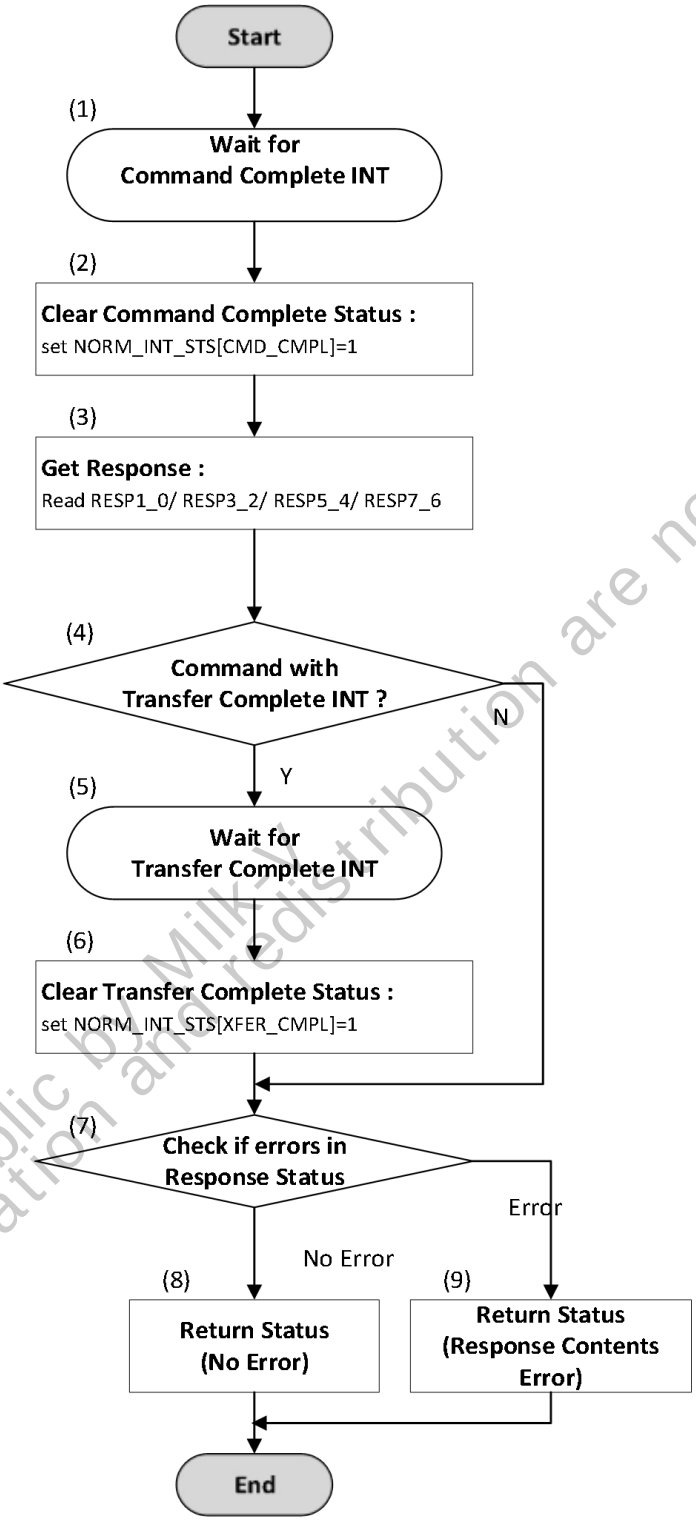


Figure 12- 33 Command Complete Sequence

12.4.2.5 Abort Data Transfer

Abort Command Sequence

For the SD device, the abort command is performed by CMD12, while for the SDIO device, it is performed by CMD52. There are two main conditions of the time to use the abort command.

(1) Stop data transmission of infinite block.

(2) Stop multi block data transmission.

The procedure of abort command is shown in Figure 12-34, and the detailed steps are as follows.

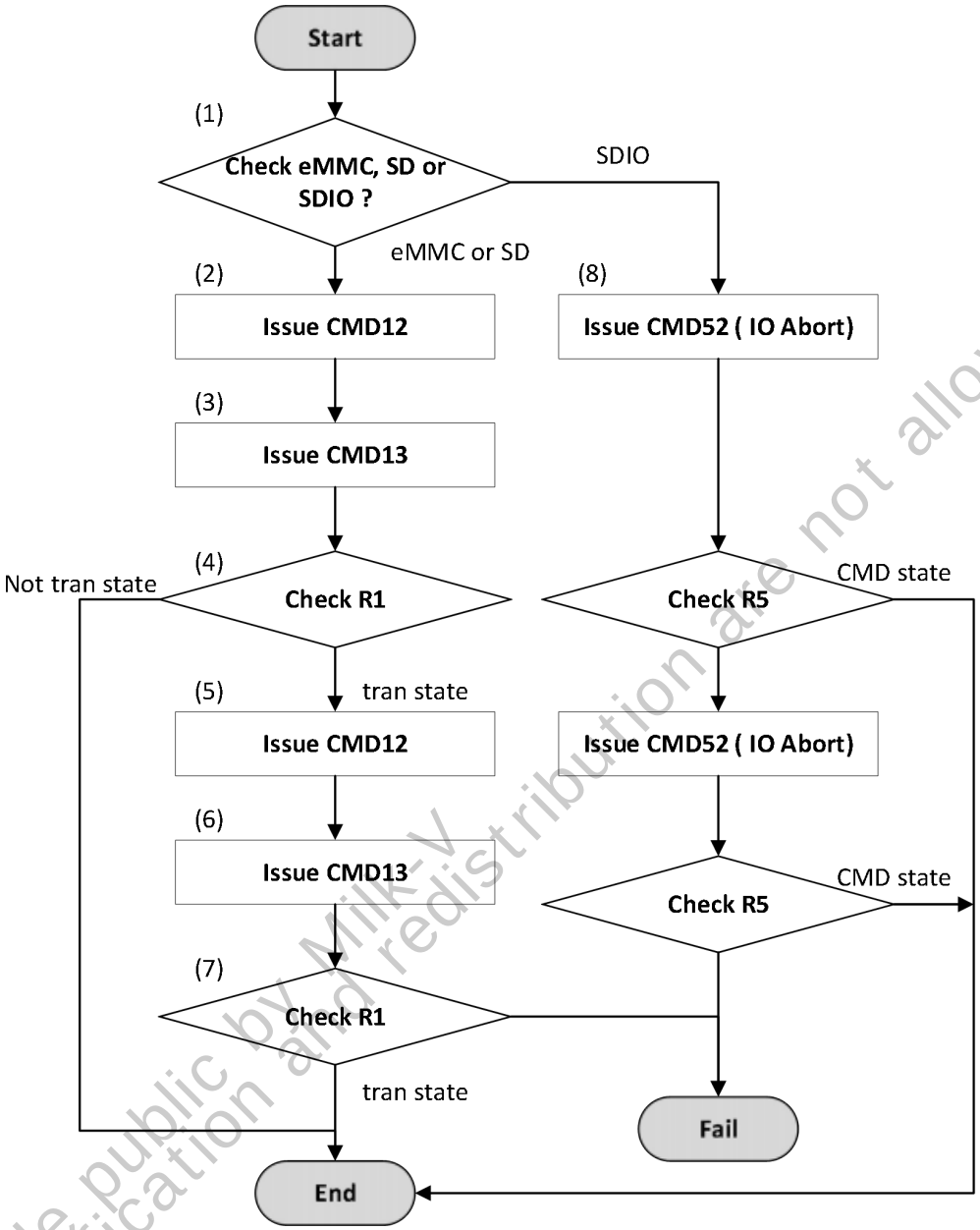


Figure 12- 347 Abort Command Sequence

There are two ways of abort command: synchronous abort command and asynchronous abort command.

Asynchronous Abort Command Sequence

Figure 12-35 shows the diagram of asynchronous abort command. The detailed steps are as follows.

(1) The abort command is executed according to different transmission modes.
(2) Set SW_RST_CMD and SW_RST_DAT in SW_RESET register to reset the signal lines of CMD and DAT.
(3) Check bit SW_RESET[SW_RST_CMD] and SW_RESET[SW_RST_DAT] to confirm whether the reset is completed. If both are 0, the procedure ends. If one of them is 1, return to step (3) to delay waiting.

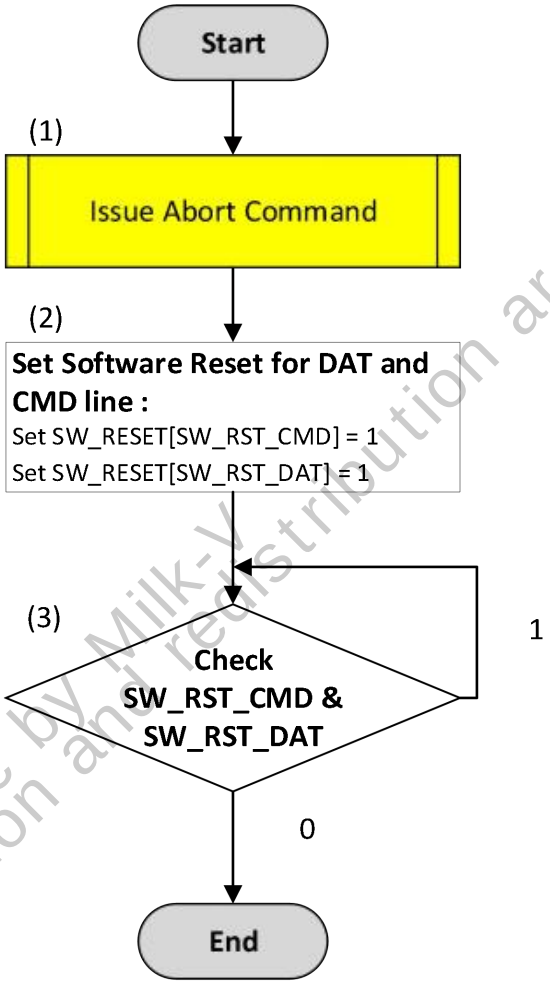


Figure 12- 358 Asynchronous Abort Command Procedure

Synchronous Abort Command Sequence

Figure 12-36 shows the diagram of the synchronous abort command. The detailed steps are as follows.

(1) Write bit BG_CTL[STOP_BG_REQ] to stop the transmission in the Block Gap.

(2) Wait for the completion of interrupt NORM_INT_STS[XFER_CMPL] transferred.
(3) Set NORM_INT_STS[XFER_CMPL]=1 to clear XFER_CMPL interrupt status after receiving interrupt.
(4) The abort command is executed according to different transmission modes.
(5) Set SW_RST_CM and SW_RST_DAT in SW_RESET register to reset the signal lines of CMD and DAT.
(6) Check bit SW_RESET[SW_RST_CMD] and SW_RESET[SW_RST_DAT] to confirm whether the reset is completed. If both are 0, the procedure ends. If one of them is 1, return to step (6) to delay waiting.

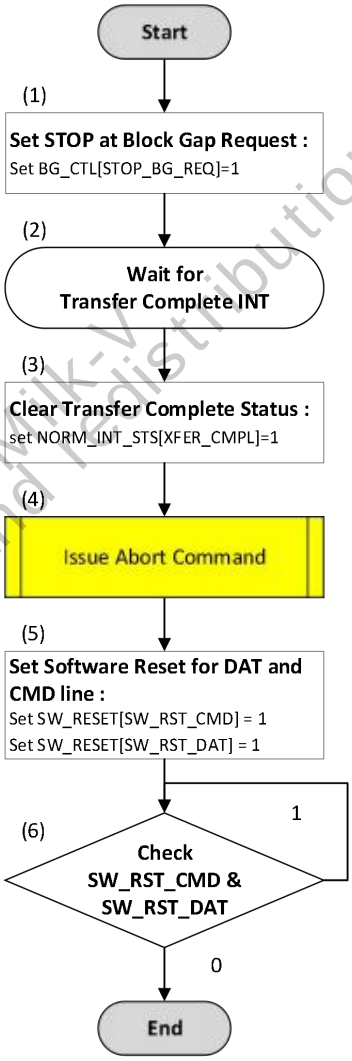


Figure 12- 36 Synchronous Abort Command Procedure

12.4.2.6 Non DMA Data Transfer Mode

The procedure of non DMA data transfer mode is shown in Figure 12-37. The detailed steps are as follows.

(1) Write BLK_SIZE register to set the block size.
(2) Write BLK_ CNT register to set the number of blocks.
(3) Write ARGUMENT register to set the command argument.
(4) Write XFER_MODE register to set the transmission mode. The host can decide the setting according to the situation, including Single or Multiple Block Select, DMA Enable, Block Count Enable, Data Transfer Direction, Auto CMD Enable.
(5) Write CMD register to set the type of command and response.
(6) Wait for the interrupt NORM_INT_STS[CMD_CMPL] completed by Command.
(7) Set NORM_INT_STS[CMD_CMPL]=1 to clear the CMD_CMPL interrupt status after receiving interrupt.
(8) Next read RESP1_0, RESP3_ 2, RESP5_ 4, and RESP7_ 6 to get the response value.
(9) Step (14) is executed for a read operation and step (10) is executed for a write operation.
(10) Wait for the Buffer Write Ready' s interrupt NORM_INT_STS[BUF_WRDY].
(11) Set NORM_INT_STS[BUFF_WRDY]=1 to clear the BUFF_WRDY interrupt status after receiving interrupt.
(12) Write the data in order to the BUF_DATA register.
(13) If there are more blocks to write, go back to step (10) until the last block is written, and then go to step (18).
(14) Wait for the Buffer Write Ready' s interrupt NORM_INT_STS[BUF_RRDY].
(15) Set NORM_INT_STS[BUFF_RRDY]=1 to clear the BUFF_RRDY interrupt status after receiving interrupt.
(16) Read the data from BUF_DATA in order, the data that received from the device.
(17) If there are more blocks to read, go back to step (14), and then go to step (18) until the last block is read.
(18) Determine the transmission is single module transmission, multi module transmission or infinite module transmission. If it is single module or multi module transmission, skip to step (19). If it is infinite module transmission, skip to step (21) and execute abort command.

(19) Wait for the interrupt NORM_INT_STS[XFER_CMPL] after the data transmission is completed.

(20) Set NORM_INT_STS[CMD_XFER]=1 to clear theXFER_CMPL interrupt status after receiving interrupt.

(21) Execute the abort command procedure.

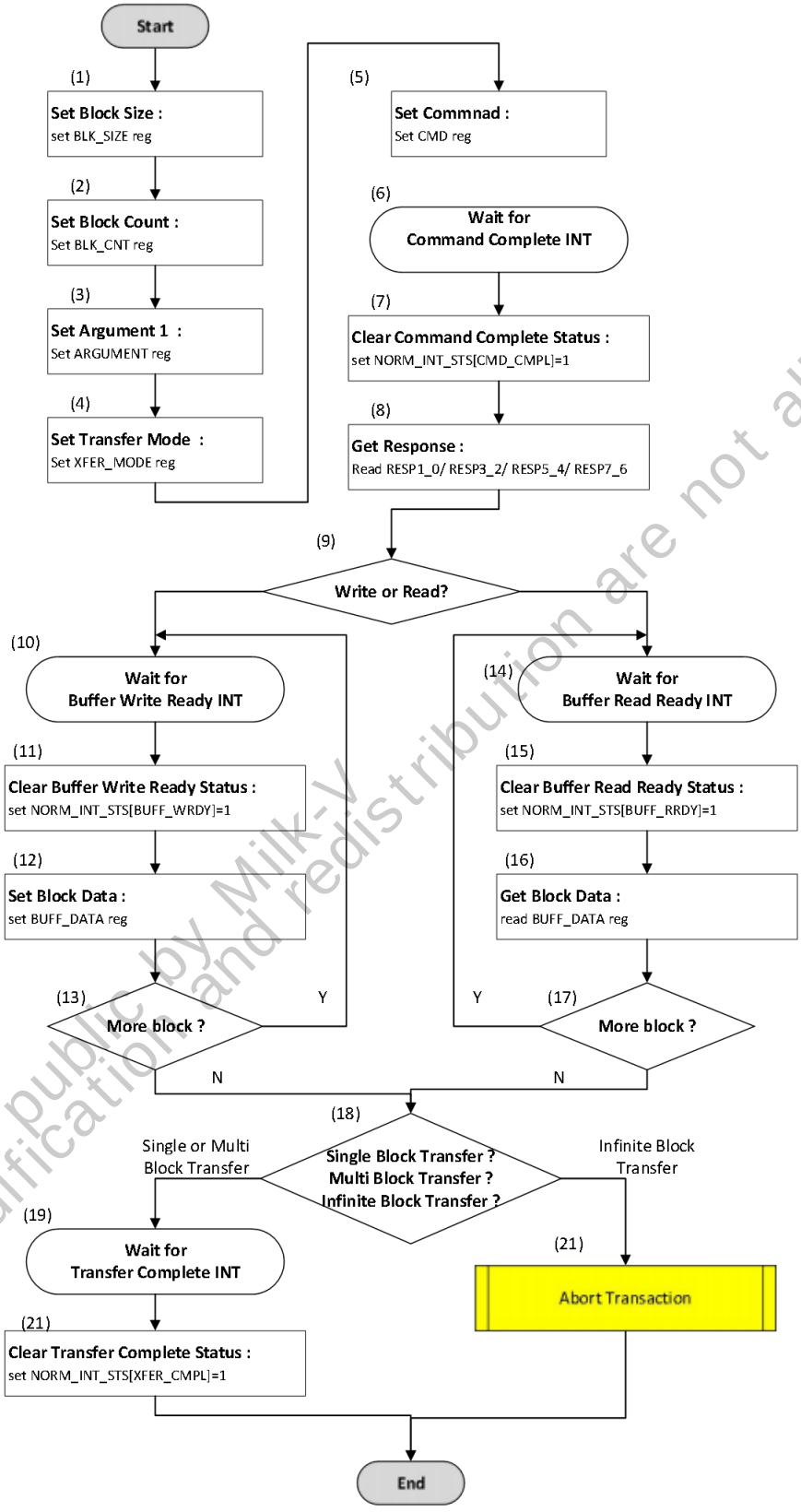


Figure 12- 37 Non DMA Data Transfer Mode Procedure

12.4.2.7 SDMA Data Transfer Mode

The procedure of SDMA data transmission mode is shown in Figure 12-38, and the detailed steps are as follows.

(1) Write SDMA_ SA register to set the starting address of system memory used in data transmission.
(2) Write BLK_ SIZE register to set the block size.
(3) Write BLK_ CNT register to set the number of blocks.
(4) Write the ARGUMENT register to set the command argument.
(5) Write to XFER_MODE register to set the transmission mode. The host can decide the setting according to the situation, including Single or Multiple Block Select, DMA Enable, Block Count Enable, Data Transfer Direction, Auto CMD Enable.
(6) Write to the CMD register to set the type of command and response.
(7) Wait for the interrupt NORM_INT_STS[CMD_CMPL] completed by Command.
(8) Set **NORM_INT_STS[CMD_CMPL]=1** to clear the CMD_ Cmpl interrupt status after receiving interrupt.
(9) Next read RESP1_0, RESP3_ 2, RESP5_ 4, and RESP7_ 6 to get the response value.
(10) Waiting for data transfer interrupt and DMA interrupt
(11) Read interrupt state register NORM_INT_STS to determine the type of interrupt. If it is a DMA interrupt, skip to step (12); if it is a data transmission interrupt, skip to step (14).
(12) Set nNORM_INT_STS[DMA_INT]=1 to clear DMA_ Int status value.
(13) Write to SDMA_SA register to reset the next starting address of the system memory for the DMA, and then skip to step (10).
(14) Set NORM_INT_STS[DMA_INT]=1 and [NORM_INT_STS[XFER_CMPL]=1 to clear DMA_INT and XFER_CMPL status value, and then end the sequence.

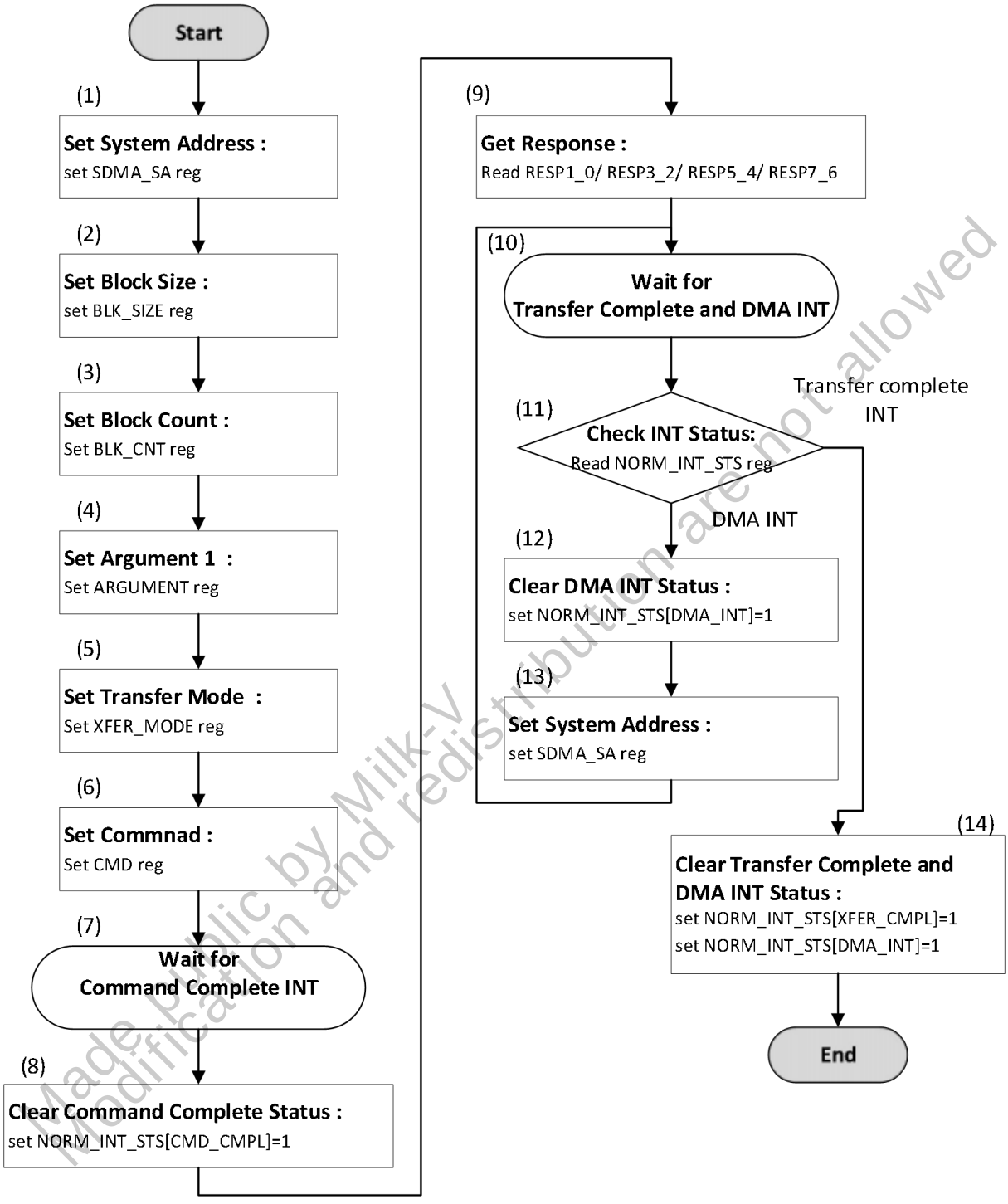


Figure 12- 38 SDMA Data Transfer Mode Procedure

12.4.2.8 ADMA Data Transfer Mode

The procedure of ADMA data transmission mode is shown in Figure 12-39, and the detailed steps are as follows.

(15) Fill ADMA description table into system memory.

(16) Write ADMA_ SA_ L and ADMA_ SA_ H register to set the starting address of the system memory used by the description table.

(17) Write BLK_ SIZE register to set the block size.

(18) Write BLK_ CNT register to set the number of blocks.

(19) Write the ARGUMENT register to set the command argument.

(20) Write to XFER_MODE register to set the transmission mode. The main controller can decide the setting according to the situation, including Single or Multiple Block Select, DMA Enable, Block Count Enable, Data Transfer Direction, Auto CMD Enable.

(21) Write to the CMD register to set the type of command and response.

(22) Wait for the interrupt NORM_INT_STS[CMD_CMPL] completed by Command.

(23) Set NORM_INT_STS[CMD_CMPL]=1 to clear the CMD_ Cmpl interrupt status after receiving interrupt.

(24) Next read RESP1_0, RESP3_ 2, RESP5_ 4, and RESP7_ 6 to get the response value.

(25) Wait for data transmission interrupt or ADMA error interrupt.

(26) Read interrupt state register NORM_INT_STS and ERR_INT_STS to determine the type of interrupt. If it is an ADMA error interrupt, skip to step (13); if it is a data transmission interrupt, skip to step (15).

(27) Set ERR_INT_STS[ADMA_ERR]=1 to clear ADMA_ERR status value.

(28) Enter ADMA abort transaction and execute abort command to abort the data transmission with the device. If necessary, check the ADMA Error Status register to check the cause of the error.

(29) Set NORM_INT_STS[XFER_CMPL]=1 to clear XFER_CMPL status value, and then end the sequence.

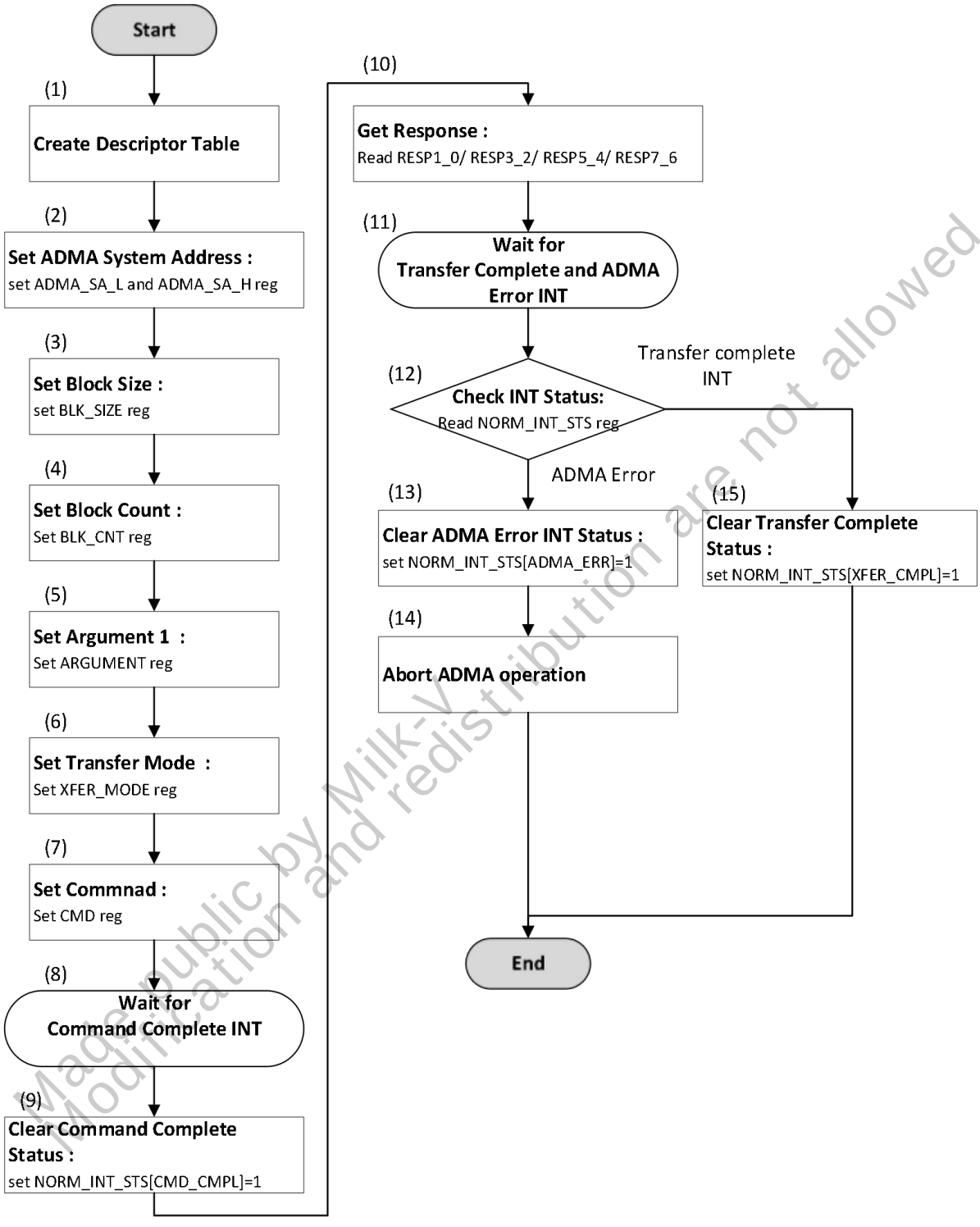


Figure 12- 39 ADMA Data Transfer Mode Procedure

12.4.3 Register Overview

Table 12-6 shows an overview of SD registers (the base address of SDIO0 is 0x0431_0000, while the address of SDIO1 is 0x0500_0000).

Table 12- 2 SD Register Overview

Name	Address Offset	Description
SDMA_SADDR	0x000	SDMA System Memory Address/ Argument2
BLK_SIZE_AND_CNT	0x004	Block Size and Block Count Register
ARGUMENT	0x008	Argument 1 Register
XFER_MODE_AND_CMD	0x00c	Transfer Mode and Command Register
RESP31_0	0x010	Response Bit 31-0 Regsiter
RESP63_32	0x014	Response Bit 63-32 Regsiter
RESP95_64	0x018	Response Bit 95-64 Regsiter
RESP127_96	0x01c	Response Bit 127-96 Regsiter
BUF_DATA	0x020	Buffer Data Port Register
PRESENT_STS	0x024	Present State Register
HOST_CTL1_PWR_BG_WUP	0x028	Host Control 1 , Power, Block Gap and Wakeup Register
CLK_CTL_SWRST	0x02c	Clock and Reset Control Register
NORM_AND_ERR_INT_STS	0x030	Normal and Error Interrupt Status Register
NORM_AND_ERR_INT_STS_EN	0x034	Normal and Error Interrupt Status Enable Register
NORM_AND_ERR_INT_SIG_EN	0x038	Normal and Error Interrupt Signal Enable Register
AUTO_CMD_ERR_AND_HOST_CTL2	0x03c	Auto CMD Error Status Register and Host Control 2 register
CAPABILITIES1	0x040	Capabilities 1 Register
CAPABILITIES2	0x044	Capabilities 2 Register
FORCE_EVENT_ERR	0x050	Force Event Register for Auto CMD Error Status
ADMA_ERR_STS	0x054	ADMA Error Status Register
ADMA_SADDR_L	0x058	ADMA System Address Register for low 32-bit
ADMA_SADDR_H	0x05c	ADMA System Address Register for high 32-bit
PRESENT_VUL_INIT_DS	0x060	Present Value Register for Initialization and Default Speed
PRESENT_VUL_HS_SDR12	0x064	Present Value Register for High-speed and SDR12
PRESENT_VUL_SDR25_SDR50	0x068	Present Value Register for SDR25 and SDR50
PRESENT_VUL_SDR104_DDR50	0x06c	Present Value Register for SDR104 and DDR50
SLOT_INT_AND_HOST_VER	0x0fc	Slot Interrupt Status and Host Controller Version Register
EMMC_CTRL	0x200	MSHC Control register
CDET_TOUT_CTL	0x208	Card Detect Control Register
MBIU_CTRL	0x20c	MBIU Control register
PHY_TX_RX_DLY	0x240	PHY tx and rx delay line register
PHY_DS_DLY	0x244	PHY DS delay line register
PHY_DLY_STS	0x248	PHY delay line status register
PHY_CONFIG	0x24c	PHY Configuration register

12.4.4 Register Description

The following is a detailed description of the registers.

SDMA_SADDR

SDMA System Memory Address/ Argument2
Offset Address: 0x000

Bits	Name	Access	Description	Reset
31:0	SDMA_SA	R/W	Physical system memory address used for DMA transfer and the second argument for Auto CMD23	0x0

BLK_SIZE_AND_CNT

Block Size and Block Count Register
Offset Address: 0x004

Bits	Name	Access	Description	Reset
11:0	XFER_BLK_SIZE	R/W	Block Size of data transfer. - 0x1 : 1 byte - 0x2 : 2 bytes ..... - 0x200 : 512 bytes ..... - 0x800 : 2048 bytes	0x0
14:12	SDMA_BUF_BDARY	R/W	Host SDMA buffer Boundary - 0x0 (4K bytes) - 0x1 (8K bytes) - 0x2 (16K bytes) - 0x3 (32K bytes) - 0x4 (64K bytes) - 0x5 (128K bytes) - 0x6 (256K bytes) - 0x7 (512K bytes)	0x0
15	Reserved			
31:16	BLK_CNT	R/W	Blocks Count for Current Transfer	0x0

ARGUMENT

Argument 1 Register
Offset Address: 0x008

Bits	Name	Access	Description	Reset
31:0	ARGUMENT	R/W	Command Argument 1	0x0

XFER_MODE_AND_CMD

Transfer Mode and Command Register
Offset Address: 0x00c

Bits	Name	Access	Description	Reset
0	DMA_ENABLE	R/W	DMA enable 1 : DMA Data Transfer 0 : No data transfer or Non DMA data transfer	0x0
1	BLK_CNT_ENABLE	R/W	Block Count Enable. This bit is used to enable the block	0x0

Bits	Name	Access	Description	Reset
			count regiser, which is only relevant for multiple block transfers. 1 : Enable 0 : Disable	
3:2	AUTO_CMD_ENABLE	R/W	Auto CMD enable. This field determines use of auto command funcitons 0x0 : Auto command Disabled 0x1 : Auto CMD12 Enable 0x2 : Auto CMD23 Enable 0x3 : Reserved	0x0
4	DAT_XFER_DIR	R/W	Data Transfer Direction Select 1 : Read (card to host) 0 : Write (host to card)	0x0
5	MULTI_BLK_SEL	R/W	Multi/single Block Select 1 : Multiple block transfer 0 : Single block transfer	0x0
6	RESP_TYPE	R/W	Response Type R1/R5 0x0 : R1 (Memory) 0x1 : R5 (SDIO)	0x0
7	RESP_ERR_CHK_ENABLE	R/W	Response Error Check Enable 1 : Enable 0 : Disable	0x0
8	RESP_INT_DISABLE	R/W	Response Interrupt Disable 1 : Disable 0 : Enable	0x0
15:9	Reserved			
17:16	RESP_TYPE_SEL	R/W	Response Type Select 0x0 : No Response 0x1 : Response Length 136 0x2 : Response Length 48 0x3 : Response Length 48 with busy	0x0
18	SUB_CMD_FLAG	R/W	Sub Command Flag 1 : Sub Command 0 : Main Command	0x0
19	CMD_CRC_CHK_ENABLE	R/W	Command CRC check enable 1 : Enable 0 : Disable	0x0
20	CMD_IDX_CHK_ENABLE	R/W	Command Index Check Enable 1 : Enable 0 : Disable	0x0
21	DATA_PRESENT_SEL	R/W	Data Present Select. It is set to 0 for following : (1) Commands using only CMD line (ex. CMD52) (2) Commands with no data transfer but using busy signal on DAT0 (ex. R1b) (3) Resume command 1 : Data Present 0 : No Data Present	0x0
23:22	CMD_TYPE	R/W	Command Type 0x0 : Normal 0x1 : Suspend (CMD52 for writing "Bus Suspend" in CCCR) 0x2 : CMD52 for writing "Function Select" in CCCR) 0x3 : Abort (CMD12, CMD52 for writing	0x0

Bits	Name	Access	Description	Reset
			"I/O Abort" in CCCR)	
29:24	CMD_IDX	R/W	Command Index	0x0
31:30	Reserved			

RESP31_0

Response Bit 31-0 Regsiter
Offset Address: 0x010

Bits	Name	Access	Description	Reset
31:0	RESP31_0	RO	Command Response for RSP[39:8]	

RESP63_32

Response Bit 63-32 Regsiter
Offset Address: 0x014

Bits	Name	Access	Description	Reset
31:0	RESP63_32	RO	Command Response for RSP[71:40]	

RESP95_64

Response Bit 95-64 Regsiter
Offset Address: 0x018

Bits	Name	Access	Description	Reset
31:0	RESP95_64	RO	Command Response for RSP[103:72]	

RESP127_96

Response Bit 127-96 Regsiter
Offset Address: 0x01c

Bits	Name	Access	Description	Reset
31:0	RESP127_96	RO	Command Response for RSP[135:104]	

BUF_DATA

Buffer Data Port Register
Offset Address: 0x020

Bits	Name	Access	Description	Reset
31:0	BUF_DATA	R/W	Buffer Data	0x0

PRESENT_STS

Present State Register
Offset Address: 0x024

Bits	Name	Access	Description	Reset
0	CMD_INHIBIT	RO	Command Inhibit (CMD) 1 : Cannot issue command 0 : Can issue command using only CMD line	
1	CMD_INHIBIT_DAT	RO	Command Inhibit (DAT) 1 : Cannot issue command wihich used the DAT line 0 : Can issue command using only DAT line	
2	DAT_LINE_ACTIVE	RO	DAT Line Active This bit indicates whether one of the DAT line on SD Bus is in use. 1 : DAT Line Active 0 : DAT Line Inactive	

Bits	Name	Access	Description	Reset
3	RE_TUNE_REQ	RO	Re-Tuning Request 1 : Sampling clock need re-tuning 0 : Fixed or well tuned sampling clock	
7:4	Reserved			
8	WR_XFER_ACTIVE	RO	Write Transfer Active 1 : Transferring data 0 : No valid data	
9	RD_XFER_ACTIVE	RO	Read Transfer Active 1 : Transferring data 0 : No valid data	
10	BUF_WR_ENABLE	RO	Buffer Write Enable 1 : Enable 0 : Disable	
11	BUF_RD_ENABLE	RO	Buffer Read Enable 1 : Enable 0 : Disable	
15:12	Reserved			
16	CARD_INSERTED	RO	Card Inserted 1 : Card Inserted 0 : Reset or Debouncing or No card	
17	CARD_STABLE	RO	Card State Stable 1 : No Card or Inserted 0 : Reset or Debouncing	
18	CARD_CD_STS	RO	Card Detect Pin Level 1 : Card Present (SD_CD = 0) 0 : No Card Present (SD_CD = 1)	
19	CARD_WP_STS	RO	Write Protect Switch Pin Level 1 : Write enabled (SD_WP =0) 0 : Write protected (SD_WP = 1)	
23:20	DAT_3_0_STS	RO	DAT[3:0] Line Signal Level	
24	CMD_LINE_STS	RO	CMD Line Signal Level	
31:25	Reserved			

HOST_CTL1_PWR_BG_WUP

Host Control 1 Register
Offset Address: 0x028

Bits	Name	Access	Description	Reset
0	LEC_CTL	R/W	LED Control This bit is used to caution the user not to remove the card while the SD card is being accessed. 1 : LED on 0 : LED off	0x0
1	DAT_XFER_WIDTH	R/W	Data Transfer Width. 1 : 4-bit mode 0 : 1-bit mode	0x0
2	HS_ENABLE	R/W	High Speed Enable 1 : High Speed Enable 0 : Normal Speed Enable	0x0
4:3	DMA_SEL	R/W	DMA Select. 0x0 : SDMA mode 0x1 : Reserved 0x2 : ADMA2 0x3 : ADMA2 or ADMA3	0x0

Bits	Name	Access	Description	Reset
5	EXT_DAT_WIDTH	R/W	Extended Data Transfer Width 1 : 8-bit mode 0 : Selected by DAT_XFER_WIDTH	0x0
6	CRAD_DET_TEST	R/W	Card Detect Test Level 1 : Card Inserted 0 : No card	0x0
7	CARD_DET_SEL	R/W	Card Detect Signal Selection 1 : CARD_DET_TEST is selected (for test purpose) 0 : SD_CD Is selected	0x0
8	SD_BUS_PWR	R/W	SD Bus Power. 1 : Power on 0 : Power off	0x0
11:9	SD_BUS_VOL_SEL	R/W	SD Bus Voltage Select 111b : 3.3V 110b : 3.0V 101b : 1.8V 100b - 000b : Reserved	0x0
15:12	Reserved			
16	STOP_BG_REQ	R/W	Stop At Block Gap Request. This bit is used to stop executing read and write transaction at the next block gap for non-DMA, SDMA and ADMA transfers. 1 : Stop 0 : Transfer	0x0
17	CONTINUE_REQ	R/W	Continue Request. This bit is used to restart a transaction, which was stoped using the STOP_BG_REQ. 1 : Restart 0 : Not affect	0x0
18	READ_WAIT	R/W	Read Wait Control 1 : Enable Read Wait Control 0 : Disable Read Wait Control	0x0
19	INT_BG	R/W	Interrupt At Block Gap 1 : Enable 0 : Disabel	0x0
23:20	Reserved			
24	WAKEUP_ON_CARD_INT	R/W	Wakeup Event Enable On Card Interrupt. 1 : Enable 0 : Disable	0x0
25	WAKEUP_ON_CARD_INSERT	R/W	Wakeup Event Enable On Card Insertion. 1 : Enable 0 : Disable	0x0
26	WAKEUP_ON_CARD_REMV	R/W	Wakeup Event Enable On Card Removal. 1 : Enable 0 : Disable	0x0
31:27	Reserved			

CLK_CTL_SWRST

Clock and Timeout Control Register
Offset Address: 0x02c

Bits	Name	Access	Description	Reset

Bits	Name	Access	Description	Reset
0	INT_CLK_EN	R/W	Internal Clock Enable 1 : Oscillate 0 : Stop	0x0
1	INT_CLK_STABLE	RO	Internal Clock Stable. 1 : Ready 0 : Not Ready	
2	SD_CLK_EN	R/W	SD Clock Enable for Card 1 : Enable 0 : Disable	0x0
3	PLL_EN	R/W	PLL Enable 1 : Enable 0 : Disable	0x0
5:4	Reserved			
7:6	UP_FREQ_SEL	R/W	Upper Bits of SDCLK Frequency Select	0x0
15:8	FREQ_SEL	R/W	SDCLK Frequency Select	0x0
19:16	TOUT_CNT	R/W	Data Timeout Counter Value 0x0 : TMCLK x 2^13 0x1 : TMCLK x 2^14 ..... 0xe : TMCLK x 2^ 27 0xf : Reserved	0x0
23:20	Reserved			
24	SW_RST_ALL	R/W	Software Reset For All	0x0
25	SW_RST_CMD	R/W	Software Reset For CMD Line	0x0
26	SW_RST_DAT	R/W	Software Reset For DATA Line	0x0
31:27	Reserved			

NORM_AND_ERR_INT_STS

Normal and Error Interrupt Status Register
Offset Address: 0x030

Bits	Name	Access	Description	Reset
0	CMD_CMPL	RWC	Command Complete	
1	XFER_CMPL	RWC	Transfer Complete	
2	BG_EVENT	RWC	Block Gap Event	
3	DMA_INT	RWC	DMA Interrupt	
4	BUF_WRDY	RWC	Buffer Write Ready	
5	BUF_RRDY	RWC	Buffer Read Ready	
6	CARD_INSERT_INT	RWC	Card Insertion	
7	CARD_REMOV_INT	RWC	Card Removal	
8	CARD_INT	RO	Card Interrupt	
9	INT_A	RO	INT_A. This status is set if INT_A is enabled and INT_A pin is in low level	
10	INT_B	RO	INT_B. This status is set if INT_B is enabled and INT_B pin is in low level	
11	INT_C	RO	INT_C. This status is set if INT_C is enabled and INT_C pin is in low level	
12	RE_TUNE_EVENT	RO	Re-Tuning Event	
13	Reserved			

Bits	Name	Access	Description	Reset
14	CQE_EVENT	RO	Command Queuing Event	
15	ERR_INT	RO	Error Interrupt	
16	CMD_TOUT_ERR	RWC	Command Timeout Error	
17	CMD_CRC_ERR	RWC	Command CRC Error	
18	CMD_ENDBIT_ERR	RWC	Command End Bit Error	
19	CMD_IDX_ERR	RWC	Command Index Error	
20	DAT_TOUT_ERR	RWC	Data Timeout Error	
21	DAT_CRC_ERR	RWC	Data CRC Error	
22	DAT_ENDBIT_ERR	RWC	Data End Bit Error	
23	CURR_LIMIT_ERR	RWC	Current Limit Error	
24	AUTO_CMD_ERR	RWC	Auto Command Error	
25	ADMA_ERR	RWC	ADMA Error	
26	TUNE_ERR	RWC	Tuning Error	
27	Reserved			
28	BOOT_ACK_ERR	RWC		
31:29	Reserved			

NORM_AND_ERR_INT_STS_EN

Normal and Error Interrupt Status Enable Register
Offset Address: 0x034

Bits	Name	Access	Description	Reset
0	CMD_CMPL_EN	R/W	Command Complete Status Enable	0x0
1	XFER_CMPL_EN	R/W	Transfer Complete Status Enable	0x0
2	BG_EVENT_EN	R/W	Block Gap Event Status Enable	0x0
3	DMA_INT_EN	R/W	DMA Interrupt Status Enable	0x0
4	BUF_WRDY_EN	R/W	Buffer Write Ready Status Enable	0x0
5	BUF_RRDY_EN	R/W	Buffer Read Ready Status Enabel	0x0
6	CARD_INSERT_INT_EN	R/W	Card Insertion Status Enable	0x0
7	CARD_REMOV_INT_EN	R/W	Card Removal Status Enable	0x0
8	CARD_INT_EN	R/W	Card Interrupt Status Enable	0x0
9	INT_A_EN	R/W	INT_A Status Enable.	0x0
10	INT_B_EN	R/W	INT_B Status Enable.	0x0
11	INT_C_EN	R/W	INT_C Status Enable.	0x0
12	RE_TUNE_EVENT_EN	R/W	Re-Tuning Event Status Enable	0x0
13	Reserved			
14	CQE_EVENT_EN	R/W	Command Queuing Event Status Enable	0x0
15	ERR_INT_EN	R/W	Error Interrupt Status Enable	0x0
16	CMD_TOUT_ERR_EN	R/W	Command Timeout Error Status Enable	0x0
17	CMD_CRC_ERR_EN	R/W	Command CRC Error Status Enable	0x0
18	CMD_ENDBIT_ERR_EN	R/W	Command End Bit Error Status Enable	0x0
19	CMD_IDX_ERR_EN	R/W	Command Index Error Status Enable	0x0
20	DAT_TOUT_ERR_EN	R/W	Data Timeout Error Status Enable	0x0
21	DAT_CRC_ERR_EN	R/W	Data CRC Error Status Enable	0x0
22	DAT_ENDBIT_ERR_EN	R/W	Data End Bit Error Statue Enable	0x0
23	CURR_LIMIT_ERR_EN	R/W	Current Limit Error Status Enable	0x0
24	AUTO_CMD_ERR_EN	R/W	Auto Command Error Status Enable	0x0
25	ADMA_ERR_EN	R/W	ADMA Error Status Enable	0x0
26	TUNE_ERR_EN	R/W	Tuning Error Status Enable	0x0
27	Reserved			

Bits	Name	Access	Description	Reset
28	BOOT_ACK_ERR_EN	R/W	Boot Ack Error Status Enable	0x0
31:29	Reserved			

NORM_AND_ERR_INT_SIG_EN

Normal and Error Interrupt Signal Enable Register
Offset Address: 0x038

Bits	Name	Access	Description	Reset
0	CMD_CMPL_SIG_EN	R/W	Command Complete Signal Enable	0x0
1	XFER_CMPL_SIG_EN	R/W	Transfer Complete Signal Enable	0x0
2	BG_EVENT_SIG_EN	R/W	Block Gap Event Signal Enable	0x0
3	DMA_INT_SIG_EN	R/W	DMA Interrupt Signal Enable	0x0
4	BUF_WRDY_SIG_EN	R/W	Buffer Write Ready Signal Enable	0x0
5	BUF_RRDY_SIG_EN	R/W	Buffer Read Ready Signal Enabel	0x0
6	CARD_INSERT_INT_SIG_EN	R/W	Card Insertion Signal Enable	0x0
7	CARD_REMOV_INT_SIG_EN	R/W	Card Removal Signal Enable	0x0
8	CARD_INT_SIG_EN	R/W	Card Interrupt Signal Enable	0x0
9	INT_A_SIG_EN	R/W	INT_A Signal Enable.	0x0
10	INT_B_SIG_EN	R/W	INT_B Signal Enable.	0x0
11	INT_C_SIG_EN	R/W	INT_C Signal Enable.	0x0
12	RE_TUNE_EVENT_SIG_EN	R/W	Re-Tuning EventSignal Enable	0x0
13	Reserved			
14	CQE_EVENT_SIG_EN	R/W	CQE EventSignal Enable	0x0
15	Reserved			
16	CMD_TOUT_ERR_SIG_EN	R/W	Command Timeout Error Signal Enable	0x0
17	CMD_CRC_ERR_SIG_EN	R/W	Command CRC Error Signal Enable	0x0
18	CMD_ENDBIT_ERR_SIG_EN	R/W	Command End Bit Error Signal Enable	0x0
19	CMD_IDX_ERR_SIG_EN	R/W	Command Index Error Signal Enable	0x0
20	DAT_TOUT_ERR_SIG_EN	R/W	Data Timeout Error Signal Enable	0x0
21	DAT_CRC_ERR_SIG_EN	R/W	Data CRC Error Signal Enable	0x0
22	DAT_ENDBIT_ERR_SIG_EN	R/W	Data End Bit Error Signal Enable	0x0
23	CURR_LIMIT_ERR_SIG_EN	R/W	Current Limit Error Signal Enable	0x0
24	AUTO_CMD_ERR_SIG_EN	R/W	Auto Command Error Signal Enable	0x0
25	ADMA_ERR_SIG_EN	R/W	ADMA Error Signal Enable	0x0
26	TUNE_ERR_SIG_EN	R/W	Tuning Error Signal Enable	0x0
27	Reserved			
28	BOOT_ACK_ERR_SIG_EN	R/W	Boot Ack Error Signal Enable	0x0
31:29	Reserved			

AUTO_CMD_ERR_AND_HOST_CTL2

Auto CMD Error Status Register and Host Control 2 register
Offset Address: 0x03c

Bits	Name	Access	Description	Reset
0	AUTO_CMD12_NO_EXE	RO	Auto CMD12 Not Executed	
1	AUTO_CMD_TOUT_ERR	RO	Auto CMD Timeout Error	
2	AUTO_CMD_CRC_ERR	RO	Auto CMD CRC Error	
3	AUTO_CMD_ENDBIT_ERR	RO	Auto CMD End Bit Error	
4	AUTO_CMD_IDX_ERR	RO	Auto CMD Index Error	
6:5	Reserved			
7	CMD_NOT_ISSUE_BY_CMD12	RO	Command Not Issued By Auto CMD12 Error	
15:8	Reserved			

Bits	Name	Access	Description	Reset
18:16	UHS_MODE_SEL	R/W	USH Speed Mode Select (for SD) 0x0 : SDR12 0x1 : SDR25 0x2 : SDR50 0x3 : SDR104 0x4 : DDR50 0x5 : Reserved 0x6 : Reserved 0x7 : Reserved eMMC Speed Mode Select (for eMMC) 0x0 : Default speed 0x1 : High speed 0x2 : Reserved 0x3 : HS200 0x4 : DDR52 0x5 : Reserved 0x6 : Reserved 0x7 : Reserved	0x0
19	EN_18_SIG	R/W	1.8V Signaling Enable	0x0
21:20	DRV_SEL	R/W	Driver Strength Select 0x0 : Driver Type B 0x1 : Driver Type A 0x2 : Driver Type C 0x3 : Driver Type D	0x0
22	EXECUTE_TUNE	R/W	Execute Tuning 1 : Execute Tuning 0 : Not Tuned or Tuning Completed	0x0
23	SAMPLE_CLK_SEL	R/W	Sampling Clock Select 1 : Tuned clock is used to sample data 0 : Fixed clock is used to sample data	0x0
29:24	Reserved			
30	ASYNC_INT_EN	R/W	Asynchronous Interrupt Enable. 1 : Enable 0 : Disable	0x0
31	PRESET_VAL_ENABLE	R/W	Preset Value Enable 1 : Automatic Selection by Preset Value are Enabled 0 : SDLCK and Driver Strength are controlled by Host Driver	0x0

CAPABILITIES1

Capabilities 1 Register

Offset Address: 0x040

Bits	Name	Access	Description	Reset
5:0	TOUT_CLK_FREQ	RO	Timeout Clock Frequency Not 0 : 1KHz~ 63KHz or 1Mhz~ 63Mhz	
6	Reserved			
7	TOUT_CLK_UNIT	RO	Timeout Clock Unit 1 : 1MHz 0 : 1KHz	
15:8	BASE_CLK_FREQ	RO	Base Clock Frequency for SD clock 0x0 : Get information through another method 0x1 : 1MHz 0x2 : 2MHz	

Bits	Name	Access	Description	Reset
			0xFF : 255Mhz	
17:16	MAX_BLK_LEN	RO	Max Block Length 0x0 : 512 (byte) 0x1 : 1024 0x2 : 2048 0x3 : Reserverd	
18	EMBEDDED_8BIT	RO	8-bit Support for Embedded Device	
19	ADMA2_SUPPORT	RO	ADMA2 Support	
20	Reserved			
21	HS_SUPPORT	RO	High Speed Support	
22	SDMA_SUPPORT	RO	SDMA Support	
23	SUSP_RES_SUPPORT	RO	Suspend/Resume Support	
24	V33_SUPPORT	RO	3.3V Support	
25	V30_SUPPORT	RO	3.0V Support	
26	V18_SUPPORT	RO	1.8V Support	
27	Reserved			
28	BUS64_SUPPORT	RO	64-bit System Bus Support	
29	ASYNC_INT_SUPPORT	RO	Asynchronous Interrupt Support	
31:30	SLOT_TYPE	RO	Slot Type 0x0 : Removable Card 0x1 : Embedded Slot 0x2 : Shared Bus Slot	

CAPABILITIES2

Capabilities 2 Register
Offset Address: 0x044

Bits	Name	Access	Description	Reset
0	SDR50_SUPPORT	RO	SDR50 Support	
1	SDR104_SUPPORT	RO	SDR104 Support	
2	DDR50_SUPPORT	RO	DDR50 Support	
3	Reserved			
4	DRV_A_SUPPORT	RO	Driver Type A Support	
5	DRV_C_SUPPORT	RO	Driver Type C Support	
6	DRV_D_SUPPORT	RO	Driver Type D Support	
7	Reserved			
11:8	RETUNE_TIMER	RO	Timer Count for Re-Tuning 0x0 : Disable n : 2^(n-1) seconds 0xB : 1024 seconds 0xC ~ 0xE : Reserved 0xF : Get Information from other source	
12	Reserved			
13	TUNE_SDR50	RO	Use Tuning for SDR50	
15:14	RETUNE_MODE	RO	Re-Tuning Modes	
23:16	CLK_MULTIPLIER	RO	Clock Multiplier	
31:24	Reserved			

FORCE_EVENT_ERR

Force Event Register for Auto CMD Error Status
Offset Address: 0x050

Bits	Name	Access	Description	Reset
0	FORCE_AUTO_CMD12_NOT_EXE	R/W	Force Event for Auto CMD12 Not Executed	0x0
1	FORCE_AUTO_CMD_TOUT_ERR	R/W	Force Event for Auto CMD Timeout Error	0x0
2	FORCE_AUTO_CMD_CRC_ERR	R/W	Force Event for Auto CMD CRC Error	0x0
3	FORCE_AUTO_CMD_EBIT_ERR	R/W	Force Event for Auto CMD End Bit Error	0x0
4	FORCE_AUTO_CMD_IDX_ERR	R/W	Force Event for Auto CMD Index Error	0x0
6:5	Reserved			
7	FORCE_AUTO_CMD_NOT_ISSUE	R/W	Force Event for Command Not Issued By Auto CMD12 Error	0x0
15:8	Reserved			
16	FORCE_CMD_TOUT_ERR	R/W	Force Event for Auto CMD12 Not Executed	0x0
17	FORCE_CMD_CRC_ERR	R/W	Force Event for CMD Timeout Error	0x0
18	FORCE_CMD_EBIT_ERR	R/W	Force Event for CMD End Bit Error	0x0
19	FORCE_CMD_IDX_ERR	R/W	Force Event for CMD Index Error	0x0
20	FORCE_DAT_TOUT_ERR	R/W	Force Event for DATA Timeout Error	0x0
21	FORCE_DAT_CRC_ERR	R/W	Force Event for DATA End Bit Error	0x0
22	FORCE_DAT_EBIT_ERR	R/W	Force Event for DATA Index Error	0x0
23	FORCE_CURR_LIMIT_ERR	R/W	Force Event for current limit error	0x0
24	FORCE_AUTO_CMD_ERR	R/W	Force Event for Auto CMD Error	0x0
25	FORCE_ADMA_ERR	R/W	Force Event for ADMA Error	0x0
26	FORCE_TUNING_ERR	R/W	Force Event for Tuning Error	0x0
27	Reserved			
28	FORCE_BOOT_ACK_ERR	R/W	Force Event for Response Error	0x0
31:29	Reserved			

ADMA_ERR_STS

ADMA Error Status Register
Offset Address: 0x054

Bits	Name	Access	Description	Reset
1:0	ADMA_ERR_STS	RO	ADMA Error Status 0x0 : ST_STOP (Stop DMA) 0x1 : ST_FDS (Fetch Descriptor) 0x2 : Never set this state 0x3 : ST_TFR (transfer data)	
2	ADMA_LEN_MISMATCH	RO	ADMA Length Mismatch Error	
31:3	Reserved			

ADMA_SADDR_L

ADMA System Address Register for low 32-bit
Offset Address: 0x058

Bits	Name	Access	Description	Reset
31:0	ADMA_SA_L	R/W	ADMA System Address for low 32-bit	0x0

ADMA_SADDR_H

ADMA System Address Register for high 32-bit
Offset Address: 0x05c

Bits	Name	Access	Description	Reset

Bits	Name	Access	Description	Reset
31:0	ADMA_SA_H	R/W	ADMA System Address for high 32-bit	0x0

PRESENT_VUL_INIT_DS

Present Value Register for Initialization and Default Speed
Offset Address: 0x060

Bits	Name	Access	Description	Reset
31:0	PRESENT_VUL_INIT_DS	RO	Present Value Register for Initialization and Default Speed	

PRESENT_VUL_HS_SDR12

Present Value Register for High-speed and SDR12
Offset Address: 0x064

Bits	Name	Access	Description	Reset
31:0	PRESENT_VUL_HS_SDR12	RO	Present Value Register for High-speed and SDR12	

PRESENT_VUL_SDR25_SDR50

Present Value Register for SDR25 and SDR50
Offset Address: 0x068

Bits	Name	Access	Description	Reset
31:0	PRESENT_VUL_SDR25_SDR50	RO	Present Value Register for SDR25 and SDR50	

PRESENT_VUL_SDR104_DDR50

Present Value Register for SDR104 and DDR50
Offset Address: 0x06c

Bits	Name	Access	Description	Reset
31:0	PRESENT_VUL_SDR104_DDR50	RO	Present Value Register for SDR104 and DDR50	

SLOT_INT_AND_HOST_VER

Slot Interrupt Status and Host Controller Version Register
Offset Address: 0x0fc

Bits	Name	Access	Description	Reset
7:0	INT_SLOT	RO	Interrupt Signal for Each Slot	
15:8	Reserved			
23:16	SPEC_VER	RO	Specification Version Number 00h : SD Host 1.00 01h : SD Host 2.00 02h : SD Host 3.00 03h : SD Host 4.00 04h : SD Host 4.10 05h : SD Host 4.20	
31:24	VENDOR_VER	RO	Verdor Version Number	

EMMC_CTRL

MSHC Control register
Offset Address: 0x200

Bits	Name	Access	Description	Reset
0	EMMC_FUNC_EN	R/W	eMMC Card present	0x0
1	LATANCY_1T	R/W	Latancy 1t for cmd in	0x1
2	CLK_FREE_EN	R/W	Internal clock gating disable control	0x0
3	DISABLE_DATA_CRC_CHK	R/W	Disable Data CRC Check	0x0

Bits	Name	Access	Description	Reset
7:4	Reserved			
8	EMMC_RSTN	R/W	EMMC Device Reset Signal control	0x1
9	EMMC_RSTN_OEN	R/W	Output Enable control for EMMC Device Reset Signal PAD	0x1
11:10	Reserved			
12	CQE_ALGO_SEL	R/W	Scheduler algorithm selected for execution 1 : First come First serve (FCFS_ONLY) 0 : Priority based reordering with FCFS (PRI_REORDER_PLUS_FCFS)	0x0
13	CQE_PREFETCH_DISABLE	R/W	Enable or Disable CQE's PREFETCH Feature 1 : Disable 0 : Enable	0x0
15:14	Reserved			
16	timer_clk_sel	R/W	timer clock source selection 1 : 32K 0 : 100K	0x0
31:17	Reserved			

CDET_TOUT_CTL

Card Detect Control Register
Offset Address: 0x208

Bits	Name	Access	Description	Reset
15:0	CDET_DEBUUNCE_CNT	R/W	card detect debounce counter	0x000F
31:16	Reserved			

MBIU_CTRL

MBIU Control register
Offset Address: 0x20c

Bits	Name	Access	Description	Reset
0	UNDEFL_INCR_EN	R/W	Undefined INCR Burst	0x1
1	BURST_INCR4_EN	R/W	INCR4 Burst	0x1
2	BURST_INCR8_EN	R/W	INCR8 Burst	0x1
3	BURST_INCR16_EN	R/W	INCR16 Burst	0x1
31:4	Reserved			

PHY_TX_RX_DLY

PHY tx and rx delay line register
Offset Address: 0x240

Bits	Name	Access	Description	Reset
6:0	PHY_TX_DLY	R/W	PHY tx delay line phase selection	0x0
7	Reserved			
9:8	PHY_TX_SRC	R/W	PHY tx delay line clock source selection 2'b00 : clk_tx 2'b01 : inverse of clk_tx 2'b1x : reserved	0x0
10	PHY_TX_EVEN_ODD	R/W	PHY tx delay line clock source selection	0x0
15:11	Reserved			
22:16	PHY_RX_DLY	R/W	PHY rx delay line phase selection 2'b00 : clk_tx 2'b01 : inverse of clk_tx	0x0

Bits	Name	Access	Description	Reset
			2'b1x : reserved	
23	Reserved			
25:24	PHY_RX_SRC	R/W	PHY rx delay line clock source selection	0x0
26	PHY_RX_EVEN_ODD	R/W	PHY rx delay line clock source selection	0x0
31:27	Reserved			

PHY_DS_DLY

PHY DS delay line register
Offset Address: 0x244

Bits	Name	Access	Description	Reset
6:0	PHY_DS_DLY	R/W	PHY DS delay line phase selection	0x0
7	Reserved			
9:8	PHY_DS_SRC	R/W	PHY DS delay line clock source selection	0x0
10	PHY_DS_EVEN_ODD	R/W	PHY DS delay line clock source selection	0x0
31:11	Reserved			

PHY_DLY_STS

PHY delay line status register
Offset Address: 0x248

Bits	Name	Access	Description	Reset
0	PHY_TX_LEAD_LAG	RO	PHY tx delay line lead or lag flag	
1	PHY_RX_LEAD_LAG	RO	PHY rx delay line lead or lag flag	
2	PHY_DS_LEAD_LAG	RO	PHY ds delay line lead or lag flag	
31:3	Reserved			

PHY_CONFIG

PHY Configuration register
Offset Address: 0x24c

Bits	Name	Access	Description	Reset
0	PHY_TX_BPS	R/W	PHY tx data path bypass enable 0 : Pipe enable 1 : Bypass	0x1
1	ADJ_TIMING_EN	R/W	Adjust bus timing enable	0x0
7:2	Reserved			
9:8	ADJ_NCR	R/W	Adjust NCR counter	0x0
11:10	ADJ_NCRC	R/W	Adjust NCRC counter	0x0
31:12	Reserved			

12.5 GPIO

12.5.1 Overview

The system is equipped with four groups of GPIO (General Purpose Input/Output), namely GPIO0, GPIO1, GPIO2, and GPIO3. Each group of GPIO provides 32 programmable Input / Output pins.
The direction of each pin can be arbitrarily set as input or output to generate the output signal of a specific application or collect the input signal of a specific application. When set to input pin, GPIO can be used as interrupt source; when set to output pin, each GPIO can output 0 or 1 independently.
GPIO can generate maskable interrupt according to the level or value change of input signal(level or edge sensitive). GPIOx_INTR_FLAG（x=0~3）signal gives an indication to the interrupt controller, indicating that an interrupt has occurred.

12.5.2 Characteristics

The direction of each pin can be set as input or output.
When set to input pin, GPIO can be used as interrupt source.
When set to output pin, each GPIO can output 0 or 1 independently.

12.5.3 Working Mode

12.5.3.1 Interface Reset

When the chip is powered on or the system is reset, the four GPIO modules are reset at the same time, and the GPIO pins are in the input state by default after reset.

12.5.3.2 General Input and Output

Each pin can be set as input or output at will. The steps are as follows.

Step 1 Configure register GPIO_SWPORTA_DDR and set GPIO as input or output.
Step 2 When the input pin is configured, read GPIO_EXT_PORTA register in order to check the value of the input signal; when the output pin is configured, write the

output value to the GPIO_SWPORTA_DR register to control the GPIO output level.

12.5.3.3 Interrupt Operation

Each GPIO can be used as an interrupt source through the control of nine registers including GPIO_INTEN and so on. By these registers, the user can select the interrupt source, interrupt level polarity and edge trigger characteristics.
When multiple GPIO interrupts occur at the same time, one interrupt will be aggregated for reporting (each of the four groups of GPIO will have a collective flag broken reporting).
The characteristics of interrupt source and the type of interrupt trigger are determined by the following five registers: GPIO_INTTYPE_LEVEL, GPIO_INT_POLARITY, GPIO_INTMASK, GPIO_DEBOUNCE, and GPIO_LS_SYNC.
The original state and masked state of the interrupt are read through GPIO_RAW_INTSTATUS and GPIO_INTSTATUS. By setting GPIO_PORTA_EOI, that can control the interrupt state clearing.
Each GPIO can support interrupt. The setting steps are as follows.

Step 1 Configure register GPIO_INTTYPE_LEVEL and select level trigger or edge trigger.

Step 2 Configure register GPIO_INT_POLARITY and select low level / high level trigger as well as falling edge / rising edge trigger.

Step 3 Write 0xFFFFFFFF to register GPIO_PORTA_EOI to clear the interrupt.

步骤 4 Configure GPIO_INTEN register; enable GPIO pin interrupt function.

12.5.4 GPIORegister Overview

The base addresses of the four GPIO modules are shown in Table 12-7.

Table 12- 3 Four GPIO Module Base Addresses of the Chip

GPIO Module	Base Address
GPIO0	0x03020000
GPIO1	0x03021000
GPIO2	0x03022000
GPIO3	0x03023000
RTCSYS_GPIO	0x05021000

Table 12-8 shows the offset addresses and definitions of the first group of GPIO module (GPIO0) registers. GPIO0 to GPIO3 have the same register definitions.

Table 12- 9 GPIO Register Overview

Name	Address Offset	Description
GPIO_SWPORTA_DR	0x000	Port A data register
GPIO_SWPORTA_DDR	0x004	Port A data direction register
GPIO_INTEN	0x030	Interrupt enable register
GPIO_INTMASK	0x034	Interrupt mask register
GPIO_INTTYPE_LEVEL	0x038	Interrupt level register
GPIO_INT_POLARITY	0x03c	Interrupt polarity register
GPIO_INTSTATUS	0x040	Interrupt status of Port A
GPIO_RAW_INTSTATUS	0x044	Raw interrupt status of Port A (pre-masking)
GPIO_DEBOUNCE	0x048	Debounce enable register
GPIO_PORTA_EOI	0x04c	Port A clear interrupt register
GPIO_EXT_PORTA	0x050	Port A external port register
GPIO_LS_SYNC	0x060	Level-sensitive synchronization enable register

12.5.5 GPIO Register Description

GPIO_SWPORTA_DR

Offset Address: 0x000

Bits	Name	Access	Description	Reset
31:0	GPIO_SWPORTA_DR	R/W	Values written to this register are output on the I/O signals for Port A if the corresponding data direction bits for Port A are set to Output mode and the corresponding control bit for Port A is set to Software mode. The value read back is equal to the last value written to this register.	0x0

GPIO_SWPORTA_DDR

Offset Address: 0x004

Bits	Name	Access	Description	Reset
31:0	GPIO_SWPORTA_DDR	R/W	Values written to this register independently control the direction of the corresponding data bit in Port A. The default direction can be configured as input or output after system reset through the GPIO_DFLT_DIR_A parameter. 0 – Input (default)	0x0

Bits	Name	Access	Description	Reset
			1 – Output	

GPIO_INTEN

Offset Address: 0x030

Bits	Name	Access	Description	Reset
31:0	GPIO_INTEN	R/W	Allows each bit of Port A to be configured for interrupts. By default the generation of interrupts is disabled. Whenever a 1 is written to a bit of this register, it configures the corresponding bit on Port A to become an interrupt; otherwise, Port A operates as a normal GPIO signal. Interrupts are disabled on the corresponding bits of Port A if the corresponding data direction register is set to Output or if Port A mode is set to Hardware. 0 – Configure Port A bit as normal GPIO signal (default) 1 – Configure Port A bit as interrupt	0x0

GPIO_INTMASK

Offset Address: 0x034

Bits	Name	Access	Description	Reset
31:0	GPIO_INTMASK	R/W	Controls whether an interrupt on Port A can create an interrupt for the interrupt controller by not masking it. By default, all interrupts bits are unmasked. Whenever a 1 is written to a bit in this register, it masks the interrupt generation capability for this signal; otherwise interrupts are allowed through. The unmasked status can be read as well as the resultant status after masking. 0 – Interrupt bits are unmasked (default) 1 – Mask interrupt	0x0

GPIO_INTTYPE_LEVEL

Offset Address: 0x038

Bits	Name	Access	Description	Reset
31:0	GPIO_INTTYPE_LEVEL	R/W	Controls the type of interrupt that can occur on Port A. Whenever a 0 is written to a bit of this register, it configures the interrupt type to be level-sensitive; otherwise, it is edge-sensitive. 0 – Level-sensitive (default) 1 – Edge-sensitive	0x0

GPIO_INT_POLARITY

Offset Address: 0x03c

Bits	Name	Access	Description	Reset
31:0	GPIO_INT_POLARITY	R/W	Controls the polarity of edge or level sensitivity that can occur on input of Port A. Whenever a 0 is written to a bit of this register, it configures the interrupt type to falling-edge or active-low sensitive; otherwise, it is rising-edge or active-high sensitive. 0 – Active-low (default) 1 – Active-high	0x0

GPIO_INTSTATUS

Offset Address: 0x040

Bits	Name	Access	Description	Reset
31:0	GPIO_INTSTATUS	RO	Interrupt status of Port A	

GPIO_RAW_INTSTATUS

Offset Address: 0x044

Bits	Name	Access	Description	Reset
31:0	GPIO_RAW_INTSTATUS	RO	Raw interrupt of status of Port A (premasking bits)	

GPIO_DEBOUNCE

Offset Address: 0x048

Bits	Name	Access	Description	Reset
31:0	GPIO_DEBOUNCE	R/W	Controls whether an external signal that is the source of an interrupt needs to be debounced to remove any spurious glitches. Writing a 1 to a bit in this register enables the debouncing circuitry. A signal must be valid for two periods of an external clock before it is internally processed. 0 – No debounce (default) 1 – Enable debounce	0x0

GPIO_PORTA_EOI

Offset Address: 0x04c

Bits	Name	Access	Description	Reset
31:0	GPIO_PORTA_EOI	R/W	Controls the clearing of edge type interrupts from Port A. When a 1 is written into a corresponding bit of this register, the interrupt is cleared. All interrupts are cleared when Port A is not configured for interrupts. 0 – No interrupt clear (default) 1 – Clear interrupt	0x0

GPIO_EXT_PORTA

Offset Address: 0x050

Bits	Name	Access	Description	Reset
31:0	GPIO_EXT_PORTA	RO	When Port A is configured as Input, then reading this location reads the values on the signal. When the data direction of Port A is set as Output, reading this location reads the data register for Port A.	

GPIO_LS_SYNC

Offset Address: 0x060

Bits	Name	Access	Description	Reset
0	GPIO_LS_SYNC	R/W	[0] Synchronization level Writing a 1 to this register results in all level-sensitive interrupts being synchronized to pclk_intr. 0 – No synchronization to pclk_intr (default) 1 – Synchronize to pclk_intr	0x0
31:1	Reserved			

12.6 USB DRD (Dual Role Device)

12.6.1 Overview

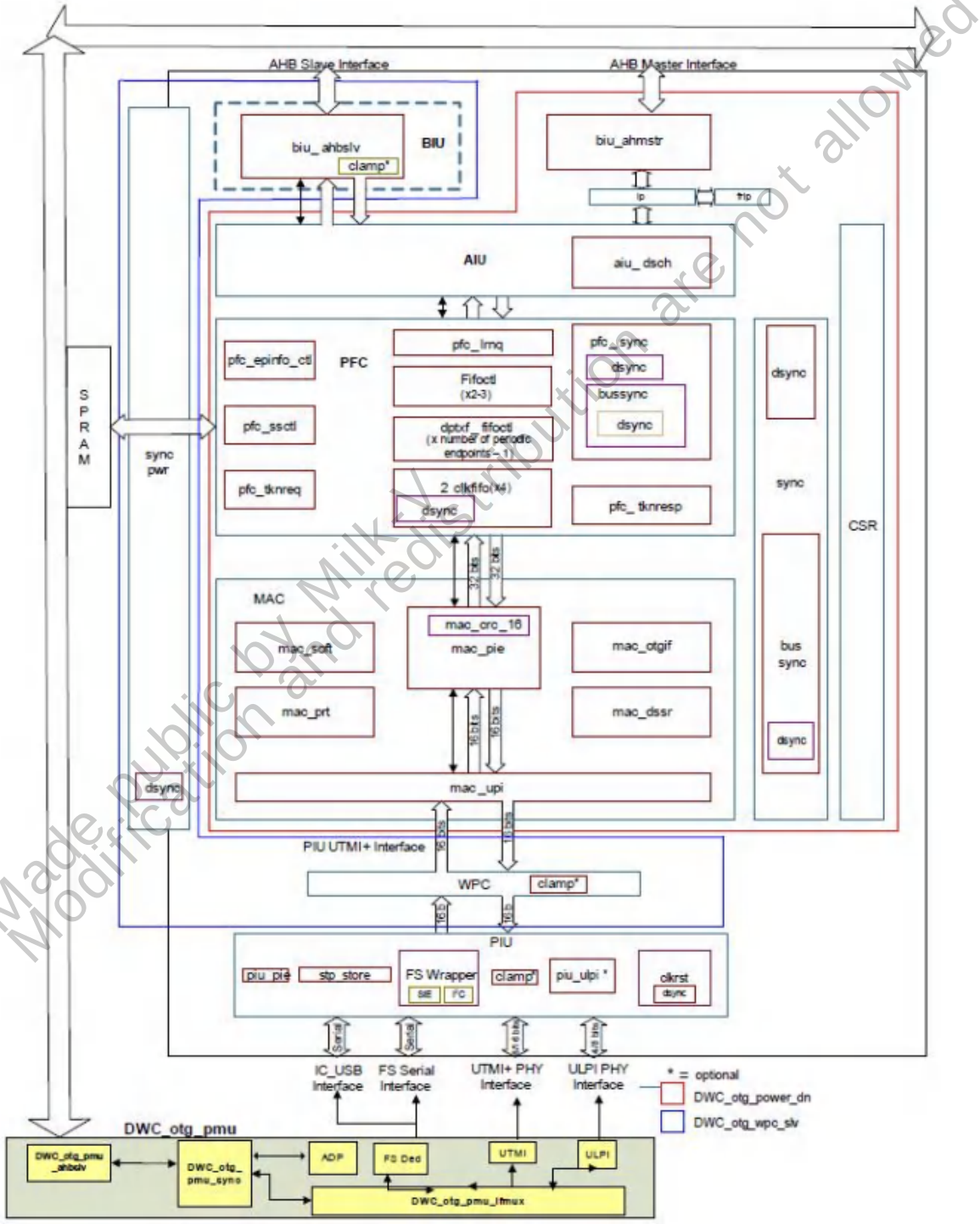
The function of USB DRD is to play the role of Hostor Device respectively, which can be changed by software setting. The transfer protocol conforms to USB 2.0 specification, and the maximum transfer rate can reach more than 40 MB/s. The software interface of Host conforms to xHCI specification, and the main operation mode of Device is scatter/gather DMA. Details will be described in the following sections. The functions of USB DRD are as follows.

- Control Transfer
- Bulk Transfer
- Isochronous Transfer
- Host can connect USB Hub and support Interrupt Transfer)
- USB DRD passed the USB electrical characteristic test (USBET), showing that the signal quality and compatibility are good

12.6.2 Function Description

12.6.2.1 System Block Diagram

The picture below shows the internal system block diagram of USB DRD.



12.6.2.2 Function Characteristics

The functions of USB DRD are as follows.

- Comply with USB2.0 transmission protocol
- Compatible with USB1.1 transmission protocol
- Support HS / FS / LS three speed modes
- Support Host or Device functions
- Support four kinds of USB transmission modes: control transfer, bulk transfer, isochronous transfer and interrupt transfer
- It can connect USB Hub and expand single interface to multiple USB interfaces
- Up to 127 device devices can be connected through USB hub extension
- Support USB2.0 suspend / resume power saving mode
- Support keyboard, mouse and other HID devices
- Device mode is mainly used for downloading and updating internal software. It can also be used for other functions, such as data transmission
- The maximum transmission rate is more than 40 MB/s

12.6.3 USBC Function and Register Description

12.6.3.1 USBC Function Description

USB DRD can switch between Host and Device functions, and can choose between one of them, However, it can't work at the same time. Its function selection and management are controlled by the USB block. In addition, there are some events and interrupt triggers on the serial bus between the host and device that also place buffers in this block.

12.6.3.2 USBC Register Abstract

Name	Address Offset	Description
GOTGCTL	0x000	Control and Status Register
GOTGINT	0x004	Interrupt Register
GAHBCFG	0x008	AHB Configuration Register
GUSBCFG	0x00c	USB Configuration Register
GRSTCTL	0x010	Reset Register
GINTSTS	0x014	Interrupt Status Register
GINTMSK	0x018	Interrupt Mask Register
GUID	0x03c	User ID Register
GLPMCFG	0x054	Core LPM Configuration Register
GPWRDN	0x058	Power Down Register

12.6.3.3 The Detailed List of USBC Registers

The memory address of USBC is 0x0434_0000, represented as USBC_BASE_ADDR in the document. To read and write to it, the real memory address in the memory space is represented as USBC_BASE_ADDR + Offset. Each register has its corresponding relative address (Offset), which is described in detail below.

GOTGCTL

Control and Status Register
Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	SesReqScs	RO	Mode: Device only Session Request Success (SesReqScs) The core sets this bit when a session request initiation is successful. ■ 1'b0: Session request failure ■ 1'b1: Session request success Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
1	SesReq	R/W	Mode: SRP-capable device Session Request (SesReq) The application sets this bit to initiate a session request on the USB. The application can clear this bit by writing a 0 when the Host Negotiation Success Status Change bit in the OTG Interrupt register (GOTGINT.HstNegSucStsChng) is SET. The core clears this bit when the HstNegSucStsChng bit is cleared. If you use the USB 1.1 Full-Speed Serial Transceiver interface to initiate the session request, the application must wait until the VBUS discharges to 0.2 V, after the B-Session Valid bit in this register (GOTGCTL.BSesVld) is cleared. This discharge time varies between different PHYs and can be obtained from the PHY vendor. ■ 1'b0: No session request ■ 1'b1: Session request Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	0x0
2	VbvalidOvEn	R/W	Mode: Host only VBUS Valid Override Enable (VbvalidOvEn) This bit is used to enable/disable the software to override the Bvalid signal using the GOTGCTL.VbvalidOvVal. ■ 1'b1: Internally Bvalid received from the PHY is overridden with GOTGCTL.VbvalidOvVal. ■ 1'b0: Override is disabled and bvalid signal from the respective PHY selected is used internally by the core. Shadow: Yes	0x0

Bits	Name	Access	Description	Reset
			Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
3	VbvalidOvVal	R/W	Mode: Host only VBUS Valid Override Value (VbvalidOvVal) This bit is used to set Override value for vbusvalid signal when GOTGCTL.VbvalidOvEn is set. ■ 1'b0: vbusvalid value is 1'b0 when GOTGCTL.VbvalidOvEn =1 ■ 1'b1: vbusvalid value is 1'b1 when GOTGCTL.VbvalidOvEn =1	0x0
4	AvalidOvEn	R/W	Mode: Host only A-Peripheral Session Valid Override Enable (AvalidOvEn) This bit is used to enable/disable the software to override the Avalid signal using the GOTGCTL.AvalidOvVal. ■ 1'b1: Internally Avalid received from the PHY is overridden with GOTGCTL.AvalidOvVal. ■ 1'b0: Override is disabled and avalid signal from the respective PHY selected is used internally by the core.	0x0
5	AvalidOvVal	R/W	Mode: Host only A-Peripheral Session Valid Override Value (AvalidOvVal) This bit is used to set Override value for Avalid signal when GOTGCTL.AvalidOvEn is set. ■ 1'b0: Avalid value is 1'b0 when GOTGCTL.AvalidOvEn =1 ■ 1'b1: Avalid value is 1'b1 when GOTGCTL.AvalidOvEn =1	0x0
6	BvalidOvEn	R/W	Mode: Device only B-Peripheral Session Valid Override Enable (BvalidOvEn) This bit is used to enable/disable the software to override the Bvalid signal using the GOTGCTL.BvalidOvVal. ■ 1'b1: Internally Bvalid received from the PHY is overridden with GOTGCTL.BvalidOvVal. ■ 1'b0: Override is disabled and bvalid signal from the respective PHY selected is used internally by the force	0x0
7	BvalidOvVal	R/W	Mode: Device only B-Peripheral Session Valid Override Value (BvalidOvVal) This bit is used to set Override value for Bvalid signal when GOTGCTL.BvalidOvEn is set. ■ 1'b0: Bvalid value is 1'b0 when GOTGCTL.BvalidOvEn =1 ■ 1'b1: Bvalid value is 1'b1 when GOTGCTL.BvalidOvEn =1	0x0
8	HstNegScs	RO	Mode: HNP-capable device Host Negotiation Success (HstNegScs) The core sets this bit when host negotiation is successful. The core clears this bit when the HNP Request (HNPReq) bit in this register is set. ■ 1'b0: Host negotiation failure ■ 1'b1: Host negotiation success	

Bits	Name	Access	Description	Reset
9	HNPReq	R/W	Mode: HNP Capable OTG Device HNP Request (HNPReq) The application sets this bit to initiate an HNP request to the connected USB host. The application can clear this bit by writing a 0 when the Host Negotiation Success Status Change bit in the OTG Interrupt register (GOTGINT.HstNegSucStsChng) is set. The core clears this bit when the HstNegSucStsChng bit is cleared. ■ 1'b0: No HNP request ■ 1'b1: HNP request	0x0
10	HstSetHNPEn	R/W	Mode: HNP Capable OTG Host Host Set HNP Enable (HstSetHNPEn) The application sets this bit when it has successfully enabled HNP (using the SetFeature.SetHNPEnable command) on the connected device. ■ 1'b0: Host Set HNP is not enabled ■ 1'b1: Host Set HNP is enabled	0x0
11	DevHNPEn	R/W	Mode: HNP Capable OTG Device Device HNP Enabled (DevHNPEn) The application sets this bit when it successfully receives aSetFeature.SetHNPEnable command from the connected USB host. ■ 1'b0: HNP is not enabled in the application ■ 1'b1: HNP is enabled in the application	0x0
12	EHEn	R/W	Embedded Host Enable It is used to select between OTG A Device state Machine andEmbedded Host state machine. ■ 1'b1: Embedded Host State Machine is selected ■ 1'b0: OTG A Device state machine is selected Note: This field is valid only in SRP-Capable OTG Mode (OTG_MODE=0,1)	0x0
14:13	Reserved_00_14_13	RO	Reserved for future use.	
15	DbnceFltrBypass	R/W	Mode: Host and Device Debounce Filter Bypass Bypass Debounce filters for avalid, bvalid, vbusvalid, sessend, and iddig signals when enabled. ■ 1'b0: Disabled ■ 1'b1: Enabled	0x0
16	ConIDSts	RO	Mode: Host and Device Connector ID Status (ConIDSts) Indicates the connector ID status on a connect event. ■ 1'b0: The DWC_otg core is in A-Device mode ■ 1'b1: The DWC_otg core is in B-Device mode	
17	DbncTime	RO	Mode: Host only Long/Short Debounce Time (DbncTime) Indicates the debounce time of a detected connection. ■ 1'b0: Long debounce time, used for physical connections (100 ms +	

Bits	Name	Access	Description	Reset
			2.5 µs) ■ 1'b1: Short debounce time, used for soft connections (2.5 µs)	
18	ASesVld	RO	Mode: Host only A-Session Valid (ASesVld) Indicates the Host mode transceiver status. ■ 1'b0: A-session is not valid ■ 1'b1: A-session is valid	
19	BSesVld	RO	Mode: Device only B-Session Valid (BSesVld) Indicates the Device mode transceiver status. ■ 1'b0: B-session is not valid. ■ 1'b1: B-session is valid. In OTG mode, you can use this bit to determine if the device is connected or disconnected.	
20	OTGVer	R/W	OTG Version (OTGVer) Indicates the OTG revision. ■ 1'b0: OTG Version 1.3. In this version the core supports Data line pulsing and VBus pulsing for SRP. ■ 1'b1: OTG Version 2.0. In this version the core supports only Data line pulsing for SRP.	0x0
21	CurMod_operation	RO	Current Mode of Operation (CurMod) Mode: Host and Device Indicates the current mode. ■ 1'b0: Device mode ■ 1'b1: Host mode	
26:22	MultValIdBC_operation	RO	Multi Valued ID pin (MultValIdBC) Mode: Host and Device Battery Charger ACA inputs in the following order: ■ Bit 26 - rid_float ■ Bit 25 - rid_gnd ■ Bit 24 - rid_a ■ Bit 23 - rid_b ■ Bit 22 - rid_c	
27	ChirpEn	RO	Chirp On Enable (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. This bit is present only if OTG_BC_SUPPORT = 1. If OTG_BC_SUPPORT!=1, this bit is a reserved bit.	
31:28	Reserved_00_31_28	RO	Reserved for future use.	

GOTGINT

Interrupt Register
Offset Address: 0x004

Bits	Name	Access	Description	Reset
1:0	Reserved_04_1_0	RO	Reserved for future use. Shadow: Yes	

Bits	Name	Access	Description	Reset
			Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
2	SesEndDet	RWC	Write Behavior: One to clear Mode: Host and Device Session End Detected (SesEndDet) The core sets this bit when the utmiotg_bvalid signal is deasserted.This bit can be set only by the core and the application should write 1 to clear it.	
7:3	Reserved_04_7_3	RO	Reserved for future use.	
8	SesReqSucStsChng	RWC	Write Behavior: One to clear Mode: Host and Device Session Request Success Status Change (SesReqSucStsChng) The core sets this bit on the success or failure of a session request. The application must read the Session Request Success bit in the OTG Control and Status register (GOTGCTL.SesReqScs) to check for success or failure.This bit can be set only by the core and the application should write 1 to clear it.	
9	HstNegSucStsChng	RWC	Write Behavior: One to clear Mode: Host and Device Host Negotiation Success Status Change (HstNegSucStsChng) The core sets this bit on the success or failure of a USB host negotiation request. The application must read the Host Negotiation Success bit of the OTG Control and Status register (GOTGCTL.HstNegScs) to check for success or failure.This bit can be set only by the core and the application should write 1 to clear it.	
16:10	Reserved_04_16_10	RO	Reserved for future use.	
17	HstNegDet	RWC	Write Behavior: One to clear Mode: Host and Device Host Negotiation Detected (HstNegDet) The core sets this bit when it detects a host negotiation request on the USB.This bit can be set only by the core and the application should write 1 to clear it.	
18	ADevTOUTChg	RWC	Write Behavior: One to clear Mode: Host and Device A-Device Timeout Change (ADevTOUTChg) The core sets this bit to indicate that the A-device has timed out while waiting for the B-device to connect.This bit can be set only by the core and the application should write 1 to clear it.	
19	DbnceDone	RWC	Write Behavior: One to clear	

Bits	Name	Access	Description	Reset
			Mode: Host only Debounce Done (DbnceDone) The core sets this bit when the debounce is completed after the device connect. The application can start driving USB reset after seeing this interrupt. This bit is only valid when the HNP Capable or SRP Capable bit is set in the Core USB Configuration register (GUSBCFG.HNPCap or GUSBCFG.SRPCap, respectively).This bit can be set only by the core and the application should write 1 to clear it.	
20	MultValIpChng	RWC	Write Behavior: One to clear This bit when set indicates that there is a change in the value of at least one ACA pin value. This bit is present only if OTG_BC_SUPPORT=1, otherwise it is reserved.	
31:21	Reserved_04_31_21	RO	Reserved for future use. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	

GAHBCFG

AHB Configuration Register
Offset Address: 0x008

Bits	Name	Access	Description	Reset
0	GlblIntrMsk	R/W	Mode: Host and device Global Interrupt Mask (GlblIntrMsk) 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 core. ■ 1'b0: Mask the interrupt assertion to the application. ■ 1'b1: Unmask the interrupt assertion to the application. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	0x0
4:1	HBstLen	R/W	Mode: Host and device Burst Length/Type (HBstLen) This field is used in both External and Internal DMA modes. In External DMA mode, these bits appear on dma_burst[3:0] ports, which can be used by an external wrapper to interface the External DMA Controller interface to Synopsys DW_ahb_dmac or ARM PrimeCell. External DMA Mode defines the DMA burst length in terms of 32-bit words: ■ 4'b0000: 1 word ■ 4'b0001: 4 words	0x0

Bits	Name	Access	Description	Reset
			■ 4'b0010: 8 words ■ 4'b0011: 16 words ■ 4'b0100: 32 words ■ 4'b0101: 64 words ■ 4'b0110: 128 words ■ 4'b0111: 256 words ■ Others: Reserved Internal DMA Mode-AHB Master burst type: ■ 4'b0000 Single ■ 4'b0001 INCR 4'b0011 ■ INCR4 4'b0101 ■ INCR8 4'b0111 ■ INCR16 ■ Others: Reserved	
5	DMAEn	R/W	Mode: Host and device DMA Enable (DMAEn) ■ 1'b0: Core operates in Slave mode ■ 1'b1: Core operates in a DMA mode This bit is always 0 when Slave-Only mode has been selected.	0x0
6	Reserved_08_6	RO	Reserved for future use. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
7	NPTxFEmpLvl	R/W	Mode: Host and device Non-Periodic TxFIFO Empty Level (NPTxFEmpLvl) This bit is used only in Slave mode. In host mode and with Shared FIFO with device mode, this bit indicates when the Non-Periodic TxFIFO Empty Interrupt bit in the Core Interrupt register (GINTSTS.NPTxFEmp) is triggered. With dedicated FIFO in device mode, this bit indicates when IN endpoint Transmit FIFO empty interrupt (DIEPINTn.TxFEmp) is triggered. Host mode and with Shared FIFO with device mode: ■ 1'b0: GINTSTS.NPTxFEmp interrupt indicates that the Non- Periodic TxFIFO is half empty ■ 1'b1: GINTSTS.NPTxFEmp interrupt indicates that the Non- Periodic TxFIFO is completely empty Dedicated FIFO in device mode: ■ 1'b0: DIEPINTn.TxFEmp interrupt indicates that the IN Endpoint TxFIFO is half empty ■ 1'b1: DIEPINTn.TxFEmp interrupt indicates that the IN Endpoint TxFIFO is completely empty	0x0
8	PTxFEmpLvl	R/W	Mode: Host only Periodic TxFIFO Empty Level (PTxFEmpLvl) Indicates when the Periodic TxFIFO Empty Interrupt bit in the Core	0x0

Bits	Name	Access	Description	Reset
			Interrupt register (GINTSTS.PTxFEmp) is triggered. This bit is used only in Slave mode. ■ 1'b0: GINTSTS.PTxFEmp interrupt indicates that the Periodic TxFIFO is half empty ■ 1'b1: GINTSTS.PTxFEmp interrupt indicates that the Periodic TxFIFO is completely empty	
20:9	Reserved_08_20_9	RO	Reserved for future use. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
21	RemMemSupp	R/W	Mode: Host and Device Remote Memory Support (RemMemSupp) This bit is programmed to enable the functionality to wait for the system DMA Done Signal for the DMA Write Transfers. ■ GAHBCFG.RemMemSupp=1 The int_dma_req output signal is asserted when HSOTG DMA starts write transfer to the external memory. When the core is done with the Transfers it asserts int_dma_done signal to flag the completion of DMA writes from HSOTG. The core then waits for sys_dma_done signal from the system to proceed further and complete the Data Transfer corresponding to a particular Channel/Endpoint. ■ GAHBCFG.RemMemSupp=0 The int_dma_req and int_dma_done signals are not asserted and the core proceeds with the assertion of the XferComp interrupt as soon as the DMA write transfer is done at the HSOTG Core Boundary and it doesn't wait for the sys_dma_done signal to complete the DATA transfers.	0x0
22	NotiAllDmaWrit	R/W	Mode: Host and Device Notify all DMA Write Transactions (NotiAllDmaWrit) This bit is programmed to enable the System DMA Done functionality for all the DMA write Transactions corresponding to the Channel/Endpoint. This bit is valid only when GAHBCFG.RemMemSupp is set to 1. ■ GAHBCFG.NotiAllDmaWrit = 1 DWC_otg core asserts int_dma_req for all the DMA write transactions on the AHB interface along with int_dma_done, chep_last_transact and chep_number signal informations. The core waits for sys_dma_done signal for all the DMA write transactions in order to complete the transfer of a particular Channel/Endpoint.	0x0

Bits	Name	Access	Description	Reset
			■ GAHBCFG.NotiAllDmaWrit = 0 DWC_otg core asserts int_dma_req signal only for the last transaction of DMA write transfer corresponding to a particular Channel/Endpoint. Similarly, the core waits for sys_dma_done signal only for that transaction of DMA write to complete the transfer of a particular Channel/Endpoint.	
23	AHBSingle	R/W	Mode: Host and Device AHBSingleSupport (AHBSingle) This bit when programmed supports Single transfers for the remaining data in a transfer when the DWC_otg core is operating in DMA mode. ■ 1'b0: This is the default mode. When this bit is set to 1'b0, the remaining data in the transfer is sent using INCR burst size. ■ 1'b1: When set to 1'b1, the remaining data in a 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 DWC_otg core does not support INCR burst (and when Split, and Retry transactions are not being used in the bus.)	0x0
24	InvDescEndianness	R/W	Mode: Host and Device Inverse Descriptor Endianness ■ 1'b0: Descriptor endianness is similar to the 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.	0x0
31:25	Reserved_08_31_25	RO	Reserved for future use. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	

GUSBCFG

USB Configuration Register
Offset Address: 0x00c

Bits	Name	Access	Description	Reset
2:0	TOutCal	R/W	Mode: Host and Device HS/FS Timeout Calibration (TOutCal) 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.	0x0

Bits	Name	Access	Description	Reset
			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 fullspeed 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: High-speed operation: ■ One 30-MHz PHY clock = 16 bit times ■ One 60-MHz PHY clock = 8 bit times Full-speed operation: ■ One 30-MHz PHY clock = 0.4 bit times ■ One 60-MHz PHY clock = 0.2 bit times ■ One 48-MHz PHY clock = 0.25 bit times Using the HS as an example, if you set ToutCal to '001' you add one 30MHz PHY clock or 16 bit times. If you set ToutCal to '010' you add two 30MHz PHY clocks or 32 bit times, and so on. The 3 bits allow you to add up to 7 PHY clocks, and the number of bit times depend on the speed, and the PHY clock you are using. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
3	PHYIf	RO	Mode: Host and Device PHY Interface (PHYIf) The application uses this bit to configure the core to support a UTMI+ PHY with an 8- or 16-bit interface. When a ULPI PHY is chosen, this must be set to 8-bit mode. ■ 1'b0: 8 bits ■ 1'b1: 16 bits This bit is writable only If UTMI+ and ULPI were selected. Otherwise, this bit returns the value for the power-on interface selected during configuration. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
4	ULPI_UTMI_Sel	R/W	Mode: Host and Device ULPI or UTMI+ Select (ULPI_UTMI_Sel) The application uses this bit to select either a UTMI+ interface or ULPI Interface. ■ 1'b0: UTMI+ Interface ■ 1'b1: ULPI Interface This bit is writable only If UTMI+ and ULPI was specified for High- Speed PHY Interface(s).	0x0

Bits	Name	Access	Description	Reset
			Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
5	FSIntf	R/W	Mode: Host and Device Full-Speed Serial Interface Select (FSIntf) The application uses this bit to select either a unidirectional or bidirectional USB 1.1 full-speed serial transceiver interface. ■ 1'b0: 6-pin unidirectional full-speed serial interface ■ 1'b1: 3-pin bidirectional full-speed serial interface If a USB 1.1 Full-Speed Serial Transceiver interface was not selected, this bit is always 0, with Read Only access. If a USB 1.1 FS interface was selected, then the application can set this bit to select between the 3- and 6-pin interfaces, and access is Read and Write.	0x0
6	PHYSel	R/W	Mode: Host and Device USB 2.0 High-Speed PHY or USB 1.1 Full-Speed Serial Transceiver Select (PHYSel) The application uses this bit to select either a high-speed UTMI+ or ULPI PHY, or a full-speed transceiver. ■ 1'b0: USB 2.0 high-speed UTMI+ or ULPI PHY ■ 1'b1: USB 1.1 full-speed serial transceiver If a USB 1.1 Full-Speed Serial Transceiver interface was not selected, this bit is always 0, with Read Only access. If a high-speed PHY interface was not selected, this bit is always 1, with Read Only access. If both interface types were selected (parameters have non-zero values), the application uses this bit to select which interface is active, and access is Read and Write.	0x0
7	DDRSel	R/W	Mode: Host and Device ULPI DDR Select (DDRSel) The application uses this bit to select a Single Data Rate (SDR) or Double Data Rate (DDR) or ULPI interface. ■ 1'b0: Single Data Rate ULPI Interface, with 8-bit-wide data bus ■ 1'b1: Double Data Rate ULPI Interface, with 4-bit-wide data bus This bit is valid only when OTG_HSPHY_INTERFACE = 2 or 3.	0x0
8	SRPCap	R/W	Mode: Host and Device SRP-Capable (SRPCap) The application uses this bit to control the DWC_otg core SRP capabilities. If the core operates as a non-SRP-capable B-device, it cannot request the connected A-device (host) to activate VBUS and start a session. ■ 1'b0: SRP capability is not enabled. ■ 1'b1: SRP capability is enabled. This bit is writable only if an SRP mode was specified for Mode	0x0

Bits	Name	Access	Description	Reset
			of Operation in coreConsultant (parameter OTG_MODE). Otherwise, reads return 0. If SRP functionality is disabled by the software, the OTG signals on the PHY domain must be tied to the appropriate values.	
9	HNPCap	R/W	Mode: Host and Device HNP-Capable (HNPCap) The application uses this bit to control the DWC_otg core's HNP capabilities. ■ 1'b0: HNP capability is not enabled. ■ 1'b1: HNP capability is enabled. This bit is writable only if an HNP mode was specified for Mode of Operation in coreConsultant (parameter OTG_MODE). Otherwise, reads return 0. If HNP functionality is disabled by the software, the OTG signals on the PHY domain must be tied to the appropriate values.	0x0
13:10	USBTrdTim	R/W	Mode: Device only USB Turnaround Time (USBTrdTim) 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 (SPRAM). This must be programmed to ■ 4'h5: When the MAC interface is 16-bit UTMI+. ■ 4'h9: When the MAC interface is 8-bit UTMI+. Note: The values above are calculated for the minimum AHB frequency of 30 MHz. USB turnaround time is critical for certification where long cables and 5-Hubs are used, so If you need the AHB to run at less than 30 MHz, and If USB turnaround time is not critical, these bits can be programmed to a larger value.	0x5
14	Reserved_0C_14	RO	Reserved for future use.	
15	PhyLPwrClkSel	R/W	Mode: Host and Device PHY Low-Power Clock Select (PhyLPwrClkSel) Selects either 480-MHz or 48-MHz (low-power) PHY mode. In FS and LS modes, the PHY can usually operate on a 48-MHz clock to save power. ■ 1'b0: 480-MHz Internal PLL clock ■ 1'b1: 48-MHz External Clock In 480 MHz mode, the UTMI interface operates at either 60 or 30- MHz, depending upon whether 8- or 16-bit data width is selected. In 48-MHz mode, the UTMI interface operates at 48 MHz in FS mode and at either 48 or 6 MHz in LS mode (depending on the PHY	0x0

Bits	Name	Access	Description	Reset
			vendor). This bit drives the utmi_fsls_low_power core output signal, and is valid only For UTMI+ PHYs.	
16	OtgI2CSel	R/W	Mode: Host and Device UTMIFS or I2C Interface Select (OtgI2CSel) The application uses this bit to select the I2C interface. ■ 1'b0: UTMI USB 1.1 Full-Speed interface for OTG signals ■ 1'b1: I2C interface for OTG signals This bit is writable only if I2C and UTMIFS were specified for Enable I2C Interface? in coreConsultant (parameter OTG_I2C_INTERFACE = 2). Otherwise, reads return 0.	0x0
17	ULPIFsLs	R/W	Mode: Host and Device ULPI FS/LS Select (ULPIFsLs) The application uses this bit to select the FS/LS serial interface for the ULPI PHY. This bit is valid only when the FS serial transceiver is selected on the ULPI PHY. ■ 1'b0: ULPI interface ■ 1'b1: ULPI FS/LS serial interface (Valid only when RTL parameters OTG_HSPHY_INTERFACE = 2 or 3 and OTG_FSPHY_INTERFACE = 1, 2, or 3) Before setting this bit, the application needs to ensure that GUSBCFG.ULPI_UTMI_SEL = 1'b1.	0x0
18	ULPIAutoRes	R/W	Mode: Host and Device ULPI Auto Resume (ULPIAutoRes) This bit sets the AutoResume bit in the Interface Control register on the ULPI PHY. ■ 1'b0: PHY does not use AutoResume feature. ■ 1'b1: PHY uses AutoResume feature. (Valid only when RTL parameter OTG_HSPHY_INTERFACE = 2 or 3)	0x0
19	ULPIClkSusM	R/W	Mode: Host and Device ULPI Clock SuspendM (ULPIClkSusM) This bit sets the ClockSuspendM bit in the Interface Control register on the ULPI PHY. This bit applies only in serial or carkit modes. ■ 1'b0: PHY powers down internal clock during suspend. ■ 1'b1: PHY does not power down internal clock. (Valid only when RTL parameter OTG_HSPHY_INTERFACE = 2 or 3)	0x0
20	ULPIExtVbusDrv	R/W	Mode: Host only ULPI External VBUS Drive (ULPIExtVbusDrv) This bit selects between internal or external supply to drive 5V on VBUS, in ULPI PHY. ■ 1'b0: PHY drives VBUS using internal charge pump (Default). ■ 1'b1: PHY drives VBUS using external supply. (Valid only when RTL parameter OTG_HSPHY_INTERFACE = 2 or 3)	0x0
21	ULPIExtVbusInd	R/W	Mode: Host only	0x0

Bits	Name	Access	Description	Reset
	icator		ULPI External VBUS Indicator (ULPIExtVbusIndicator) This bit indicates to the ULPI PHY to use an external VBUS overcurrent indicator. ■ 1'b0: PHY uses internal VBUS valid comparator. ■ 1'b1: PHY uses external VBUS valid comparator. (Valid only when RTL parameter OTG_HSPHY_INTERFACE = 2 or 3)	
22	TermSelDLPulse	R/W	Mode: Device only TermSel DLine Pulsing Selection (TermSelDLPulse) This bit selects utmi_termselect to drive data line pulse during SRP. ■ 1'b0: Data line pulsing using utmi_txvalid (Default). ■ 1'b1: Data line pulsing using utmi_termsel.	0x0
23	Complement	R/W	Mode: Host only Indicator Complement Controls the PHY to invert the ExternalVbusIndicator input signal, generating the Complement Output. For more information, refer to the ULPI Specification. ■ 1'b0: PHY does not invert ExternalVbusIndicator signal ■ 1'b1: PHY does invert ExternalVbusIndicator signal This bit is reserved and read-only when OTG_HSPHY_INTERFACE is set to 0 or 1.	0x0
24	Indicator	R/W	Mode: Host only Indicator Pass Through Controls whether the Complement Output is qualified with the Internal Vbus Valid comparator before being used in the Vbus State in the RX CMD. For more information, refer to the ULPI Specification. ■ 1'b0: Complement Output signal is qualified with the Internal VbusValid comparator. ■ 1'b1: Complement Output signal is not qualified with the Internal VbusValid comparator. This bit is reserved and read-only when OTG_HSPHY_INTERFACE is set to 0 or 1.	0x0
25	ULPI	R/W	Mode: Host only ULPI Interface Protect Disable Controls circuitry built into the PHY For protecting the ULPI interface when the link tri-states STP and data. Any pull-ups or pull-downs employed by this feature can be disabled. For more information, refer to the ULPI Specification. ■ 1'b0: Enables the interface protect circuit ■ 1'b1: Disables the interface protect circuit This bit is reserved and read-only when OTG_HSPHY_INTERFACE is set to 0 or 1.	0x0
26	IC_USBCap	RO	Mode: Host and Device	

Bits	Name	Access	Description	Reset
			IC_USB-Capable (IC_USBCap) The application uses this bit to control the DWC_otg core's IC_USB capabilities. ■ 1'b0: IC_USB PHY Interface is not selected. ■ 1'b1: IC_USB PHY Interface is selected. This bit is writable only if OTG_ENABLE_IC_USB=1 and OTG_FSPHY_INTERFACE!=0. The reset value depends on the configuration parameter OTG_SELECT_IC_USB when OTG_ENABLE_IC_USB = 1. In all other cases, this bit is set to 1'b0 and the bit is read only.	
27	IC_USBTrafCtl	R/W	Mode: Device only IC_USB TrafficPullRemove Control (IC_USBTrafCtl) When this bit is set, pullup/pulldown resistors are detached from the USB during traffic signaling, per section 6.3.4 of the IC_USB specification. This bit is valid only when configuration parameter OTG_ENABLE_IC_USB = 1 and register field USBCFG.IC_USBCap is set to 1.	0x0
28	TxEndDelay	R/W	Mode: Device only Tx End Delay (TxEndDelay) Writing 1'b1 to this bit enables the core 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.	0x0
29	ForceHstMode	R/W	Mode: Host and device Force Host Mode (ForceHstMode) Writing a 1 to this bit forces the core to host mode irrespective of utmiotg_iddig input pin. ■ 1'b0: Normal Mode. ■ 1'b1: Force Host Mode. After setting the force bit, the application must wait at least 25 ms before the change to take effect. When the simulation is in scale down mode, waiting for 500 µs is sufficient. This bit is valid only when OTG_MODE = 0, 1 or 2. In all other cases, this bit reads 0.	0x0
30	ForceDevMode	R/W	Mode: Host and device Force Device Mode (ForceDevMode) Writing a 1 to this bit forces the core to device mode irrespective of utmiotg_iddig input pin. ■ 1'b0: Normal Mode. ■ 1'b1: Force Device Mode. After setting the force bit, the application must wait at least 25 ms before the change to take effect. When the simulation is in scale down mode, waiting for 500 µs is sufficient. This bit is valid	0x0

Bits	Name	Access	Description	Reset
			only when OTG_MODE = 0, 1 or 2. In all other cases, this bit reads 0.	
31	CorruptTxPkt	R/W	Mode: Host and device Corrupt Tx packet (CorruptTxPkt) This bit is for debug purposes only. Never set this bit to 1.The application should always write 1'b0 to this bit.	0x0

GRSTCTL

Reset Register

Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	CSftRst	RO	Write Behavior: One to set Mode: Host and Device Core Soft Reset (CSftRst) Resets the hclk and phy_clock domains as follows: ■ Clears the interrupts and all the CSR registers except the following register bits: - PCGCCTL.RstPdwnModule - PCGCCTL.GateHclk - PCGCCTL.PwrClmp - PCGCCTL.StopPPhyLPwrClkSelclk - GUSBCFG.PhyLPwrClkSel - GUSBCFG.DDRSel - GUSBCFG.PHYSel - GUSBCFG.FSIntf - GUSBCFG.ULPI_UTMI_Sel - GUSBCFG.PHYIf - GUSBCFG.TxEndDelay - GUSBCFG.TermSelDLPulse - GUSBCFG.ULPIClkSusM - GUSBCFG.ULPIAutoRes - GUSBCFG.ULPIFsLs - GGPIO - GPWRDN - GADPCTL - HCFG.FSLSPclkSel - DCFG.DevSpd - DCTL.SftDiscon ■ 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. ■ When Hibernation or ADP feature is enabled, the PMU module is not reset by the Core Soft Reset. 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	

Bits	Name	Access	Description	Reset
			in the core, which can take several clocks, depending on the current state of the core. After this bit is cleared, the application must wait at least 3 PHY clocks before doing any access to the PHY domain (synchronization delay). The application 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. After a new clock is selected, the PHY domain has to be reset for proper operation. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
1	PIUFSSftRst	RO	Write Behavior: One to set Mode: Host and Device PIU FS Dedicated Controller Soft Reset (PIUFSSftRst) 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. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
2	FrmCntrRst	RO	Write Behavior: One to set Mode: Host only Host Frame Counter Reset (FrmCntrRst) The application writes this bit to reset the (micro)frame number counter inside the core. When the (micro)frame counter is reset, the subsequent SOF sent out by the core has a (micro)frame number of 0. If the application writes 1 to the bit, it may not be able to read back the value as it gets cleared by the core in a few clock cycles. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
3	INTknQFlsh	RWS	Mode: Device only IN Token Sequence Learning Queue Flush (INTknQFlsh) This bit is valid only if OTG_EN_DED_TX_FIFO = 0.	

Bits	Name	Access	Description	Reset
			The application writes this bit to flush the IN Token Sequence Learning Queue.	
4	RxFFlsh	RO	Write Behavior: One to set Mode: Host and Device RxFIFO Flush (RxFFlsh) 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 core 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 8 clocks (slowest of PHY or AHB clock) to clear.	
5	TxFFlsh	RWS	Write Behavior: One to set Mode: Host and Device TxFIFO Flush (TxFFlsh) 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: ■ Read - NAK Effective Interrupt ensures the core is not reading from the FIFO ■ Write - GRSTCTL.AHBIdle ensures the core is not writing anything to the FIFO. Flushing is normally recommended when FIFOs are reconfigured or when switching between Shared FIFO and Dedicated Transmit FIFO operation. 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.	
10:6	TxFNum	R/W	Mode: Host and Device TxFIFO Number (TxFNum) 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: - Non-periodic TxFIFO flush in Host mode - Non-periodic TxFIFO flush in device mode when in shared FIFO operation - Tx FIFO 0 flush in device mode when in dedicated FIFO mode ■ 5'h1:	0x0

Bits	Name	Access	Description	Reset
			- Periodic TxFIFO flush in Host mode - Periodic TxFIFO 1 flush in Device mode when in shared FIFO operation - TXFIFO 1 flush in device mode when in dedicated FIFO mode ■ 5'h2: - Periodic TxFIFO 2 flush in Device mode when in shared FIFO operation - TXFIFO 2 flush in device mode when in dedicated FIFO mode ... ■ 5'hF: - Periodic TxFIFO 15 flush in Device mode when in shared FIFO operation - TXFIFO 15 flush in device mode when in dedicated FIFO mode ■ 5'h10: - Flush all the transmit FIFOs in device or host mode.	
29:11	Reserved_10_29_11	RO	Reserved for future use.	
30	DMAReq	RO	Mode: Host and Device DMA Request Signal (DMAReq) Indicates that the DMA request is in progress. Used for debug.	
31	AHBIdle	RO	Mode: Host and Device AHB Master Idle (AHBIdle) Indicates that the AHB Master State Machine is in the IDLE condition.	

GINTSTS

Interrupt Status Register
Offset Address: 0x014

Bits	Name	Access	Description	Reset
0	CurMod	RO	Mode: Host and Device Current Mode of Operation (CurMod) Indicates the current mode. ■ 1'b0: Device mode ■ 1'b1: Host mode Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
1	ModeMis	RWC	Write Behavior: One to clear Mode: Host and Device Mode Mismatch Interrupt (ModeMis) The core sets this bit when the application is trying to access: ■ A Host mode register, when the core is operating in Device mode ■ A Device mode register, when the core is operating in Host mode The register access is completed on the AHB with an OKAY response, but is ignored by the core internally and does not affect the operation of the core. This bit can be set only by the core and the application should write 1 to clear it. Shadow: Yes	

Bits	Name	Access	Description	Reset
			Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
2	OTGInt	RO	Mode: Host and Device OTG Interrupt (OTGInt) The core sets this bit to indicate an OTG protocol event. The application must read the OTG Interrupt Status (GOTGINT) register to determine the exact event that caused this interrupt. The application must clear the appropriate status bit in the GOTGINT register to clear this bit. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
3	Sof	RWC	Write Behavior: One to clear Mode: Host and Device Start of (micro)Frame (Sof) In Host mode, the core sets this bit to indicate that an SOF (FS), micro-SOF (HS), or Keep-Alive (LS) is transmitted on the USB. The application must write a 1 to this bit to clear the interrupt. In Device mode, in the core 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: The 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 sent (in host mode), or SOF has been received (in device mode). 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.	
4	RxFLvl	RO	Mode: Host and Device RxFIFO Non-Empty (RxFLvl) Indicates that there is at least one packet pending to be read from the RxFIFO.	
5	NPTxFEmp	RO	Mode: Host and Device Non-periodic TxFIFO Empty (NPTxFEmp) This interrupt is asserted when the Non-periodic TxFIFO is either half or completely empty, and there is space for at least one entry to be	

Bits	Name	Access	Description	Reset
			written to the Non-periodic Transmit Request Queue. The half or completely empty status is determined by the Non-periodic TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.NPTxFEmpLvl). In host mode, the application can use GINTSTS.NPTxFEmp with the OTG_EN_DED_TX_FIFO parameter set to either 1 or 0. In device mode, the application uses GINTSTS.NPTxFEmp when OTG_EN_DED_TX_FIFO=0. When OTG_EN_DED_TX_FIFO=1, the application uses DIEPINTn.TxFEmp.	
6	GINNakEff	RO	Mode: Device only Global IN Non-periodic NAK Effective (GINNakEff) 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.	
7	GOUTNakEff	RO	Mode: Device only Global OUT NAK Effective (GOUTNakEff) 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).	
8	ULPICKINT_I2CCKINT	RWC	Write Behavior: One to clear Mode: Host and Device ULPI Carkit Interrupt (ULPICKINT) The core sets this interrupt when a ULPI Carkit interrupt is received. The core's PHY sets ULPI Carkit interrupt in UART or Audio mode. This field is used only if the Carkit interface was enabled in coreConsultant (parameter OTG_ULPI_CARKIT = 1). Otherwise, reads return 0. I2C Carkit Interrupt (I2CCKINT) The core sets this interrupt when a Carkit interrupt is received. The core's PHY sets the I2C Carkit interrupt in Audio mode. This field is used only if the I2C interface was enabled in coreConsultant (parameter OTG_I2C_INTERFACE = 1). Otherwise, reads return 0.	
9	I2CINT	RWC	Write Behavior: One to clear Mode: Host and Device	

Bits	Name	Access	Description	Reset
			I2C Interrupt (I2CINT) The core sets this interrupt when I2C access is completed on the I2C interface. This field is used only if the I2C interface was enabled in coreConsultant (parameter OTG_I2C_INTERFACE = 1). Otherwise, reads return 0.	
10	ErlySusp	RWC	Write Behavior: One to clear Mode: Device only Early Suspend (ErlySusp) The core sets this bit to indicate that an Idle state has been detected on the USB For 3 ms.	
11	USBSusp	RWC	Write Behavior: One to clear Mode: Device only USB Suspend (USBSusp) The core sets this bit to indicate that a suspend was detected on the USB. The core enters the Suspend state when there is no activity on the linestate signal for an extended period of time.	
12	USBRst	RWC	Write Behavior: One to clear Mode: Device only USB Reset (USBRst) The core sets this bit to indicate that a reset is detected on the USB.	
13	EnumDone	RWC	Write Behavior: One to clear Mode: Device only Enumeration Done (EnumDone) 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.	
14	ISOOutDrop	RWC	Write Behavior: One to clear Mode: Device only Isochronous OUT Packet Dropped Interrupt (ISOOutDrop) The core 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.	
15	EOPF	RWC	Write Behavior: One to clear Mode: Device only End of Periodic Frame Interrupt (EOPF) 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.	
16	RstrDoneInt	RWC	Mode: Host and Device Restore Done Interrupt (RstrDoneInt) The core sets this bit to indicate that the Restore command after Hibernation was completed by the core.	

Bits	Name	Access	Description	Reset
			The core continues from Suspend state into the mode dictated by the PCGCCTL.RestoreMode field. This bit is valid only when Hibernation feature is enabled (OTG_EN_PWRPOPT=2).	
17	EPMis	RO	Write Behavior: One to clear Mode: Device only Endpoint Mismatch Interrupt (EPMis) Note: This interrupt is valid only in shared FIFO operation. Indicates that an IN token has been received for a non-periodic endpoint, but the data for another endpoint is present in the top of the Non-periodic Transmit FIFO and the IN endpoint mismatch count programmed by the application has expired.	
18	IEPInt	RO	Mode: Device only IN Endpoints Interrupt (IEPInt) The core sets this bit to indicate that an interrupt is pending on one of the IN endpoints of the core (in Device mode). The application must read the Device All Endpoints Interrupt (DAINT) register to determine the exact number of the IN endpoint on 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.	
19	OEPInt	RO	Mode: Device only OUT Endpoints Interrupt (OEPInt) The core sets this bit to indicate that an interrupt is pending on one of the OUT endpoints of the core (in Device mode). The application must read the Device All Endpoints Interrupt (DAINT) register to determine the exact number of the OUT endpoint on which the interrupt occurred, and Then read the corresponding Device OUT Endpoint-n Interrupt (DOEPINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the corresponding DOEPINTn register to clear this bit.	
20	incompISOIN	RWC	Write Behavior: One to clear Mode: Device only Incomplete Isochronous IN Transfer (incompISOIN) The core sets this interrupt to indicate that there is at least one isochronous IN endpoint on which the transfer is not completed in the current microframe. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register. Note: This interrupt is not asserted in Scatter/Gather DMA	

Bits	Name	Access	Description	Reset
			mode.	
21	incomplP_incompISOOUT	RWC	Write Behavior: One to clear Incomplete Periodic Transfer (incomplP) Mode: Host only In Host mode, the core sets this interrupt bit when there are incomplete periodic transactions still pending which are scheduled for the current microframe. Incomplete Isochronous OUT Transfer (incompISOOUT) Mode: Device only In Device mode, the core sets this interrupt to indicate that there is at least one isochronous OUT endpoint on which the transfer is not completed in the current microframe. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register.	
22	FetSusp	RWC	Write Behavior: One to clear Mode: Device only Data Fetch Suspended (FetSusp) This interrupt is valid only in DMA mode. This interrupt indicates that the core has stopped fetching data For IN endpoints due to the unavailability of 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. DWC_otg 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.	
23	ResetDet	RWC	Write Behavior: One to clear Mode: Device only Reset detected Interrupt (ResetDet)	

Bits	Name	Access	Description	Reset
			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. In Host mode, this interrupt is not asserted.	
24	PrtInt	RO	Mode: Host only Host Port Interrupt (PrtInt) The core sets this bit to indicate a change in port status of one of the DWC_otg core ports in Host mode. The application must read the Host Port Control and Status (HPRT) register to determine the exact event that caused this interrupt. The application must clear the appropriate status bit in the Host Port Control and Status register to clear this bit.	
25	HChInt	RO	Mode: Host only Host Channels Interrupt (HChInt) The core sets this bit to indicate that an interrupt is pending on one of the channels of the core (in Host mode). The application must read the Host All Channels Interrupt (HAINT) register to determine the exact number of the channel on which the interrupt occurred and then read the corresponding Host Channel-n Interrupt (HCINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the HCINTn register to clear this bit.	
26	PTxFEmp	RO	Mode: Host only Periodic TxFIFO Empty (PTxFEmp) This interrupt is asserted when the Periodic Transmit FIFO is either half or completely empty and there is space for at least one entry to be written in the Periodic Request Queue. The half or completely empty status is determined by the Periodic TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.PTxFEmpLvl).	
27	LPM_Int	RWC	Write Behavior: One to clear Mode: Host and Device LPM Transaction Received Interrupt (LPM_Int) ■ Device Mode - This interrupt is asserted when the device receives an LPM transaction and responds with a non-ERRORed response. ■ Host Mode - This interrupt is asserted when the device responds to an LPM transaction with a non-ERRORed response or when the host core has completed LPM transactions for the	

Bits	Name	Access	Description	Reset
			programmed number of times (GLPMCFG.RetryCnt).	
28	ConIDStsChng	RWC	Write Behavior: One to clear Mode: Host and Device Connector ID Status Change (ConIDStsChng) The core sets this bit when there is a change in connector ID status.	
29	DisconnInt	RWC	Write Behavior: One to clear Mode: Host only Disconnect Detected Interrupt (DisconnInt) Asserted when a device disconnect is detected.	
30	SessReqInt	RWC	Write Behavior: One to clear Mode: Host and Device Session Request/New Session Detected Interrupt (SessReqInt) In Host mode, this interrupt is asserted when a session request is detected from the device. In Device mode, this interrupt is asserted when the utmisrp_bvalid signal goes high.	
31	WkUpInt	RWC	Write Behavior: One to clear Mode: Host and Device Resume/Remote Wakeup Detected Interrupt (WkUpInt) Wakeup Interrupt during Suspend(L2) or LPM(L1) state. ■ During Suspend (L2): - Device Mode - This interrupt is asserted only when Host Initiated Resume is detected on USB. - Host Mode - This interrupt is asserted only when Device Initiated Remote Wakeup is detected on USB. ■ During LPM (L1): - Device Mode - This interrupt is asserted for either Host Initiated Resume or Device Initiated Remote Wakeup on USB. - Host Mode - This interrupt is asserted for either Host Initiated Resume or Device Initiated Remote Wakeup on USB.	

GINTMSK

Interrupt Mask Register
Offset Address: 0x018

Bits	Name	Access	Description	Reset
0	Reserved_18_0	RO	Reserved for future use. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
1	ModeMisMsk	R/W	Mode: Host and Device Mode Mismatch Interrupt Mask (ModeMisMsk) Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	0x0
2	OTGIntMsk	R/W	Mode: Host and Device OTG Interrupt Mask (OTGIntMsk) Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	0x0

Bits	Name	Access	Description	Reset
			One-Way: Enabled	
3	SofMsk	R/W	Mode: Host and Device Start of (micro)Frame Mask (SofMsk)	0x0
4	RxFLvlMsk	R/W	Mode: Host and Device Receive FIFO Non-Empty Mask (RxFLvlMsk)	0x0
5	NPTxFEmpMsk	R/W	Mode: Host and Device Non-periodic TxFIFO Empty Mask (NPTxFEmpMsk)	0x0
6	GINNakEffMsk	R/W	Mode: Device only Global Non-periodic IN NAK Effective Mask (GINNakEffMsk)	0x0
7	GOUTNakEffMsk	R/W	Mode: Device only Global OUT NAK Effective Mask (GOUTNakEffMsk)	0x0
8	ULPICKINTMsk_I2CCKINTMsk	R/W	ULPI Carkit Interrupt Mask (ULPICKINTMsk) Mode: Host and Device I2C Carkit Interrupt Mask (I2CCKINTMsk) Mode: Host and Device	0x0
9	I2CIntMsk	R/W	Mode: Host and Device I2C Interrupt Mask (I2CIntMsk)	0x0
10	ErlySuspMsk	R/W	Mode: Device only Early Suspend Mask (ErlySuspMsk)	0x0
11	USBSuspMsk	R/W	Mode: Device only USB Suspend Mask (USBSuspMsk)	0x0
12	USBRstMsk	R/W	Mode: Device only USB Reset Mask (USBRstMsk)	0x0
13	EnumDoneMsk	R/W	Mode: Device only Enumeration Done Mask (EnumDoneMsk)	0x0
14	ISOOutDropMsk	R/W	Mode: Device only Isochronous OUT Packet Dropped Interrupt Mask (ISOOutDropMsk)	0x0
15	EOPFMsk	R/W	Mode: Device only End of Periodic Frame Interrupt Mask (EOPFMsk)	0x0
16	RstrDoneIntMsk	R/W	Mode: Host and Device Restore Done Interrupt Mask (RstrDoneIntMsk) This field is valid only when Hibernation feature is enabled (OTG_EN_PWROPT=2).	0x0
17	EPMisMsk	R/W	Mode: Device only Endpoint Mismatch Interrupt Mask (EPMisMsk)	0x0
18	IEPIntMsk	R/W	Mode: Device only IN Endpoints Interrupt Mask (IEPIntMsk)	0x0
19	OEPIntMsk	R/W	Mode: Device only OUT Endpoints Interrupt Mask (OEPIntMsk)	0x0
20	incomplSOINMsk	R/W	Mode: Device only Incomplete Isochronous IN Transfer Mask (incomplSOINMsk) This bit is enabled only when device periodic endpoints are enabled in Dedicated TxFIFO mode.	0x0
21	incomplPMsk_incompISOOUTMsk	R/W	Incomplete Periodic Transfer Mask (incomplPMsk) Mode: Host only Incomplete Isochronous OUT Transfer Mask (incomplSOOUTMsk) Mode: Device only	0x0
22	FetSuspMsk	R/W	Mode: Device only Data Fetch Suspended Mask (FetSuspMsk)	0x0
23	ResetDetMsk	R/W	Mode: Device only	0x0

Bits	Name	Access	Description	Reset
			Reset detected Interrupt Mask (ResetDetMsk)	
24	PrtIntMsk	R/W	Mode: Host only Host Port Interrupt Mask (PrtIntMsk)	0x0
25	HChIntMsk	R/W	Mode: Host only Host Channels Interrupt Mask (HChIntMsk)	0x0
26	PTxFEmpMsk	R/W	Mode: Host only Periodic TxFIFO Empty Mask (PTxFEmpMsk)	0x0
27	LPM_IntMsk	R/W	Mode: Host and Device LPM Transaction received interrupt Mask (LPM_IntMsk)	0x0
28	ConIDStsChngMsk	R/W	Mode: Host and Device Connector ID Status Change Mask (ConIDStsChngMsk)	0x0
29	DisconnIntMsk	R/W	Mode: Host and Device Disconnect Detected Interrupt Mask (DisconnIntMsk)	0x0
30	SessReqIntMsk	R/W	Mode: Host and Device Session Request/New Session Detected Interrupt Mask (SessReqIntMsk)	0x0
31	WkUpIntMsk	R/W	Mode: Host and Device Resume/Remote Wakeup Detected Interrupt Mask (WkUpIntMsk) The WakeUp bit is used for LPM state wake up in a way similar to that of wake up in suspend state.	0x0

GUID

User ID Register
Offset Address: 0x03c

Bits	Name	Access	Description	Reset
31:0	UserID	R/W	User ID (UserID) Application-programmable ID field. Reset: Configurable	0x0

GLPMCFG

Core LPM Configuration Register
Offset Address: 0x054

Bits	Name	Access	Description	Reset
0	LPMCap	R/W	Mode: Host and Device LPM-Capable (LPMCap) The application uses this bit to control the DWC_otg core LPM capabilities. If the core operates as a non-LPM-capable host, it cannot request the connected device or hub to activate LPM mode. If the core operates as a non-LPM-capable device, it cannot respond to any LPM transactions. ■ 1b0: LPM capability is not enabled ■ 1b1: LPM capability is enabled This bit is writable only if an LPM mode was specified for Mode of Operation in coreConsultant (parameter OTG_ENABLE_LPM). Otherwise, reads return 0. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
1	AppL1Res	R/W	Mode: Device only	0x0

Bits	Name	Access	Description	Reset
			LPM response programmed by application (AppL1Res) Handshake response to LPM token pre-programmed by device application software. The response depends on GLPMCFG.LPMCap. If GLPMCFG.LPMCap is 1'b0, then the core always responds with NYET response. If GLPMCFG.LPMCap is 1'b1, the core response is as follows. ■ 1: ACK Even though ACK is pre-programmed, the core Device responds with ACK only on successful LPM transaction. The LPM transaction is successful if: - No PID/CRC5 Errors in either EXT token or LPM token (else ERROR) - Valid bLinkState = 0001B (L1) received in LPM transaction (else STALL) - No data pending in transmit queue (else NYET). ■ 0: NYET The pre-programmed software bit is over-ridden for response to LPM token when: - The received bLinkState is not L1 (STALL response), or - An error is detected in either of the LPM token packets because of corruption (ERROR response). Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
5:2	HIRD	R/W	Mode: Host and Device ■ EnBESL = 1'b0 Host-Initiated Resume Duration (HIRD) Host Mode: The value of HIRD to be sent in an LPM transaction. This value is also used to initiate resume for a duration TL1HubDrvResume1 for host initiated resume. Device Mode (Read-Only): This field is updated with the Received LPM Token HIRD bmAttribute when an ACK, NYET, or STALL response is sent to an LPM transaction. Sl. No HIRD[3:0] THIRD (µs) 1 4'b0000 50 2 4'b0001 125 3 4'b0010 200 4 4'b0011 275 5 4'b0100 350 6 4'b0101 425 7 4'b0110 500 8 4'b0111 575 9 4'b1000 650 10 4'b1001 725 11 4'b1010 800 12 4'b1011 875 13 4'b1100 950 14 4'b1101 1025	0x0

Bits	Name	Access	Description	Reset
			15 4'b1110 1100 16 4'b1111 1175 Reset: 4'b0000 ■ EnBESL = 1'b1 Best Effort Service Latency (BESL) Host Mode: The value of BESL to be sent in an LPM transaction. This value is also used to initiate resume for a duration TL1HubDrvResume1 for host initiated resume. Device Mode (Read-Only): This field is updated with the Received LPM Token BESL bmAttribute when an ACK, NYET, or STALL response is sent to an LPM transaction. Sl. No BESL[3:0] TBESL (µs) 1 4'b0000 125 2 4'b0001 150 3 4'b0010 200 4 4'b0011 300 5 4'b0100 400 6 4'b0101 500 7 4'b0110 1000 8 4'b0111 2000 9 4'b1000 3000 10 4'b1001 4000 11 4'b1010 5000 12 4'b1011 6000 13 4'b1100 7000 14 4'b1101 8000 15 4'b1110 9000 16 4'b1111 10000 Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
6	bRemoteWake	R/W	Mode: Host and Device RemoteWakeEnable (bRemoteWake) Host Mode: The value of remote wake up to be sent in the wIndex field of LPM transaction. Device Mode (Read-Only): This field is updated with the Received LPM Token bRemoteWake bmAttribute when an ACK, NYET, or STALL response is sent to an LPM transaction.	0x0
7	EnblSlpM	R/W	Mode: Host and Device Enable utmi_sleep_n (EnblSlpM) ULPI Interface: The application uses this bit to control the utmi_sleep_n assertion to the PHY when in L1 state. For the host, this bit is valid only in "local device" mode. ■ 1b0: utmi_sleep_n assertion from the core is not transferred to the external PHY. ■ 1b1: utmi_sleep_n assertion from the core is transferred	0x0

Bits	Name	Access	Description	Reset
			to the external PHY. Note: When a ULPI interface is configured, enabling this bit results in a write to Bit 7 of the ULPI Function Control register. The Synopsys ULPI PHY supports writing to this bit, and in the L1 state asserts SleepM when utmi_l1_suspend_n cannot be asserted. Other Interfaces: The application uses this bit to control utmi_sleep_n assertion to the PHY in the L1 state. For the host, this bit is valid only in Local Device mode. ■ 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.	
12:8	HIRD_Thres	R/W	Mode: Host and Device BESL or HIRD Threshold (HIRD_Thres) Device Mode: ■ 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 1'b1. ■ 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_Thresh[3:0]. The Device sends a NYET response when the received HIRD in LPM token is greater than HIRD threshold. ■ Note: To differentiate between Deep Sleep and Shallow sleep HIRD greater than or equal to HIRD threshold comparison is done. For differentiating between NYET or ACK response for LPM token HIRD greater than HIRD Threshold comparison is used. Host Mode: The core puts the PHY into deep low power mode in L1 (by core asserting L1SuspendM) when HIRD_Thres[4] is set to 1b1. HIRD_Thres[3:0] specifies the time for which resume signaling is to be reflected by host (TL1HubDrvResume2) on the USB bus when it detects device initiated resume.	0x0

Bits	Name	Access	Description	Reset
			HIRD_Thres must not be programmed with a value greater than 4'b1100 in Host mode, because this exceeds maximum TL1HubDrvResume2. Sl. No ----Thres[3:0] --- -Host Mode Resume Signaling Time (µs) --- - --- - --- - --- - --------- - --- - EnBESL = 1'b0 - -EnBESL = 1'b1 1 4'b0000 60 75 2 4'b0001 135 100 3 4'b0010 -210 150 4 4'b0011 285 250 5 4'b0100 360 350 6 4'b0101 435 450 7 4'b0110 510 950 8 4'b0111 585 Invalid 9 4'b1000 660 Invalid 10 4'b1001 735 Invalid 11 4'b1010 810 Invalid 12 4'b1011 885 Invalid 13 4'b1100 960 Invalid 14 4'b1101 Invalid Invalid 15 4'b1110 Invalid Invalid 16 4'b1111 Invalid Invalid The following truth table explains the difference in behavior between the UTMI and ULPI interface in different modes of operation: Bit 7 --Bit 6 --sleep_n ---l1_suspend_n --suspend_n ---Mode of Operation 0 --- -----1- --- ----1--- --- --1--- --- ------- --- --1--- --- ------- -- Normal Operation 0 --- -----0- --- ----1--- --- --1--- --- ------- --- --0--- --- ------- --L2 Suspend 1 --- -----0- --- ----1--- --- --0--- --- ------- --- --1--- --- ------- --L1 Deep Sleep 1 --- -----1- --- ----0--- --- --1--- --- ------- --- --1--- --- ------- --L1 Shallow Sleep	
14:13	CoreL1Res	RO	Mode: Host and Device LPM Response (CoreL1Res) Device Mode: The response of the core to LPM transaction received is reflected in these two bits. Host Mode: Handshake response received from local device for LPM transaction ■ 11 - ACK ■ 10 - NYET ■ 01 - STALL ■ 00 - ERROR (No handshake response)	
15	SlpSts	RO	Mode: Device only Port Sleep Status (SlpSts) 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	

Bits	Name	Access	Description	Reset
			CoreL1Res to confirm transition into sleep. The core comes out of sleep: ■ When there is any activity on the USB linestate ■ When the application writes to the Remote Wakeup Signaling bit in the Device Control register (DCTL.RmtWkUpSig) or when the application resets or softdisconnects the device. Host Mode: The host transitions to Sleep (L1) state as a side-effect of a successful LPM transaction by the core to the local port with ACK response from the device. The read value of this bit reflects the current Sleep status of the port. The core clears this bit after: ■ The core detects a remote L1 Wakeup signal, ■ The application sets the Port Reset bit or the Port L1Resume bit in the HPRT register, or ■ The application sets the L1Resume/ Remote Wakeup Detected Interrupt bit or Disconnect Detected Interrupt bit in the Core Interrupt register (GINTSTS.L1WkUpInt or GINTSTS.DisconnInt, respectively). Values: ■ 1b0: Core not in L1 ■ 1b1: Core in L1	
16	L1ResumeOK	RO	Mode: Host and device Sleep State Resume OK (L1ResumeOK) 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 µs (TL1Residency). This bit is reset when SlpSts is 0. ■ 1b1: The application or core can start Resume from Sleep state ■ 1b0: The application or core cannot start Resume from Sleep state	
20:17	LPM_Chnl_Indx	R/W	Mode: Host only LPM Channel Index (LPM_Chnl_Indx) The channel number on which the LPM transaction has to be applied while sending an LPM transaction to the local device. Based on the LPM channel index, the core automatically inserts the device address and end point number programmed in the corresponding channel into the LPM transaction.	0x0
23:21	LPM_Retry_Cnt	R/W	Mode: Host only LPM Retry Count (LPM_Retry_Cnt) When the device gives an ERROR response, this is the number of additional LPM retries that the host performs until a valid device response (STALL, NYET, or ACK)	0x0

Bits	Name	Access	Description	Reset
			is received.	
24	SndLPM	RWS	Write Behavior: One to set Mode: Host only Send LPM Transaction (SndLPM) When the application software sets this bit, an LPM transaction containing two tokens, EXT and LPM is sent. The hardware clears this bit once a valid response (STALL, NYET, or ACK) is received from the Device or the core has finished transmitting the programmed number of LPM retries. Note: This bit must be set only when the host is connected to a local port.	
27:25	LPM_RetryCnt_Sts	RO	Mode: Host only LPM Retry Count Status (LPM_RetryCnt_Sts) Number of LPM Host Retries still remaining to be transmitted for the current LPM sequence.	
28	EnBESL	R/W	Mode: Host and device Enable Best Effort Service Latency (BESL) This bit enables the BESL feature as defined in the LPM errata: ■ 1'b0: The core works as described in the following document: USB 2.0 Link Power Management Addendum Engineering Change Notice to the USB 2.0 specification, July 16, 2007 ■ 1'b1: The core works as per the LPM Errata	0x0
29	RstrSlpSts	R/W	Mode: Device only Restore SlpSts (RstrSlpSts) When the application power-gates the core (partial power-down or hibernation), the application needs to program this bit to restore the LPM status in the core. Based on the BESL value received from the Host, the application needs to program this bit during restore process. The application should program this bit depending on whether it decided to put the core in Shallow Sleep (Clock Gating Only) or Deep Sleep (Power Gating) mode: ■ 1'b0: The application puts the core in Shallow Sleep mode based on the BESL value from the Host. ■ 1'b1: The application puts the core in Deep Sleep mode based on the BESL value from the Host.	0x0
30	HSICCon	R/W	Mode: Host and device HSIC-Connect (HSICCon) The application must use this bit to initiate the HSIC Attach sequence. Host Mode: Once this bit is set, the Host Core configures to drive HSIC Idle state (STROBE=1&DATA=0) on the bus. It then waits for device to initiate the HSIC Connect sequence. Device Mode: Once this bit is set, the Device Core waits for	0x0

Bits	Name	Access	Description	Reset
			HSIC Idle linestate on the bus. After receiving the Idle linestate it then initiates the HSIC Connect. This bit is valid only if OTG_ENABLE_HSIC = 1, if_sel_hsic = 1, and InvSelHSIC = 0. Otherwise, it is read-only.	
31	InvSelHsic	R/W	Mode: Host and device HSIC-Invert Select HSIC (InvSelHsic) The application uses this bit to control the DWC_otg core HSIC enable/disable. This bit overrides and functionally inverts the if_sel_hsic input port signal. If the core is non-HSIC-capable, it can connect to only PHYs that are not HSIC capable. If the core is HSIC-capable, it can connect only to PHYs that are HSIC capable. ■ If if_sel_hsic input signal is 1: - InvSelHsic = 1b1: HSIC capability is not enabled - InvSelHsic = 1b0: HSIC capability is enabled ■ If if_sel_hsic input signal is 0: - InvSelHsic = 1b1: HSIC capability is enabled - InvSelHsic = 1b0: HSIC capability is not enabled This bit is writable only if HSIC mode is specified for Mode of Operation in coreConsultant (parameter OTG_ENABLE_HSIC). This bit is valid only if OTG_ENABLE_HSIC is enabled. Otherwise, reads return 0.	0x0

GPWRDN

Power Down Register
Offset Address: 0x058

Bits	Name	Access	Description	Reset
0	PMUIntSel	R/W	Mode: Host and Device PMU Interrupt Select (PMUIntSel) When the hibernation functionality is selected (OTG_EN_PWROPT = 2), a write to this bit with 1'b1 enables the PMU to generate interrupts to the application. During this state, all interrupts from the DWC_otg_core module are blocked to the application. Note: This bit must be set to 1'b1 before the core is put into hibernation ■ 1'b0: Internal DWC_otg_core interrupt is selected ■ 1'b1: External DWC_otg_pmu interrupt is selected Note: This bit must not be written to during normal mode of operation. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
1	PMUActv	R/W	Mode: Host and Device PMU Active (PMUActv) This bit enables or disables the PMU logic. ■ 1'b0: Disable PMU module ■ 1'b1: Enable PMU module	0x0

Bits	Name	Access	Description	Reset
			Note: This bit must not be written to during normal mode of operation. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
2	Restore	R/W	Mode: Host and Device Restore The application must program this bit to enable or disable restore mode from the PMU module. ■ 1'b0: DWC_otg in normal mode of operation ■ 1'b1: DWC_otg in restore mode Note: This bit must not be written to during normal mode of operation. This bit is valid only when OTG_EN_PWROPT = 2. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
3	PwrDnClmp	R/W	Mode: Host and Device Power Down Clamp (PwrDnClmp) The application must program this bit to enable or disable the clamps to all the outputs of the DWC_otg core module to prevent the corruption of other active logic. ■ 1'b0: Disable PMU power clamp ■ 1'b1: Enable PMU power clamp	0x0
4	PwrDnRst_n	R/W	Mode: Host and Device Power Down ResetN (PwrDnRst_n) The application must program this bit to reset the DWC_otg core during the Hibernation exit process or during ADP when powering up the core (if the DWC_otg core was powered off during ADP process). ■ 1'b1: DWC_otg is in normal operation ■ 1'b0: Reset DWC_otg Note: This bit must not be written to during normal mode of operation.	0x1
5	PwrDnSwtch	R/W	Mode: Host and Device Power Down Switch (PwrDnSwtch) This bit indicates to the DWC_otg core whether the VDD switch is in ON or OFF state. ■ 1'b0: DWC_otg is in ON state ■ 1'b1: DWC_otg is in OFF state Note: This bit must not be written to during normal mode of operation.	0x0
6	DisableVBUS	R/W	Mode: Host and Device DisableVBUS Host Mode: The application must program this bit if HPRT0.PrtPwr was programmed to 0 before switching off the Core. This indicates to the PMU whether session was ended before entering	0x0

Bits	Name	Access	Description	Reset
			Hibernation. ■ 1'b0: HPRT0.PrtPwr was not programed to 0. ■ 1'b1: HPRT0.PrtPwr was programmed to 0. Device Mode: The application must program this bit to inform the PMU whether the bvalid valid signal is high (session valid) or low (session end) whenever the core is switched off. ■ 1'b0: bvalid signal is High (Session Valid) ■ 1'b1: bvalid signal is Low (Session End) This bit is valid only when GPWRDN.PMUActv is 1.	
7	LnStsChng	RWC	Write Behavior: One to clear Mode: Host and Device Line State Change (LnStsChng) This interrupt is asserted when there is a linestate change detected by the PMU. The application must read GPWRDN.Linestate to determine the current linestate on USB. ■ 1'b0: No LineState change on USB ■ 1'b1: LineState change on USB This bit is valid only when GPWRDN.PMUActv is 1 and OTG_EN_PWROPT = 2.	
8	LineStageChange Msk	R/W	Mode: Host and Device Mask For LineStateChange interrupt (LineStageChangeMsk) This bit is valid only when OTG_EN_PWROPT = 2.	0x0
9	ResetDetected	RWC	Write Behavior: One to clear Mode: Device only ResetDetected This field indicates that Reset has been detected by the PMU module. This field generates an interrupt. ■ 1'b0: Reset Not Detected ■ 1'b1: Reset Detected This bit is valid only when OTG_EN_PWROPT = 2.	
10	Mask_ResetDetMsk	R/W	Mode: Device only Mask For ResetDetected interrupt (ResetDetMsk) This bit is valid only when OTG_EN_PWROPT = 2.	0x0
11	DisconnectDetect	RWC	Write Behavior: One to clear Mode: Host only DisconnectDetect This field indicates that Disconnect has been detected by the PMU. This field generates an interrupt. After detecting disconnect during hibernation the application must not restore the core, but instead start the initialization process. ■ 1'b0: Disconnect not detected ■ 1'b1: Disconnect detected This bit is valid only when OTG_EN_PWROPT = 2.	
12	DisconnectDetect Msk	R/W	Mode: Host only Mask For DisconnectDetect Interrupt (DisconnectDetectMsk) This bit is valid only when OTG_EN_PWROPT = 2.	0x0
13	ConnectDet	RO	Mode: Host and Device	

Bits	Name	Access	Description	Reset
			Write Behavior: One to clear ConnectDet This field indicates that a new connect has been detected ■ 1'b0: Connect not detected ■ 1'b1: Connect detected This bit is valid only when OTG_EN_PWROPT = 2.	
14	ConnDetMsk	R/W	Mode: Host and Device ConnDetMsk Mask for ConnectDet interrupt This bit is valid only when OTG_EN_PWROPT = 2.	0x0
15	SRPDetect	RWC	Mode: Host only SRPDetect This field indicates that SRP has been detected by the PMU. This field generates an interrupt. After detecting SRP during hibernation the application must not restore the core. The application must get into the initialization process. ■ 1'b0: SRP not detected ■ 1'b1: SRP detected	
16	SRPDetectMsk	R/W	Mode: Host only Mask For SRPDetect Interrupt (SRPDetectMsk)	0x0
17	StsChngInt	RWC	Write Behavior: One to clear Status Change Interrupt (StsChngInt) This field indicates a status change in either the IDDIG or BSessVld signal. ■ 1'b0: No Status change ■ 1'b1: status change detected After receiving this interrupt the application must read the GPWRDN register and interpret the change in IDDIG or BSesVld with respect to the previous value stored by the application. Note: When Battery Charger is Enabled and the ULPI interface is used, if StsChngInt is received and the application reads GPWRDN register and determines that it is due to a change in the value of IDDIG, then StsChngInt may be generated once again within the next few clock cycles. This occurs because of an ambiguity in the implementation of Battery Charger Support over the ULPI interface. After receiving the StsChngInt for the second time the application can once again read the GPWRDN register. However, this time the value IDDIG (or BSesVld) will not have changed. The application then processes the second interrupt but no further action will be required as a result.	
18	StsChngIntMsk	R/W	Mode: Host and Device Mask For StsChng Interrupt (StsChngIntMsk)	0x0

Bits	Name	Access	Description	Reset
20:19	LineState	RO	Mode: Host and Device LineState This field indicates the current linestate on USB as seen by the PMU module. ■ 2'b00: DM = 0, DP = 0 ■ 2'b01: DM = 0, DP = 1 ■ 2'b10: DM = 1, DP = 0 ■ 2'b11: Not-defined This bit is valid only when GPWRDN.PMUActv is 1.	
21	IDDIG	RO	Mode: Host and Device IDDIG This bit indicates the status of the IDDIG signal. The application must read this bit after receiving GPWRDN.StsChngInt and decode based on the previous value stored by the application. Indicates the current mode. ■ 1'b1: Device mode ■ 1'b0: Host mode This bit is valid only when GPWRDN.PMUActv is 1.	
22	BSessVld	RO	Mode: Device only B Session Valid (BSessVld) This field reflects the B session valid status signal from the PHY. ■ 1'b0: B-Valid is 0 ■ 1'b1: B-Valid is 1 This bit is valid only when GPWRDN.PMUActv is 1.	
23	ADPInt	RWC	Write Behavior: One to clear Mode: Host and Device ADP Interrupt (ADPInt) This bit is set whenever there is a ADP event	
28:24	MultValIdBC	RO	Mode: Host and Device MultValIdBC (MultValIdBC) Battery Charger ACA inputs in the following order: ■ Bit 28 - rid_float ■ Bit 27 - rid_gnd ■ Bit 26 - rid_a ■ Bit 25 - rid_b ■ Bit 24 - rid_c These bits are present only if BC_SUPPORT = 1. Otherwise, these bits are reserved and will read 5'h0. Reset: As per ACA input	
31:29	Reserved_58_31_29	RO	Reserved for future use.	

12.6.4 The Illustration of Host Initialization Program

After completing the [clock startup procedure] and [mode switching and initialization procedure], the XHCI initialization procedure needs to be executed. As listed below, four standard types of transmission can be started according to the requirements. For details of the method of starting standard transmission, please refer to the XHCI specification, which will not be repeated here.

1. Set the GINTMSK.PrtInt register to the unmask state.
2. Set the HCFG register to configure the FS or HS device.
3. Set the HPRT.PrtPwr register to 1, which turns on the VBUS on the USB bus.
4. Wait for the HPRT0.PrtConnDet interrupt to occur, indicating that a device is connected to the USB downstream port.
5. Set the HPRT.PrtRst register to 1 to begin the USB port reset process.
6. Wait at least 10ms to allow enough time for the USB port reset to complete the handshake.
7. Set the HPRT.PrtRst to 0 to complete the USB port reset process.
8. Wait for the HPRT.PrtEnChng interrupt to occur.
9. Read the HPRT.PrtSpd register to obtain the enumeration speed value.
10. Set the HFIR register to configure the corresponding PHY Clock.
11. Set the RXFSIZE register to configure the size of the RXFIFO.
12. Set the GNPTXFSIZ register to configure the size of the non-periodic transmission TXFIFO.
13. Set the HPTXFSIZ register to configure the size of the periodic transmission TXFIFO.

12.6.5 Host Register Description

The base address of the Host register in the whole memory space is **0x0434_0000**, indicating as HOST_BASE_ADDR in this article. Therefore, the real address of each register of the host controller in the memory space will be [HOST_BASE_ADDR+ relative address].

12.6.5.1 Register Overview

Name	Address Offset	Description
HCFG	0x400	Host Configuration Register
HFIR	0x404	Host Frame Interval Register
HFNUM	0x408	Host Frame Number/Frame Time Remaining Register
HPTXSTS	0x410	Host Periodic Transmit FIFO/Queue Status Register
HAINT	0x414	Host All Channels Interrupt Register
HAINTMSK	0x418	Host All Channels Interrupt Mask Register
HFLBAddr	0x41c	Host Frame List Base Address Register
HCCHARn	0x500	Host Channel-n Characteristics Register
HCDMAn	0x514	Host Channel-n DMA Address Register
HCDMABn	0x51c	Host Channel-n DMA Buffer Address Register

12.6.5.2 Detailed List of Registers

HCFG

Host Configuration Register

Offset Address: 0x400

Bits	Name	Access	Description	Reset
1:0	FSLSPclkSel	R/W	FS/LS PHY Clock Select (FSLSPclkSel) When the core is in FS Host mode: ■ 2'b00: PHY clock is running at 30/60 MHz ■ 2'b01: PHY clock is running at 48 MHz ■ Others: Reserved When the core is in LS Host mode: ■ 2'b00: PHY clock is running at 30/60 MHz. When the UTMI+/ULPI PHY Low Power mode is not selected, use 30/60 MHz. ■ 2'b01: PHY clock is running at 48 MHz. When the UTMI+ PHY Low Power mode is selected, use 48MHz If the PHY supplies a 48 MHz clock during LS mode. ■ 2'b10: PHY clock is running at 6 MHz. In USB 1.1 FS mode, use 6 MHz when the UTMI+ PHY Low Power mode is selected and the PHY supplies a 6 MHz clock during LS mode. If you select a 6 MHz clock during LS mode, you must do a soft reset. ■ 2'b11: Reserved Notes: ■ When Core in FS mode, the internal and external clocks have the same frequency. ■ When Core in LS mode, - If FSLSPclkSel = 2'b00: Internal and external clocks have the same frequency - If FSLSPclkSel = 2'b10: Internal clock is divided by eight version of external 48 MHz clock (utmifs_clk). Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
2	FSLSSupp	R/W	FS- and LS-Only Support (FSLSSupp) The application uses this bit to control the core's enumeration speed. Using this bit, the application can make the core enumerate as a FS host, even If the connected device supports HS traffic. Do not make changes to this field after initial programming. ■ 1'b0: HS/FS/LS, based on the maximum speed supported by the	0x0

Bits	Name	Access	Description	Reset
			connected device ■ 1'b1: FS/LS-only, even If the connected device can support HS Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
6:3	Reserved_400_6_3	RO	Reserved for future use. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
7	Ena32KHzS	R/W	Enable 32 KHz Suspend mode (Ena32KHzS) This bit can be set only if FS PHY interface is selected. Otherwise, this bit needs to be set to zero. When FS PHY interface is chosen and this bit is set, the core expects that the PHY clock is switched from 48 MHz to 32 KHz during Suspend.	0x0
15:8	ResValid	R/W	Resume Validation Period (ResValid) This field is effective only when HCFG.Ena32KHzS is set. It controls the Resume period when the core resumes from Suspend. The core counts the ResValid number of clock cycles to detect a valid resume when this is set.	0x2
22:16	Reserved_400_22_16	RO	Reserved for future use.	
23	DescDMA	R/W	Enable Scatter/gather DMA in Host mode (DescDMA) When the Scatter/Gather DMA option is selected during configuration of the RTL, 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,HCFG.DescDMA=0 => Slave mode ■ GAHBCFG.DMAEn=0,HCFG.DescDMA=1 => Invalid ■ GAHBCFG.DMAEn=1,HCFG.DescDMA=0 => Buffered DMA mode ■ GAHBCFG.DMAEn=1,HCFG.DescDMA=1 => Scatter/Gather DMA mode In non-Scatter/Gather DMA mode, this bit is reserved.	0x0
25:24	FrListEn	R/W	Frame List Entries (FrListEn) The value in the register specifies the number of entries in the Frame list. This field is valid only in Scatter/Gather DMA mode. ■ 2'b00: Reserved ■ 2'b01: 8 Entries ■ 2'b10: 16 Entries ■ 2'b11: 32 Entries In non-Scatter/Gather DMA mode, these bits are reserved.	0x0
26	PerSchedEna	R/W	Enable Periodic Scheduling (PerSchedEna)	0x0

Bits	Name	Access	Description	Reset
			Applicable in Host Scatter/Gather DMA mode only. Enables periodic scheduling within the core. Initially, the bit is res and the core does not process any periodic channels. As soon as this bit is set, the core gets ready to start scheduling periodic channels and sets HCFG.PerSchedStat. The setting of HCFG.PerSchedStat indicates the core has enabled periodic scheduling. Once HCFG.PerSchedEna is set, the application is not supposed to reset the bit unless HCFG.PerSchedStat is set. As soon as this bit is reset, the core gets ready to stop scheduling periodic channels and resets HCFG.PerSchedStat. In non-Scatter/Gather DMA mode, this bit is reserved.	
30:27	Reserved_400_30_27	RO	Reserved for future use.	
31	ModeChTimEn	R/W	Mode Change Ready Timer Enable (ModeChTimEn) This bit is used to enable/disable the Host core to wait 200 PHY clock cycles at the end of Resume to change the opmode signal to the PHY to 00 after Suspend or LPM. ■ 1'b0: The Host core waits for either 200 PHY clock cycles or a linestate of SE0 at the end of resume to the change the opmode from 2'b10 to 2'b00 ■ 1'b1: The Host core waits only for a linestate of SE0 at the end of resume to change the opmode from 2'b10 to 2'b00.	0x0

HFIR

Host Frame Interval Register

Offset Address: 0x404

Bits	Name	Access	Description	Reset
15:0	FrInt	R/W	Frame Interval (FrInt) The value that the application programs to this field specifies the interval between two consecutive SOFs (FS) or micro- SOFs (HS) or Keep-Alive tokens (HS). This field contains the number of PHY clocks that constitute the required frame interval. The default value set in this field for an FS operation when the PHY clock frequency is 60 MHz. The application can write a value to this register only after the Port Enable bit of the Host Port Control and Status register (HPRT.PrtEnaPort) has been set. If no value is programmed, the core calculates the value based on the PHY clock specified in the	0xEA60

Bits	Name	Access	Description	Reset
			FS/LS PHY Clock Select field of the Host Configuration register (HCFG.FSLSPclkSel). Do not change the value of this field after the initial configuration. ■ 125 μs * (PHY clock frequency for HS) ■ 1 ms * (PHY clock frequency for FS/LS) Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
16	HFIRRldCtrl	R/W	Reload Control (HFIRRldCtrl) This bit allows dynamic reloading of the HFIR register during run time. ■ 1'b0: The HFIR cannot be reloaded dynamically ■ 1'b1: The HFIR can be dynamically reloaded during runtime. This bit needs to be programmed during initial configuration and its value must not be changed during runtime.	0x0
31:17	Reserved_404_31_17	RO	Reserved for future use. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	

HFNUM

Host Frame Number/Frame Time Remaining Register

Offset Address: 0x408

Bits	Name	Access	Description	Reset
15:0	FrNum	RO	Frame Number (FrNum) This field increments when a new SOF is transmitted on the USB, and is reset to 0 when it reaches 16'h3FFF. This field is writable only if Remove Optional Features? was not selected in coreConsultant (OTG_RM_OTG_FEATURES = 0). Otherwise, reads return the frame number value. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
31:16	FrRem	RO	Frame Time Remaining (FrRem) Indicates the amount of time remaining in the current microframe (HS) or Frame (FS/LS), in terms of PHY clocks. This field decrements on each PHY clock. When it reaches zero, this field is reloaded with the value in the Frame Interval register and a new SOF is transmitted on the USB. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	

HPTXSTS

Host Periodic Transmit FIFO/Queue Status Register

Offset Address: 0x410

Bits	Name	Access	Description	Reset
15:0	PTxFSpcAvail	RO	Periodic Transmit Data FIFO Space Available (PTxFSpcAvail) Indicates the number of free locations available to be written to in the Periodic TxFIFO. Values are in terms of 32-bit words ■ 16'h0: Periodic TxFIFO is full ■ 16'h1: 1 word available ■ 16'h2: 2 words available ■ 16'hn: n words available (n: 0 ~ 32,768) ■ 16'h8000: 32,768 words available ■ Others: Reserved Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
23:16	PTxQSpcAvail	RO	Periodic Transmit Request Queue Space Available (PTxQSpcAvail) Indicates the number of free locations available to be written in the Periodic Transmit Request Queue. This queue holds both IN and OUT requests. ■ 8'h0: Periodic Transmit Request Queue is full ■ 8'h1: 1 location available ■ 8'h2: 2 locations available ■ n: n locations available (n: 0~16) ■ Others: Reserved Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
31:24	PTxQTop	RO	Top of the Periodic Transmit Request Queue (PTxQTop) This indicates the entry in the Periodic Tx Request Queue that is currently being processes by the MAC. This register is used for debugging. ■ Bit [31]: Odd/Even (micro)Frame - 1'b0: send in even (micro)Frame - 1'b1: send in odd (micro)Frame ■ Bits [30:27]: Channel/endpoint number ■ Bits [26:25]: Type - 2'b00: IN/OUT - 2'b01: Zero-length packet - 2'b10: CSPLIT - 2'b11: Disable channel command ■ Bit [24]: Terminate (last entry for the selected channel or endpoint)	

HAINT

Host All Channels Interrupt Register

Offset Address: 0x414

Bits	Name	Access	Description	Reset
15:0	HAINT	RO	Channel Interrupts (HAINT) One bit per channel: Bit 0 for Channel 0, bit 15 for Channel 15 Shadow: Yes Shadow Ctrl: vs_1t	

Bits	Name	Access	Description	Reset
			Shadow Read Select: shrd_sel One-Way: Enabled	
31:16	Reserved_414_31_16	RO	Reserved for future use	

HAINTMSK

Host All Channels Interrupt Mask Register

Offset Address: 0x418

Bits	Name	Access	Description	Reset
15:0	HAINTMsk	R/W	Channel Interrupt Mask (HAINTMsk) One bit per channel: Bit 0 for channel 0, bit 15 for channel 15 Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
31:16	Reserved_418_31_16	RO	Reserved for future use	

HFLBAddr

Host Frame List Base Address Register

Offset Address: 0x41c

Bits	Name	Access	Description	Reset
31:0	HFLBAddr	R/W	The starting address of the Frame list. This register is used only for Isochronous and Interrupt Channels. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0

HCCHARn

Host Channel-n Characteristics Register

Offset Address: 0x500

Bits	Name	Access	Description	Reset
10:0	MPS	R/W	Maximum Packet Size (MPS) Indicates the maximum packet size of the associated endpoint. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
14:11	EPNum	R/W	Endpoint Number (EPNum) Indicates the endpoint number on the device serving as the data source or sink. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	0x0
15	EPDir	R/W	Endpoint Direction (EPDir) Indicates whether the transaction is IN or OUT. ■ 1'b0: OUT ■ 1'b1: IN	0x0

Bits	Name	Access	Description	Reset
			Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
16	Reserved_500_16	RO	Reserved for future use.	
17	LSpdDev	R/W	Low-Speed Device (LSpdDev) This field is set by the application to indicate that this channel is communicating to a low-speed device. The application must program this bit when a low speed device is connected to the host through an FS HUB. The DWC_otg Host core uses this field to drive the XCVR_SELECT signal to 2'b11 while communicating to the LS Device through the FS hub. Note: In a peer to peer setup, the DWC_otg Host core ignores this bit even if it is set by the application software.	0x0
19:18	EPType	R/W	Endpoint Type (EPType) Indicates the transfer type selected. ■ 2'b00: Control ■ 2'b01: Isochronous ■ 2'b10: Bulk ■ 2'b11: Interrupt	0x0
21:20	EC	R/W	Multi Count (MC) / Error Count (EC) When the Split Enable bit of the Host Channel-n Split Control register (HCSPLTn.SpltEna) is reset (1'b0), this field indicates to the host the number of transactions that must be executed per microframe for this periodic endpoint. For non-periodic transfers, this field is used only in DMA mode, and specifies the number packets to be fetched for this channel before the internal DMA engine changes arbitration. ■ 2'b00: Reserved. This field yields undefined results. ■ 2'b01: 1 transaction ■ 2'b10: 2 transactions to be issued for this endpoint per microframe ■ 2'b11: 3 transactions to be issued for this endpoint per microframe When HCSPLTn.SpltEna is set (1'b1), this field indicates the number of immediate retries to be performed for a periodic split transaction on transaction errors. This field must be set to at least 2'b01.	0x0
28:22	DevAddr	R/W	Device Address (DevAddr) This field selects the specific device serving as the data source or sink.	0x0
29	OddFrm	R/W	Odd Frame (OddFrm) This field is set (reset) by the application to indicate that the OTG host must perform a transfer in an odd (micro)frame. This field is applicable for only periodic (isochronous and interrupt) transactions.	0x0

Bits	Name	Access	Description	Reset
			■ 1'b0: Even (micro)frame ■ 1'b1: Odd (micro)frame This field is not applicable for Scatter/Gather DMA mode and need not be programmed by the application and is ignored by the core.	
30	ChDis	RWS	Write Behavior: One to set Channel Disable (ChDis) The application sets this bit to stop transmitting/receiving data on a channel, even before the transfer for that channel is complete. The application must wait for the Channel Disabled interrupt before treating the channel as disabled.	
31	ChEna	RWS	Write Behavior: One to set Channel Enable (ChEna) When Scatter/Gather mode is enabled: ■ 1'b0: Indicates that the descriptor structure is not yet ready. ■ 1'b1: Indicates that the descriptor structure and data buffer with data is setup and this channel can access the descriptor. When Scatter/Gather mode is disabled: This field is set by the application and cleared by the OTG host. ■ 1'b0: Channel disabled ■ 1'b1: Channel enabled	

HCDMAn

Host Channel-n DMA Address Register

Offset Address: 0x514

Bits	Name	Access	Description	Reset
31:0	DMAAddr	R/W	DMA Address (DMAAddr) This field holds the start address in the external memory from which the data for the endpoint must be fetched or to which it must be stored. This register is incremented on every AHB transaction. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0

HCDMABn

Host Channel-n DMA Buffer Address Register

Offset Address: 0x51c

Bits	Name	Access	Description	Reset
31:0	DMABufAddr	R/W	DMA Address (DMAAddr) Holds the current buffer address. This register is updated as and when the data transfer for the corresponding end point is in progress. This register is present only	0x0

Bits	Name	Access	Description	Reset
			in Scatter/Gather DMA mode. Otherwise this field is reserved. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	

12.6.6 Device Initialization Program

Be sure to complete the [clock start procedure] in 13.8.6.1 before starting and switching to the device function. Please keep to the following steps.

1. Set DescDMA to 1 to start descriptor DMA mode.
2. Set the device speed to HS or FS.
3. Set the non-zero transfer status bit.
4. Set the interval value for periodic transfers.
5. Set the FIFO threshold size for DMA transfer.
6. Clear the DCTL.SftDiscon bit to allow the device to initiate the Connection action with the host.
7. Clear the following bits in GINTMSK:
8. USB Port Reset mask
9. Enumeration done mask
10. Early suspend mask
11. USB suspend mask
12. SOF mask
13. Wait for the GINTSTS.USBReset interrupt to occur and start the USB reset initialization process.
14. Wait for the GINTSTS.EnumerationDone interrupt to occur, indicating that the USB reset program has been completed. Then read the DSTS register to obtain the enumeration speed and start the enumeration initialization process.

12.6.7 Device Register Description

The address of the device register in the whole memory space is **0x0434_0000**, indicating as DEV_BASE_ADDR in this article. Therefore, the real address of each register in the memory space will be [DEV_BASE_ADDR+ relative address].

12.6.7.1 Register Overview

Name	Address Offset	Description
DCFG	0x800	Device Configuration Register
DCTL	0x804	Device Control Register
DSTS	0x808	Device Status Register
DIEPMSK	0x810	Device IN Endpoint Common Interrupt Mask Register
DOEPMSK	0x814	Device OUT Endpoint Common Interrupt Mask Register
DAINT	0x818	Device All Endpoints Interrupt Register
DAINTMSK	0x81c	Device Endpoints Interrupt Mask Register
DIEPEMPMSK	0x834	Device IN Endpoint FIFO Empty Interrupt Mask Register
DEACHINT	0x838	Device Each Endpoint Interrupt Register
DEACHINTMSK	0x83c	Device Each Endpoint Interrupt Register Mask

12.6.7.2 Detailed List of Registers

DCFG

Device Configuration Register

Offset Address: 0x800

Bits	Name	Access	Description	Reset
1:0	DevSpd	R/W	Device Speed (DevSpd) Indicates the speed at which the application requires the core to enumerate, or the maximum speed the application can support. However, the actual bus speed is determined only after the chirp sequence is completed, and is based on the speed of the USB host to which the core is connected. ■ 2'b00: High speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) ■ 2'b01: Full speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) ■ 2'b10: Low speed (USB 1.1 transceiver clock is 6 MHz). If you select 6MHz LS mode, you must do a soft reset. ■ 2'b11: Full speed (USB 1.1 transceiver clock is 48 MHz) Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
2	NZStsOUTHShk	R/W	Non-Zero-Length Status OUT Handshake (NZStsOUTHShk)	0x0

Bits	Name	Access	Description	Reset
			The application can use this field to select the handshake the core sends on receiving a non zero-length data packet during the OUT transaction of a control transfer's Status stage. ■ 1'b1: Send a STALL handshake on a non zero-length status OUT transaction and do not send the received OUT packet to the application. ■ 1'b0: Send the received OUT packet to the application (zero-length or non zero-length) and send a handshake based on the NAK and STALL bits for the endpoint in the Device Endpoint Control register. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
3	Ena32KHzSusp	R/W	Enable 32 KHz Suspend mode (Ena32KHzSusp) This bit can be set only if FS PHY interface is selected. Otherwise, this bit needs to be set to zero. If FS PHY interface is chosen and this bit is set, the PHY clock during Suspend must be switched from 48 MHz to 32 KHz. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
10:4	DevAddr	R/W	Device Address (DevAddr) The application must program this field after every SetAddress control command.	0x0
12:11	PerFrInt	R/W	Periodic Frame Interval (PerFrInt) 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% ■ 2'b10: 90% ■ 2'b11: 95%	0x0
13	EnDevOutNak	R/W	Enable Device OUT NAK (EnDevOutNak) This bit enables setting NAK for Bulk OUT endpoints after the transfer is completed for Device mode Descriptor DMA mode. ■ 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 is a one time programmable bit after reset like any other DCFG register bits. This bit is valid only when OTG_EN_DESC_DMA == 1'b1.	0x0
14	XCVRDLY	R/W	Enables or disables delay between xcvr_sel and txvalid during device chirp	0x0

Bits	Name	Access	Description	Reset
			■ 1'b1: Enable delay between xcvr_sel and txvalid during Device chirp ■ 1'b0: No delay between xcvr_sel and txvalid during Device chirp	
15	ErraticIntMsk	R/W	Mode: Device Erratic Error Interrupt Mask ■ 1'b1: Mask early suspend interrupt on erratic error ■ 1'b0: Early suspend interrupt is generated on erratic error	0x0
17:16	Reserved_800_17_16	RO	Reserved for future use.	
22:18	EPMisCnt	R/W	IN Endpoint Mismatch Count (EPMisCnt) This field is valid only in shared FIFO operation. The application programs this field with a count that determines when the core generates an Endpoint Mismatch interrupt (GINTSTS.EPMis). The core loads this value into an internal counter and decrements it. The counter is reloaded whenever there is a match or when the counter expires. The width of this counter depends on the depth of the Token Queue.	0x8
23	DescDMA	R/W	Enable Scatter/Gather DMA in Device mode (DescDMA). When the Scatter/Gather DMA option is selected during configuration of the RTL, 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 => Slave mode ■ GAHBCFG.DMAEn=0,DCFG.DescDMA=1 => Invalid ■ GAHBCFG.DMAEn=1,DCFG.DescDMA=0 => Buffer DMA mode ■ GAHBCFG.DMAEn=1,DCFG.DescDMA=1 => Scatter/Gather DMA mode	0x0
25:24	PerSchIntvl	R/W	Periodic Scheduling Interval (PerSchIntvl) PerSchIntvl must be programmed only 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, 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.	0x0

Bits	Name	Access	Description	Reset
			■ 2'b01: 50% of (micro)frame. ■ 2'b10: 75% of (micro)frame. ■ 2'b11: Reserved.	
31:26	ResValid	R/W	Resume Validation Period (ResValid) This field is effective only when DCFG.Ena32KHzSusp 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.	0x2

DCTL

Device Control Register

Offset Address: 0x804

Bits	Name	Access	Description	Reset
0	RmtWkUpSig	R/W	Remote Wakeup Signaling (RmtWkUpSig) When the application sets this bit, the core initiates remote signaling to wake 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 µs (TL1DevDrvResume) after being set by the application. The application must not set this bit when GLPMCFG bRemoteWake from the previous LPM transaction is zero. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
1	SftDiscon	R/W	Soft Disconnect (SftDiscon) The application uses this bit to signal the DWC_otg core to do a soft disconnect. As long as this bit is set, the host does not see that the device is connected, and the device 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. Note: This bit is not impacted by a soft reset. Shadow: Yes Shadow Ctrl: vs_1t	0x1

Bits	Name	Access	Description	Reset
			Shadow Read Select: shrd_sel	
2	GNPINNakSts	RO	Global Non-periodic IN NAK Status (GNPINNakSts) ■ 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. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
3	GOUTNakSts	RO	Global OUT NAK Status (GOUTNakSts) ■ 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.	
6:4	TstCtl	R/W	Test Control (TstCtl) ■ 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	0x0
7	SGNPInNak	RWC	Set Global Non-periodic IN NAK (SGNPInNak) A write to this field sets the Global Non-periodic IN NAK.The application uses this bit to send a NAK handshake on all nonperiodic IN endpoints. The core can also set this bit when a timeout condition is detected on a non-periodic endpoint in shared FIFO operation. 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.	
8	CGNPInNak	RWC	Clear Global Non-periodic IN NAK (CGNPInNak) A write to this field clears the Global Non-periodic IN NAK.	
9	SGOUTNak	RWC	Set Global OUT NAK (SGOUTNak) 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.	
10	CGOUTNak	RWC	Clear Global OUT NAK (CGOUTNak) A write to this field clears the Global OUT NAK.	
11	PWROnPrgDone	R/W	Power-On Programming Done (PWROnPrgDone) The application uses this bit to indicate that register programming is complete after a wake-up from Power Down mode.	0x0
12	Reserved_804_12	RO	Reserved for future use.	
14:13	GMC	R/W	Global Multi Count (GMC) GMC must be programmed only once after initialization. Applicable only for Scatter/Gather DMA mode. This indicates	0x0

Bits	Name	Access	Description	Reset
			the number of packets to be serviced for that end point before moving to the next end point. It is only for non-periodic end points. ■ 2'b00: Invalid. ■ 2'b01: 1 packet. ■ 2'b10: 2 packets. ■ 2'b11: 3 packets. The value of this field automatically changes to 2'h1 when DCFG.DescDMA is set to 1. When Scatter/Gather DMA mode is disabled, this field is reserved and reads 2'b00.	
15	IgnrFrmNum	R/W	Ignore frame number for isochronous endpoints (IgnrFrmNum) Slave Mode (GAHBCFG.DMAEn=0): This bit is not valid in Slave mode and should not be programmed to 1.Non-Scatter/Gather DMA mode (GAHBCFG.DMAEn=1,DCFG.DescDMA=0): This bit is not used when Threshold mode is enabled and should not be programmed to 1. In non-Scatter/Gather DMA mode, the application receives transfer complete interrupt after transfers for multiple (micro)frames are completed. ■ When Scatter/Gather DMA mode is disabled, this field is used by the application to enable periodic transfer interrupt. The application can program periodic endpoint transfers for multiple (micro)frames. - 0: Periodic transfer interrupt feature is disabled; the application must program transfers for periodic endpoints every (micro)frame - 1: Packets are not flushed when an ISOC IN token is received for an elapsed frame. The core ignores the frame number, sending packets as soon as the packets are ready, and the corresponding token is received. This field is also used by the application to enable periodic transfer interrupts. Scatter/Gather DMA Mode (GAHBCFG.DMAEn=1,DCFG.DescDMA=1): This bit is not applicable to high-speed, high-bandwidth transfers and should not be programmed to 1. In addition, this bit is not used when Threshold mode is enabled and should not be programmed to 1. ■ 0: The core transmits the packets only in the frame number in which they are intended to be transmitted. ■ 1: Packets are not flushed when an ISOC IN token is received for an elapsed frame. The core ignores the frame number, sending packets as soon as the packets are ready, and the corresponding token is received. When this bit is set, there must be only one packet per descriptor.	0x0
16	NakOnBble	R/W	NAK on Babble Error (NakOnBble) Set NAK automatically on babble (NakOnBble). The core sets NAK automatically for the endpoint on which babble is received.	0x0
17	EnContOnBNA	R/W	Enable Continue on BNA (EnContOnBNA) This bit enables the DWC_otg core to continue on BNA for Bulk OUT and INTR 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.	0x0

Bits	Name	Access	Description	Reset
			■ 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. This bit is valid only when OTG_EN_DESC_DMA == 1'b1. It is a one-time programmable after reset bit like any other DCTL register bits.	
18	DeepSleepBESL Reject	R/W	Deep Sleep BESL Reject 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.	0x0
31:19	Reserved_804_31_19	RO	Reserved for future use.	

DSTS

Device Status Register

Offset Address: 0x808

Bits	Name	Access	Description	Reset
0	SuspSts	RO	Suspend Status (SuspSts) 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 ■ If the application writes to the Remote Wakeup Signaling bit in the Device Control register (DCTL.RmtWkUpSig). Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
2:1	EnumSpd	RO	Enumerated Speed (EnumSpd) Indicates the speed at which the DWC_otg core has come up after speed detection through a chirp sequence. ■ 2'b00: High speed (PHY clock is running at 30 or 60 MHz) ■ 2'b01: Full speed (PHY clock is running at 30 or 60 MHz) ■ 2'b10: Low speed (PHY clock is running at 6 MHz) ■ 2'b11: Full speed (PHY clock is running at 48 MHz) Low speed is not supported for devices using a UTMI+ PHY.	
3	ErrticErr	RO	Erratic Error (ErrticErr) 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) seen on the UTMI+. Due to erratic errors, the DWC_otg core goes into Suspend state and an interrupt is generated to the application with Early	

Bits	Name	Access	Description	Reset
			Suspend bit of the Core Interrupt register (GINTSTS.ErlySusp). If the early suspend is asserted because of an erratic error, the application can only perform a soft disconnect recover.	
7:4	Reserved_808_7_4	RO	Reserved for future use.	
21:8	SOFFN	RO	Frame or Microframe Number of the Received SOF (SOFFN) 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.	
23:22	DevLnSts	RO	Device Line Status (DevLnSts) Indicates the current logic level USB data lines ■ Bit [23]: Logic level of D+ ■ Bit [22]: Logic level of D–	
31:24	Reserved_808_31_24	RO	Reserved for future use. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	

DIEPMSK

Device IN Endpoint Common Interrupt Mask Register

Offset Address: 0x810

Bits	Name	Access	Description	Reset
0	DiXferComplMsk	R/W	Transfer Completed Interrupt Mask (XferComplMsk) Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
1	DiEPDisbldMsk	R/W	Endpoint Disabled Interrupt Mask (EPDisbldMsk)	0x0
2	DiAHBErrMsk	R/W	AHB Error Mask (AHBErrMsk)	0x0
3	TimeOUTMsk	R/W	Timeout Condition Mask (TimeOUTMsk) (Non-isochronous endpoints)	0x0
4	INTknTXFEmpMsk	R/W	IN Token Received When TxFIFO Empty Mask (INTknTXFEmpMsk)	0x0
5	INTknEPMisMsk	R/W	IN Token received with EP Mismatch Mask (INTknEPMisMsk)	0x0
6	INEPNakEffMsk	R/W	IN Endpoint NAK Effective Mask (INEPNakEffMsk)	0x0
7	Reserved_810_7	RO	Reserved for future use.	
8	TxfifoUndrnMsk	R/W	Fifo Underrun Mask (TxfifoUndrnMsk)	0x0
9	BNAInIntrMsk	R/W	BNA Interrupt Mask (BNAInIntrMsk) This bit is valid only when Device Descriptor DMA is enabled.	0x0

Bits	Name	Access	Description	Reset
12:10	Reserved_810_12_10	RO	Reserved for future use.	
13	DiNAKMsk	R/W	NAK interrupt Mask (NAKMsk) Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	0x0
31:14	Reserved_810_31_14	RO	Reserved for future use.	

DOEPMSK

Device OUT Endpoint Common Interrupt Mask Register

Offset Address: 0x814

Bits	Name	Access	Description	Reset
0	XferComplMsk	R/W	Transfer Completed Interrupt Mask (XferComplMsk) Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
1	EPDisbldMsk	R/W	Endpoint Disabled Interrupt Mask (EPDisbldMsk)	0x0
2	AHBErrMsk	R/W	AHB Error (AHBErrMsk)	0x0
3	SetUPMsk	R/W	SETUP Phase Done Mask (SetUPMsk) Applies to control endpoints only.	0x0
4	OUTTknEPdisMsk	R/W	OUT Token Received when Endpoint Disabled Mask (OUTTknEPdisMsk) Applies to control OUT endpoints only.	0x0
5	StsPhseRcvdMsk	R/W	Status Phase Received Mask (StsPhseRcvdMsk)	0x0
6	Back2BackSETup Msk	R/W	Back-to-Back SETUP Packets Received Mask (Back2BackSETupMsk) Applies to control OUT endpoints only.	0x0
7	Reserved_814_7	RO	Reserved for future use.	
8	OutPktErrMsk	R/W	OUT Packet Error Mask (OutPktErrMsk)	0x0
9	BnaOutIntrMsk	R/W	BNA interrupt Mask (BnaOutIntrMsk)	0x0
11:10	Reserved_814_11 _10	RO	Reserved for future use.	
12	BbleErrMsk	R/W	Babble Error interrupt Mask (BbleErrMsk)	0x0
13	NAKMsk	R/W	NAK interrupt Mask (NAKMsk)	0x0
14	NYETMsk	R/W	NYET interrupt Mask (NYETMsk) Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	0x0
31:15	Reserved_814_31 _15	RO	Reserved for future use.	

DAINT

Device All Endpoints Interrupt Register

Offset Address: 0x818

Bits	Name	Access	Description	Reset
15:0	InEpInt	RO	OUT Endpoint Interrupt Bits (OutEPInt) One bit per OUT endpoint: Bit 16 for OUT endpoint 0, bit 31 for OUT endpoint 15 Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	

Bits	Name	Access	Description	Reset
31:16	OutEPInt	RO	OUT Endpoint Interrupt Bits (OutEPInt) One bit per OUT endpoint: Bit 16 for OUT endpoint 0, bit 31 for OUT endpoint 15	

DAINTMSK

Device Endpoints Interrupt Mask Register

Offset Address: 0x81c

Bits	Name	Access	Description	Reset
15:0	InEpMsk	R/W	IN EP Interrupt Mask Bits (InEpMsk) One bit per IN Endpoint: Bit 0 for IN EP 0, bit 15 for IN EP 15 The value of this field depends on the number of IN endpoints that are configured. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
31:16	OutEpMsk	R/W	OUT EP Interrupt Mask Bits (OutEpMsk) One per OUT endpoint: Bit 16 for OUT EP 0, bit 31 for OUT EP 15 The value of this field depends on the number of OUT endpoints that are configured.	0x0

DIEPEMPMSK

Device IN Endpoint FIFO Empty Interrupt Mask Register

Offset Address: 0x834

Bits	Name	Access	Description	Reset
15:0	InEpTxfEmpMsk	R/W	IN EP Tx FIFO Empty Interrupt Mask Bits (InEpTxfEmpMsk) These bits acts as mask bits for DIEPINTn. TxFEmp interrupt One bit per IN Endpoint: ■ Bit 0 for IN endpoint 0 ... ■ Bit 15 for endpoint 15 Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
31:16	Reserved_834_31_16	RO	Reserved for future use.	

DEACHINT

Device Each Endpoint Interrupt Register

Offset Address: 0x838

Bits	Name	Access	Description	Reset
15:0	EchInEpInt	RO	IN Endpoint Interrupt Bits (EchInEpInt) One bit per IN Endpoint: ■ Bit 0 for IN endpoint 0 ... ■ Bit 15 for endpoint 15 Shadow: Yes Shadow Ctrl: vs_1t	

Bits	Name	Access	Description	Reset
			Shadow Read Select: shrd_sel One-Way: Enabled	
31:16	EchOutEPInt	RO	OUT Endpoint Interrupt Bits (EchOutEPInt) One bit per OUT endpoint: ■ Bit 16 for OUT endpoint 0 ... ■ Bit 31 for OUT endpoint 15	

DEACHINTMSK

Device Each Endpoint Interrupt Register Mask

Offset Address: 0x83c

Bits	Name	Access	Description	Reset
15:0	EchInEpMsk	R/W	IN EP Interrupt Mask Bits (EchInEpMsk) One bit per IN Endpoint: ■ Bit 0 for IN endpoint 0 ... ■ Bit 15 for endpoint 15 Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
31:16	EchOutEpMsk	R/W	OUT EP Interrupt Mask Bits (EchOutEpMsk) One per OUT Endpoint: ■ Bit 16 for IN endpoint 0 ... ■ Bit 31 for endpoint 15	0x0

12.7 SARADC

12.7.1 Overview

The SARADC is an analog signal to digital conversion controller. There is a SARADC controller in this chip and 3 ADC channels on it .

12.7.2 Features

- Controller operating frequency 12.5MHz
- Samplig frequency can not be higher than 320K/S
- 12bit sampling accuracy, 3 independent channels
- Three channels can be triggered at once

- With sampling complete interrupt

12.7.3 Working method

The CPU configures the scanning channels, 3 channels can be configured at the same time, and then starts SARADC for channel sampling. After finishing all the enabled channels sampling, the system is notified of the completion of sampling through interrupts, and the CPU can obtain the conversion results.

12.7.4 SARADC register overview

SARADC Base address 0x030F0000

RTCSYS_SARADC Base address 0x0502C000

Name	Address Offset	Description
saradc_ctrl	0x004	control register
saradc_status	0x008	staus register
saradc_cyc_set	0x00c	saradc waveform setting register
saradc_ch1_result	0x014	channel 1 result register
saradc_ch2_result	0x018	channel 2 result register
saradc_ch3_result	0x01c	channel 3 result register
saradc_intr_en	0x020	interrupt enable register
saradc_intr_clr	0x024	interrupt clear register
saradc_intr_sta	0x028	interrupt status register
saradc_intr_raw	0x02c	interrupt raw status register

12.7.5 SARADC register description

saradc_ctrl

Offset Address: 0x004

Bits	Name	Access	Description	Reset
0	reg_saradc_en	RWS	when re_saradc_en is set , saradc start to measure the channels enabled in reg_saradc_sel	
3:1	Reserved			
7:4	reg_saradc_sel	R/W	select channel (1~3)	0x0
31:8	Reserved			

saradc_status

Offset Address: 0x008

Bits	Name	Access	Description	Reset

Bits	Name	Access	Description	Reset
0	sta_saradc_busy	RO	busy rise when re_saradc_en is set	
3:1	Reserved			
7:4	sta_saradc_ch_busy	RO	per channel busy status	
15:8	Reserved			
19:16	sta_saradc_st	RO	fsm status for debug	
24:20	sta_saradc_cycle	RO	sample cycle for debug	
31:25	Reserved			

saradc_cyc_set

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
4:0	reg_saradc_cyc_settling	R/W	saradc startup cycle = 1 + reg_saradc_cyc_settling , default is 16 cycle	0xF
7:5	Reserved			
11:8	reg_saradc_cyc_samp	R/W	saradc sample window = 1 + reg_saradc_cyc_samp , default is 4 cycle	0x3
15:12	reg_saradc_cyc_clkdiv	R/W	saradc clock divider , freq = ip_clk/(1+clk_div) , default is 25M/2 = 12.5M = 80ns	0x1
19:16	reg_saradc_cyc_comp	R/W	saradc compare cycle = 1+ reg_saradc_cyc_comp , default is 12 cycle	0xB
31:20	Reserved			

saradc_ch1_result

Offset Address: 0x014

Bits	Name	Access	Description	Reset
11:0	sta_saradc_ch1_result	RO	ch1 measure result	
14:12	Reserved			
15	sta_saradc_ch1_valid	RO	ch1 measure result is valid. The valid status will be cleared when this channel is re-triggered to remeasure,	
31:16	Reserved			

saradc_ch2_result

Offset Address: 0x018

Bits	Name	Access	Description	Reset
11:0	sta_saradc_ch2_result	RO	ch2 measure result	
14:12	Reserved			
15	sta_saradc_ch2_valid	RO	ch2 measure result is valid. The valid status will be cleared when this channel is re-triggered to remeasure,	
31:16	Reserved			

saradc_ch3_result

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
11:0	sta_saradc_ch3_result	RO	ch3 measure result	
14:12	Reserved			
15	sta_saradc_ch3_valid	RO	ch3 measure result is valid. The valid status will be cleared when this channel is re-triggered to remeasure,	
31:16	Reserved			

saradc_intr_en

Offset Address: 0x020

Bits	Name	Access	Description	Reset
0	sta_saradc_intr_en	R/W	interrupt enable (mask)	0x0
31:1	Reserved			

saradc_intr_clr

Offset Address: 0x024

Bits	Name	Access	Description	Reset
0	sta_saradc_intr_clr	RWC	interrupt clear	
31:1	Reserved			

saradc_intr_sta

Offset Address: 0x028

Bits	Name	Access	Description	Reset
0	sta_saradc_intr_sta	RO	interrup masked status [0]: all channels measurenment in this time is finished	
31:1	Reserved			

saradc_intr_raw

Offset Address: 0x02c

Bits	Name	Access	Description	Reset
0	sta_saradc_intr_raw	RO	interrupt raw status [0]: all channels measurenment in this time is finished	
31:1	Reserved			

12.8 Temperature sensor

12.8.1 Overview

The chip has two built-in temperature sensors to monitor the chip temperature periodically. The power management module can be triggered to reset the system when the chip is overheated and the CPU cannot respond to the overheating interrupt to avoid the risk of overheating.

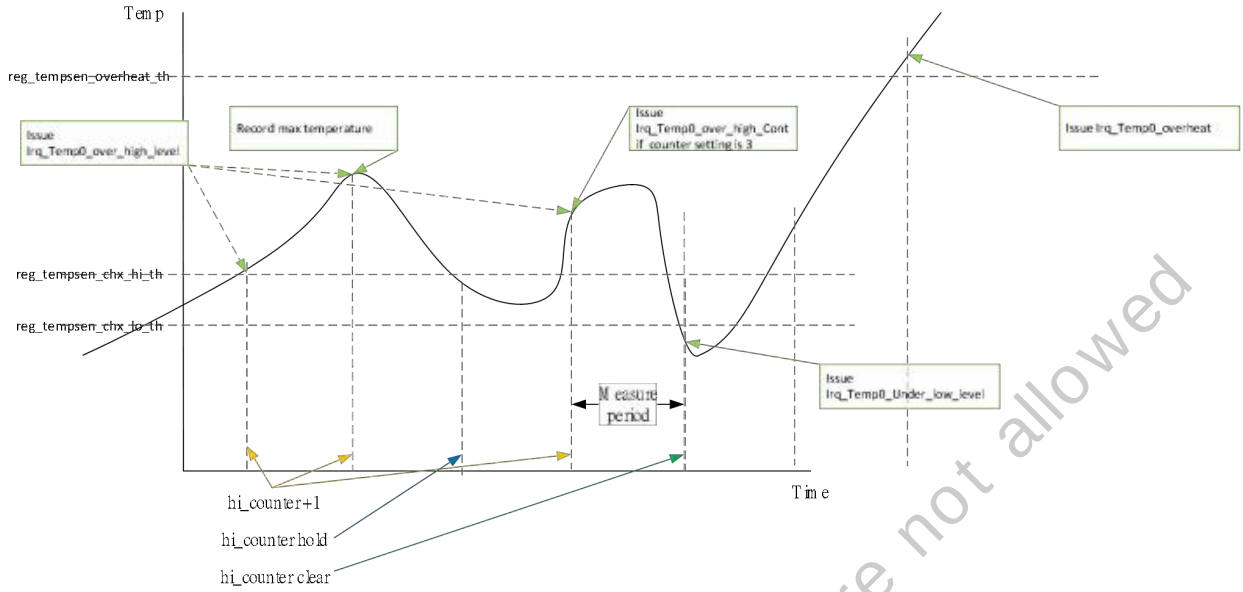
12.8.2 Working method

- Single measurement time:
  If setting reg_tempsen_accsel to 1 (1024T), the sampling time will be (1/(25M/12))*(1024+2+64) ~ 523.2us
- Period measurement time:
  default value of reg_tempsen_auto_prediv is 24, it make timing unit of reg_tempsen_auto_cycle to be 1us. The period time between two samples should be configured larger than the single measurement time. Configure reg_tempsen_auto_cycle = 1000000 if measured once per second.
- Configure measurement channnel:
  Configure reg_tempsen_sel[1:0] = 0x3 to make two temperature sensors will be measured simultaneously。
- Configure high and low temperature monitoring thresholds:
  Configure the temperature threshold that triggers high temperature alarm and low temperature recovery to the register tempsen_chx_temp_th.
- Configure interrupt setting
- Enable temperature sensor for measurement:
  Configure reg_tempsen_en to 1 to trigger measurement then wait for interruption.

Table 12- 9 Tempsensor Interrupt vector description

中斷位	信號	描述
[0]	Irq_Temp0_measure	Tempsens0 量測完成
[1]	Irq_Temp1_measure	Tempsens1 量測完成
[2]	reserved	保留

中斷位	信號	描述
[3]	reserved	保留
[4]	Irq_Temp0_over_high_level	Tempsens0 溫度大於、等於高溫臨界值
[5]	Irq_Temp1_over_high_level	Tempsens1 溫度大於、等於高溫臨界值
[6]	reserved	保留
[7]	reserved	保留
[8]	Irq_Temp0_under_low_level	Tempsens0 溫度小於、等於低溫臨界值
[9]	Irq_Temp1_under_low_level	Tempsens1 溫度小於、等於低溫臨界值
[10]	reserved	保留
[11]	reserved	保留
[12]	Irq_Temp0_over_high_cont	Tempsens0 溫度大於、等於高溫臨界值已達 reg_tempsen_ovhl_cnt_to_irq 次數
[13]	Irq_Temp1_over_high_cont	Tempsens1 溫度大於、等於高溫臨界值已達 reg_tempsen_ovhl_cnt_to_irq 次數
[14]	reserved	保留
[15]	reserved	保留
[16]	Irq_Temp0_under_low_cont	Tempsens0 溫度小於、等於低溫臨界值已達 reg_tempsen_udll_cnt_to_irq 次數
[17]	Irq_Temp1_under_low_cont	Tempsens1 溫度小於、等於低溫臨界值已達 reg_tempsen_udll_cnt_to_irq 次數
[18]	reserved	保留
[19]	reserved	保留
[20]	Irq_Temp0_overheat	Tempsens0 溫度大於、等於過熱臨界值
[21]	Irq_Temp1_overheat	Tempsens1 溫度大於、等於過熱臨界值
[22]	reserved	保留
[23]	reserved	保留



- *Figure 12- 40 Relationship between temperature measurement time, count and interruption*

- Check the temperature measurement results: sta_tempsen_chX_result record the temperature measurement results of the previous time , sta_tempsen_chX_max_result records the maximum temperature measured, and tempsen_chX_temp_th_cnt records the times of continuous high temperature and low temperature.
- Configures overheating protection reg_tempsen_overheat_th、Overtemperature reset request countdown time reg_tempsen_overheat_cycleand enables reg_overheat_reset_en.
  Please refer to register chapter of RTC, it configures hw_thm_shdn_en, RTC_EN_THM_SHDN and RTC_THM_SHDN_AUTO_REBOOT Enable overheating to trigger power down or restart. In the case of overheating, the temp sensor controller will first issue an interrupt and start counting down. When sta_tempsen_overheat_countdown equals to 1, it triggers power down protection request of RTC. If the software has been involved before and controlled the temperature, it can clear the overheat countdownreg_overheat_reset_clr. However, when the next measurement result is still overheated, the overheat protection interrupt will trigger and countdown again.

12.8.3 Temperature sensor register overview

BassAddress: 0x030A0000

Name	Address Offset	Description
tempsen_version	0x000	ip version number
tempsen_ctrl	0x004	control register
tempsen_status	0x008	staus register
tempsen_set	0x00c	temperature sensor macro setting
tempsen_intr_en	0x010	interrupt enable
tempsen_intr_clr	0x014	interrupt clear
tempsen_intr_sta	0x018	interupt status
tempsen_intr_raw	0x01c	interrupt raw status
tempsen_ch0_result	0x020	temperature sensor channel 0 result
tempsen_ch1_result	0x024	temperature sensor channel 1 result
tempsen_ch0_temp_th	0x040	temperature sensor channel 0 threshold
tempsen_ch1_temp_th	0x044	temperature sensor channel 1 threshold
Overheat_th	0x060	overheat threshold register
tempsen_auto_period	0x064	auto sample setting register
tempsen_overheat_ctrl	0x068	overheat control register
tempsen_overheat_countdown	0x06c	overheat status register
tempsen_ch0_temp_th_cnt	0x070	counter of channel 0 over/under threshold event
tempsen_ch1_temp_th_cnt	0x074	counter of channel 1 over/under threshold event

12.8.4 Temperature sensor register description

tempsen_version

Offset Address: 0x000

Bits	Name	Access	Description	Reset
31:0	reg_ip_version	RO	verision 1.0	

tempsen_ctrl

Offset Address: 0x004

Bits	Name	Access	Description	Reset
0	reg_tempsen_en	R/W	when re_tempsen_en is set , tempsen start to measure the channel set in reg_tempsen_sel	0x0
3:1	Reserved			
7:4	reg_tempsen_sel	R/W	temperature sense channel selection	0x0
15:8	Reserved			
23:16	reg_tempsen_ovhl_cnt_to_irq	R/W	counting threshold of high temperature	0x8

Bits	Name	Access	Description	Reset
31:24	reg_tempsen_udll_cnt_to_irq	R/W	counting threshold of low temperature	0x8

tempsen_status

Offset Address: 0x008

Bits	Name	Access	Description	Reset
0	sta_tempsen_busy	RO	busy status rise when re_tempsen_en is set	
31:1	Reserved			

tempsen_set

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
0	reg_tempsen_bgen	R/W	sensor macro bandgap enable	0x0
1	reg_tempsen_chopen	R/W	sensor macro chopper function enable	0x1
2	reg_tempsen_choppol	R/W	sensor macro chopper polarity when CHOPEN=0	0x1
3	reg_tempsen_clkpol	R/W	sensor macro clock polarity when DA_TEMPSEN_EN=0	0x1
5:4	reg_tempsen_chopsel	R/W	sensor macro chop period, 0:128T, 1:256T, 2:512T, 3:1024T	0x2
7:6	reg_tempsen_accsel	R/W	sensor macro accumulate period, 0:512T, 1:1024T, 2:2048T, 3:4096T	0x1
15:8	reg_tempsen_cyc_clkdiv	R/W	clock divider for sensor macro, freq = ip_clk/(1+clk_div) , default is 25M/12 =2.083M , T = 0.48us	0xB
17:16	reg_tempsen_tsel	R/W	sensor macro test selection, please keep 0	0x0
18	reg_tempsen_en_bjt_test	R/W	sensor macro test selection, please keep 0	0x0
31:19	Reserved			

tempsen_intr_en

Offset Address: 0x010

Bits	Name	Access	Description	Reset
31:0	sta_tempsen_intr_en	R/W	interrupt enable	0x0

tempsen_intr_clr

Offset Address: 0x014

Bits	Name	Access	Description	Reset
31:0	sta_tempsen_intr_clr	RWC	interrupt clear	

tempsen_intr_sta

Offset Address: 0x018

Bits	Name	Access	Description	Reset
31:0	sta_tempsen_intr_sta	RO	interrupt masked status	

tempsen_intr_raw

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
31:0	sta_tempsen_intr_raw	RO	interrupt raw status: [3:0] ch3~ch0 measurement finish [7:4] ch3~ch0 measurement result is higher than high temperature threshold [11:8] ch3~ch0 measurement result is lower than low temperature threshold [15:12] ch3~ch0's high temperature event count is more than threshold [19:16] ch3~ch0's low temperature event count is more than threshold [23:20] ch3~ch0 measurement result is higher than overheat temperature	

tempsen_ch0_result

Offset Address: 0x020

Bits	Name	Access	Description	Reset
12:0	sta_tempsen_ch0_result	RO	channel 0 current temperature measurement result	
15:13	Reserved			
28:16	sta_tempsen_ch0_max_result	RO	channel 0 max temperature measurement result	
30:29	Reserved			
31	clr_tempsen_ch0_max_result	RWC	write 1 to clear channel 0 max temperature measurement result	

tempsen_ch1_result

Offset Address: 0x024

Bits	Name	Access	Description	Reset
12:0	sta_tempsen_ch1_result	RO	channel 1 current temperature measurement result	
15:13	Reserved			
28:16	sta_tempsen_ch1_max_result	RO	channel 1 max temperature measurement result	
30:29	Reserved			
31	clr_tempsen_ch1_max_result	RWC	write 1 to clear channel 1 max temperature measurement result	

tempsen_ch0_temp_th

Offset Address: 0x040

Bits	Name	Access	Description	Reset
12:0	reg_tempsen_ch0_hi_th	R/W	channel 0 high temperature threshold to trigger interrupt	0x0
15:13	Reserved			
28:16	reg_tempsen_ch0_lo_th	R/W	channel 0 low temperature threshold to trigger interrupt	0x0
31:29	Reserved			

tempsen_ch1_temp_th

Offset Address: 0x044

Bits	Name	Access	Description	Reset

Bits	Name	Access	Description	Reset
12:0	reg_tempsen_ch1_hi_th	R/W	channel 1 high temperature threshold to trigger interrupt	0x0
15:13	Reserved			
28:16	reg_tempsen_ch1_lo_th	R/W	channel 1 low temperature threshold to trigger interrupt	0x0
31:29	Reserved			

Overheat_th

Offset Address: 0x060

Bits	Name	Access	Description	Reset
12:0	reg_tempsen_overheat_th	R/W	overheat temperature threshold	0x0
31:13	Reserved			

tempsen_auto_period

Offset Address: 0x064

Bits	Name	Access	Description	Reset
23:0	reg_tempsen_auto_cycle	R/W	auto measure period. T_measure = reg_tempsen_auto_cycle*T_prediv	0x0
31:24	reg_tempsen_auto_prediv	R/W	a predivider setting for auto measure period. T_prediv = (25M/(reg_tempsen_auto_prediv+1))	0x18

tempsen_overheat_ctrl

Offset Address: 0x068

Bits	Name	Access	Description	Reset
29:0	reg_tempsen_overheat_cycle	R/W	After overheat event happens, the cycle count will be load to a counter and trigger counting down. when counting down to 1, a reset signal will be issue to power control unit.	0x100000
30	reg_overheat_reset_clr	RWC	write 1 to stop overheat reset counting.	
31	reg_overheat_reset_en	R/W	enable overheat reset counting down.	0x0

tempsen_overheat_countdown

Offset Address: 0x06c

Bits	Name	Access	Description	Reset
29:0	sta_tempsen_overheat_countdown	RO	overheat reset counter	
30	Reserved			
31	sta_overheat_reset	RO	overheat reset signal status	

tempsen_ch0_temp_th_cnt

Offset Address: 0x070

Bits	Name	Access	Description	Reset
7:0	sta_ch0_over_hi_temp_th_cnt	RO	channel 0 high temperature event count status	
15:8	sta_ch0_under_lo_temp_th_cnt	RO	channel 0 low temperature event count status	
16	reg_ch0_temp_th_cnt_clr	RWC	write 1 to clear channel 0 temperature	

Bits	Name	Access	Description	Reset
			event count	
31:17	Reserved			

tempsen_ch1_temp_th_cnt

Offset Address: 0x074

Bits	Name	Access	Description	Reset
7:0	sta_ch1_over_hi_temp_th_cnt	RO	channel 1 high temperature event count status	
15:8	sta_ch1_under_lo_temp_th_cnt	RO	channel 1 low temperature event count status	
16	reg_ch1_temp_th_cnt_clr	RWC	write 1 to clear channel 1 temperature event count	
31:17	Reserved			

12.9 PWM

12.9.1 Overview

The chip provides a set of four independent PWM channel outputs.

12.9.2 Features

The clock source for PWM is either 100MHz or 148.5MHz (default is 100MHz). Each PWM channel can operate independently:

- Support 30-bit period counter and high/low level counter for PWM waveform
- Maximum PWM frequency up to 50MHz (100MHz/2) or 74.25MHz (148.5MHz/2), and minimum PWM frequency down to 0.093Hz (100MHz/(2^30-1)) or 0.138Hz (148.5MHz/(2^32-1))
- Support continuous mode (PWMMODE = 0) and fixed pulse count mode (PWMMODE = 1)
- Support 4-channel PWM synchronous output mode (SHIFT MODE= 1) where the phase difference between each channel are configurable

12.9.3 Operation

The basic programming flow is described as follows (takes PWM0 as an example):

1. Calculates the clock cycle counts for PWM waveform high/low-level period according to the selected clock source
2. Write the period counter value into registers HLPERIOD0、PERIOD0
3. Set register bit PWMMODE to 0 to operate at continuous mode. The PWM0 will start generating after register bit PWMSTART[0] is set to 1 and will stop until PWMSTART[0] is set to 0.
4. Set register bit PWMMODE to 1 to operate at fixed pulse count mode. Specify the required PWM waveform to be generated to register PCOUNT0. The PWM0 will start generating after PWMSTART[0] is set to 1 and will stop automatically when the number of pulse is met. The status register PWMDONE turns from 0 to 1.

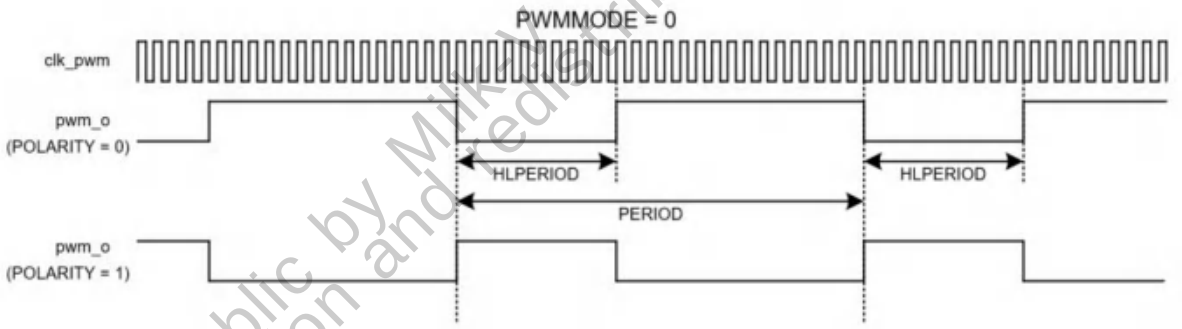


Figure 12- 41 PWM Continuous mode

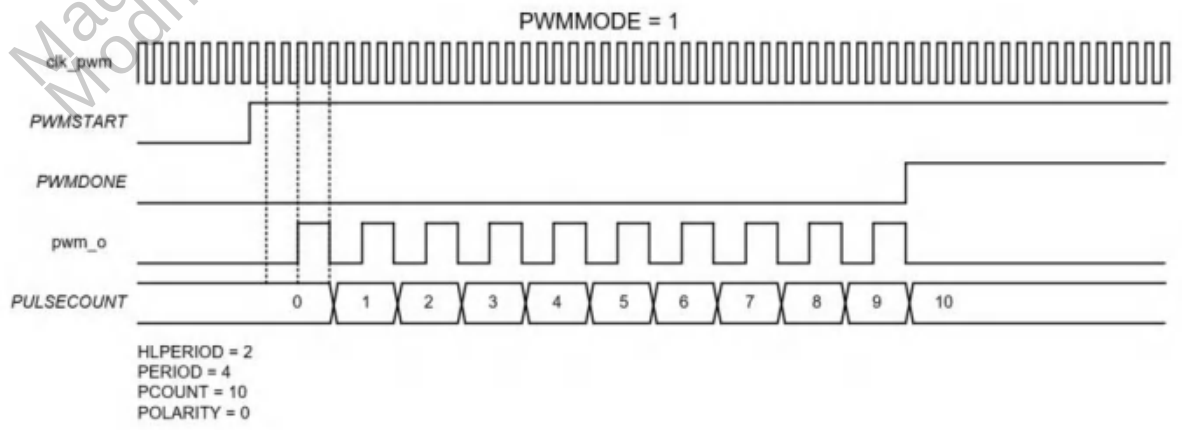


Figure 12- 42 PWM Pulse count mode

For example: to generate an 1MHz frequency square waveform which low level period percentage is 75%, and totally 16 pulses:

1. Use 100MHz clock source, the period count (PERIOD0) is 100MHz / 1MHz = 100, and the low-level count (HLPERIOD0) is 100 x 75% = 75. The pulse count (PCOUNT0) is 16.
2. Set PWMSTART[0] to 1 to start PWM.
3. Read register bit PWMDONE[0] until changes from 0 to 1.
4. The generated number of PWM pulses is stored in register PULSECOUNT0 and its value should equal to 16.

To enable PWM again, set PWMSTART[0] to 0 and then 1 to reset the counters and status registers.

Set SHIFTMODE to 1 can make 4-channel PWM operate at synchronous mode. The programming flow is described as follows:

1. Specify the four register pairs HLPERIOD0/PERIOD0, HLPERIOD1/PERIOD1, HLPERIOD2/PERIOD2, HLPERIOD3/PERIOD3 to the same value
2. Specify the phase shift cycle counts for the four waveform to registers SHIFTCOUNT0 to SHIFTCOUNT3.
3. Set PWMSTART[3:0] to 4'hF and then set register bit SHIFTSTART to 1. The period counters of four PWM will start counting at the same time, and the first rising edge of nth PWM channel depends on the value of SHIFTCOUNTn.

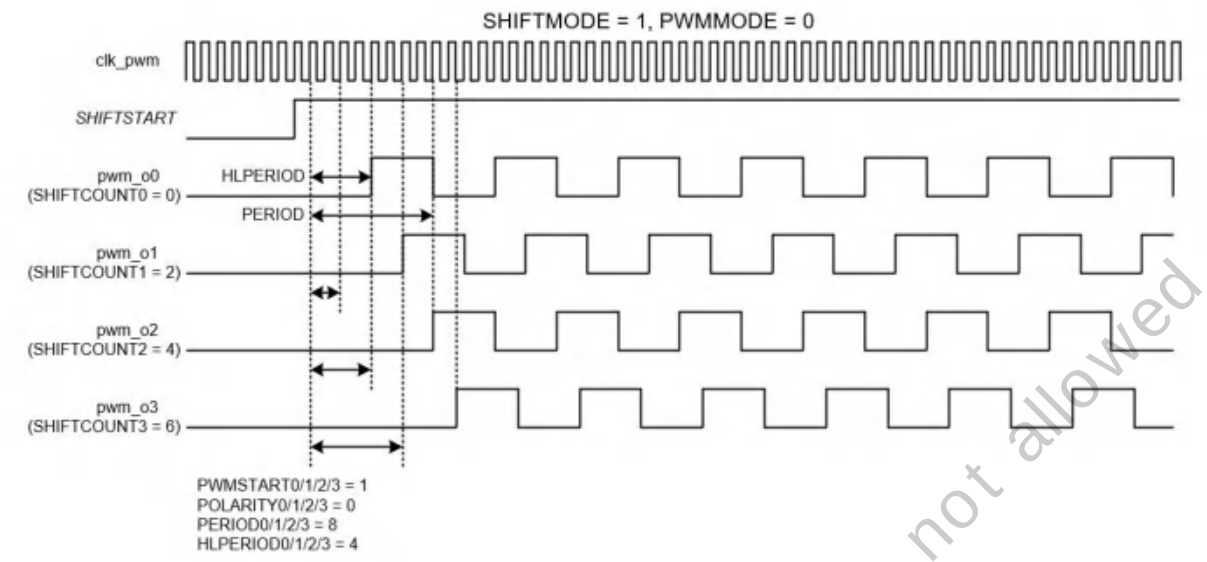


Figure 12- 43 PWM Continuous Shift Mode

For example: to simultaneously generate four channels 1KHz frequency square waveform which low level period percentage is 75% and each waveform is shifted by 1/4 cycle period sequencially:

1. Use 100MHz clock source, the period count is 100MHz / 1KHz = 100000, and the low-level count is 100000 x 75% = 75000.
2. Specify SHIFTCOUNT0 = 0; SHIFTCOUNT1= 100,000 x 1/4 = 25,000; SHIFTCOUNT2 = 100,000 x 2/4 = 50,000; SHIFTCOUNT3 = 100,000 x 3/4 = 75,000.
3. Set PWMSTART[3:0] to 4'hf and set SHIFTSTART to 1.

Set SHIFTSTART to 0 to stop PWM. Read register bits PWMDONE[3:0] until 4'hf indicates that four channels waveform are completed.

12.9.4 PWM Register Overview

PWM register overview is shown as：

错误!未找到引用源。.

12.9.5 PWM Register Description

HLPERIOD0

Offset Address: 0x000

Bits	Name	Access	Description	Reset
29:0	HLPERIOD0	R/W	PWM0 low level period counter value (unit is clk_pwm, value must > 0)	0x1

PERIOD0

Offset Address: 0x004

Bits	Name	Access	Description	Reset
29:0	PERIOD0	R/W	PWM0 period counter value (unit is clk_pwm, PERIOD must > 1 and must > HLPERIOD)	0x2

HLPERIOD1

Offset Address: 0x008

Bits	Name	Access	Description	Reset
29:0	HLPERIOD1	R/W	PWM1 low level period counter value (unit is clk_pwm, value must > 0)	0x1

PERIOD1

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
29:0	PERIOD1	R/W	PWM1 period counter value (unit is clk_pwm, PERIOD must > 1 and must > HLPERIOD)	0x2

HLPERIOD2

Offset Address: 0x010

Bits	Name	Access	Description	Reset
29:0	HLPERIOD2	R/W	PWM2 low level period counter value (unit is clk_pwm, value must > 0)	0x1

PERIOD2

Offset Address: 0x014

Bits	Name	Access	Description	Reset
29:0	PERIOD2	R/W	PWM2 period counter value (unit is clk_pwm, PERIOD must > 1 and must > HLPERIOD)	0x2

HLPERIOD3

Offset Address: 0x018

Bits	Name	Access	Description	Reset
29:0	HLPERIOD3	R/W	PWM3 low level period counter value (unit is clk_pwm, value must > 0)	0x1

PERIOD3

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
29:0	PERIOD3	R/W	PWM3 period counter value (unit is clk_pwm, PERIOD must > 1 and must > HLPERIOD)	0x2

POLARITY

Offset Address: 0x040

Bits	Name	Access	Description	Reset
3:0	POLARITY	R/W	Polarity of PWM0~3 [n] = 0: PWMn default low [n] = 1: PWMn default high	0x0
7:4	Reserved			
11:8	PWMMODE	R/W	PWM mode of PWM0~3 [n+8] = 0: PWMn is continuous mode [n+8] = 1: PWMn is pulse count mode	0x0
15:12	Reserved			
16	SHIFTMODE	R/W	PWM Phase shift mode 0 = PWM0~3 operate at normal mode 1 = PWM0~3 operate at Phase shift mode	0x0
19:17	Reserved			
20	pclk_force_en	R/W	pclk clock always enable 0 = enable clock auto gating while not access APB 1 = pclk keep on	0x0
31:21	Reserved			

PWMSTART

Offset Address: 0x044

Bits	Name	Access	Description	Reset
3:0	PWMSTART	R/W	Enable PWM0~3 [n] = 0: disable PWMn [n] = 1: enable PWMn If SHIFTMODE = 1, PWMSTART[3:0] used as the output enable of PWM0~3. The PWM is triggered by SHIFTSTART.	0x0
31:4	Reserved			

PWMDONE

Offset Address: 0x048

Bits	Name	Access	Description	Reset
3:0	PWMDONE	RO	PWM output done status for PWM0~3	

Bits	Name	Access	Description	Reset
			[n] = 1: PWMn finish output This bit will be cleared to 0 after PWMSTART[n] set to 0 then to 1.	
31:4	Reserved			

PWMUPDATE

Offset Address: 0x04c

Bits	Name	Access	Description	Reset
3:0	PWMUPDATE	R/W	Dynamic Update PWM counter value When PWMSTART set to 1, the register value (HLPERIODn, PERIODn) will be latched. In order to dynamic update PWM period, set this bit to 1 then to 0 to upload the new values of HLPERIODn and PERIOD.	0x0
31:4	Reserved			

PCOUNT0

Offset Address: 0x050

Bits	Name	Access	Description	Reset
23:0	PCOUNT0	R/W	PWM0 pulse count (value must > 0) Only valid when PWMMODE[0] = 1.	0x1
31:24	Reserved			

PCOUNT1

Offset Address: 0x054

Bits	Name	Access	Description	Reset
23:0	PCOUNT1	R/W	PWM1 pulse count (value must > 0) Only valid when PWMMODE[1] = 1.	0x1
31:24	Reserved			

PCOUNT2

Offset Address: 0x058

Bits	Name	Access	Description	Reset
23:0	PCOUNT2	R/W	PWM2 pulse count (value must > 0) Only valid when PWMMODE[2] = 1.	0x1
31:24	Reserved			

PCOUNT3

Offset Address: 0x05c

Bits	Name	Access	Description	Reset
23:0	PCOUNT3	R/W	PWM3 pulse count (value must > 0) Only valid when PWMMODE[3] = 1.	0x1
31:24				

PULSECOUNT0

Offset Address: 0x060

Bits	Name	Access	Description	Reset

Bits	Name	Access	Description	Reset
23:0	PULSECOUNT0	RO	PWM0 output pulse counter status	
31:24	Reserved			

PULSECOUNT1

Offset Address: 0x064

Bits	Name	Access	Description	Reset
23:0	PULSECOUNT1	RO	PWM1 output pulse counter status	
31:24	Reserved			

PULSECOUNT2

Offset Address: 0x068

Bits	Name	Access	Description	Reset
23:0	PULSECOUNT2	RO	PWM2 output pulse counter status	
31:24	Reserved			

PULSECOUNT3

Offset Address: 0x06c

Bits	Name	Access	Description	Reset
23:0	PULSECOUNT3	RO	PWM3 output pulse counter status	
31:24	Reserved			

SHIFTCOUNT0

Offset Address: 0x080

Bits	Name	Access	Description	Reset
23:0	SHIFTCOUNT0	R/W	PWM0 first pulse shift count Only valid when SHIFTMODE = 1	0x0
31:24	Reserved			

SHIFTCOUNT1

Offset Address: 0x084

Bits	Name	Access	Description	Reset
23:0	SHIFTCOUNT1	R/W	PWM1 first pulse shift count Only valid when SHIFTMODE = 1	0x0
31:24	Reserved			

SHIFTCOUNT2

Offset Address: 0x088

Bits	Name	Access	Description	Reset
23:0	SHIFTCOUNT2	R/W	PWM2 first pulse shift count Only valid when SHIFTMODE = 1	0x0
31:24	Reserved			

SHIFTCOUNT3

Offset Address: 0x08c

Bits	Name	Access	Description	Reset
23:0	SHIFTCOUNT3	R/W	PWM3 first pulse shift count	0x0

Bits	Name	Access	Description	Reset
			Only valid when SHIFTMODE = 1	
31:24	Reserved			

SHIFTSTART

Offset Address: 0x090

Bits	Name	Access	Description	Reset
0	SHIFTSTART	R/W	PWM start in Phase shift mode When SHIFTMODE = 1, set this bit to 1 to simultaneously start outputting PWM0~3.	0x0
31:1	Reserved			

PWM_OE

Offset Address: 0x0d0

Bits	Name	Access	Description	Reset
3:0	PWM_OE	R/W	PWM0~3 IO output enable 1 = output, 0 = input	0xF
31:4	Reserved			

12.10 Key scan

12.10.1 Overview

Keyscan supports a matrix of up to 8x8 = 64 keys. If you don't need so many keys, you can freely decide which rows or columns to mask or keep. You can select snapshot mode and FIFO mode to obtain key information according to software needs.

12.10.2 Working Method

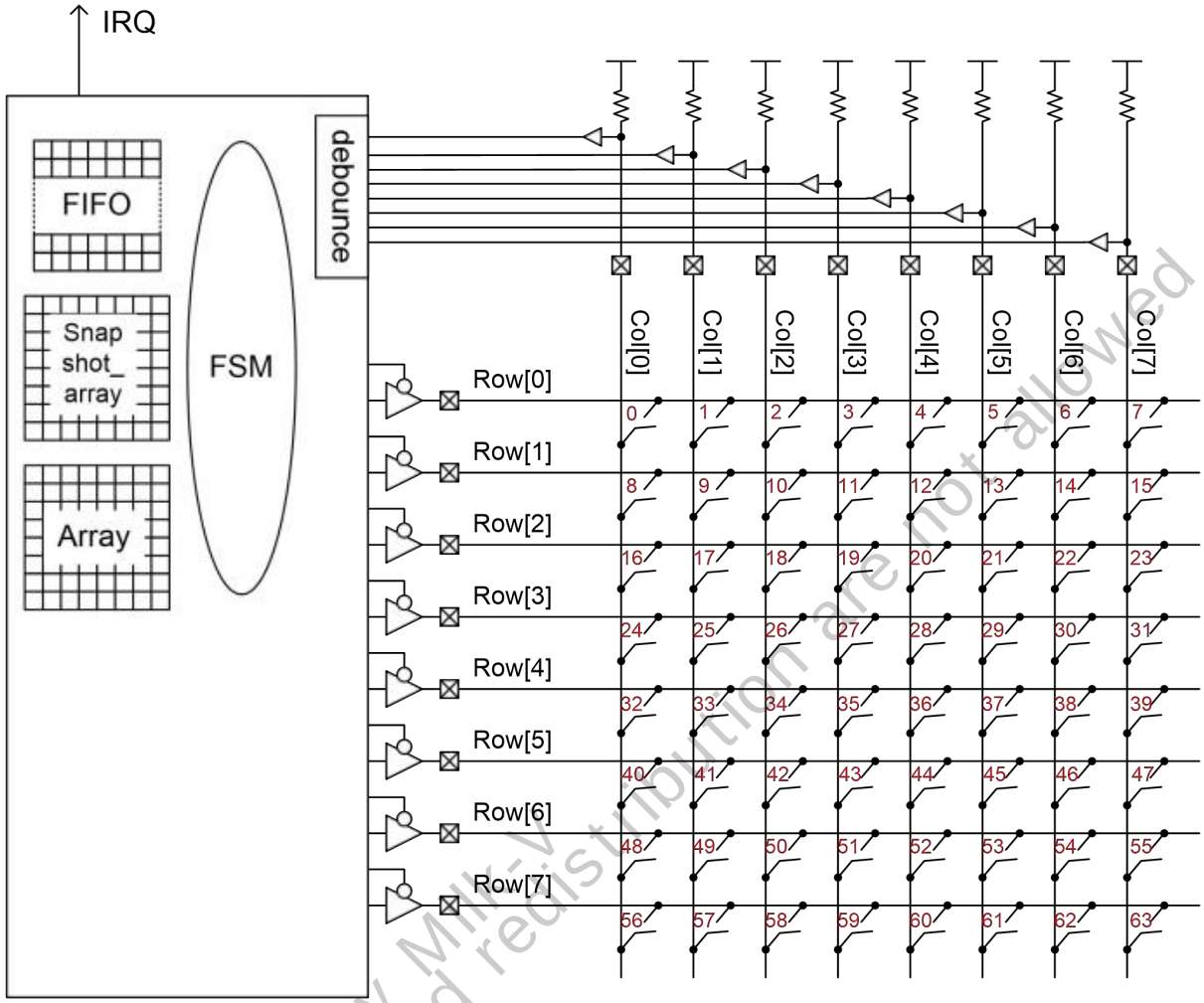


Figure 12- 44 Keyscan Architecture Block

When the state machine (FSM) is in the rest mode (no key is pressed), all rows output 0, while col is in the input mode and the weak pull-up is turned on (the weak pull-up is set in the register mapped to IOBLK, not in the keyscan module). When any key is pressed, col will see the value of not all 1 after debounce. It indicates that a key is pressed. At this time, FSM will start a scan to let row [0] - > row [7] have only one bit output 0 at a time (the rest are in HiZ high resistance state). Each result will be updated into an array

FSM will scan repeatedly until the col returned by all rows are all 1, indicating no key was pressed, and then it will go to rest mode (all rows output 0)

12.10.3 Basic Setting

The reg_row_mask, reg_col_mask and reg_enable in KEYSCAN0 means that when the matrix of 8x8 is not used, some IO can be selectively deleted without output or reference. Sine the default are all off, it neets to be turned on.

The reg_db_col in KEYSCAN_CONFIG2 determines how long debounce it takes for column input to be used.

The reg_slow_div in KEYSCAN_CONFIG1 determines the time of each stage in FSM of IP. Remember that this number must be larger than the debounce time. Otherwise, it will be wrongly interpret the IO state before debounce completion.

The reg_wait_cntr in KEYSCAN_CONFIG3 can be used to reduce the scanning speed. As long as the key is pressed, the keyscan module will scan continuously. This counter can control to wait for a certain time before starting a new round of scanning. Which result in lower scanning frequency.

12.10.4 Using FIFO mode

When FIFO mode is used, the 64 key values scanned by IP will be stored in the array. As long as the status of any key is different from that of the last scan, it will push the index of the key and the current value (0 / 1) into the FIFO. Therefore, the number in [5:0] specifies which key state change. [6] indicates whether it is pressed (0) or released (1). When the FIFO is not empty, IRQ will be issued. The advantage of this mode is to offload the software bit by bit to check which bit is changed. On the other hand, the disadvantage is that KEY_SCAN_FIFO is a register where read will pop FIFO automatically. Be careful when operating it.

.

12.10.5 Using snapshot array mode

When using snapshot array, the value of 64 keys currently scanned in IP will be stored in an array. If the content of the array is not consistent with KEYSCAN_SNAPSHOT_ARRAY , it will send IRQ. And the software can trigger KEYSCAN_SNAPSHOT_TRIG to capture the current array content to snapshot array, and then slowly compare what content has changed with the previous cognition.

Open the reg_irq_snapshot_change_enable in KEYSCAN_IRQ_ENABLE.
After receiving IRQ, read trigger KEYSCAN_SNAPSHOT_TRIG, interpret the content of KEYSCAN_SNAPSHOT_ARRAY, then clear KEY_SCAN_IRQ_CLEAR, and then finish IRQ return.

12.10.6 Key scan Register Overview

Name	Address Offset	Description
KEYSCAN_CONFIG0	0x000	
KEYSCAN_CONFIG1	0x004	
KEYSCAN_CONFIG2	0x008	
KEYSCAN_CONFIG3	0x00c	
KEYSCAN_SNAPSHOT_ARRAY	0x014	
KEYSCAN_SNAPSHOT_TRIG	0x01c	
KEYSCAN_FIFO_STATUS	0x020	
KEYSCAN_FIFO	0x024	
KEYSCAN_IRQ_ENABLE	0x028	
KEYSCAN_IRQ_FLAG	0x02c	
KEYSCAN_IRQ_CLEAR	0x030	

12.10.7 Key scan Register Description

KEYSCAN_CONFIG0

Offset Address: 0x000

Bits	Name	Access	Description	Reset
7:0	reg_row_mask	R/W	ROW[7:0] Mask 0 = enable 1 = disable	0xff

Bits	Name	Access	Description	Reset
15:8	reg_col_mask	R/W	COL[7:0] Mask 0 = enable 1 = disable	0xff
16	reg_enable	R/W	keyscan enable 0 = disable 1 = enable	0x0
31:17	Reserved			

KEYSCAN_CONFIG1

Offset Address: 0x004

Bits	Name	Access	Description	Reset
23:0	reg_slow_div	R/W	slow divider (MUST BE BIGGER THAN reg_db_col) Each step is IP clock frequency divide by reg_slow_div Scan frequency = IP clock freq / ((reg_slow_div+1) * (9+reg_wait_count+1)) IDLE -> ROW0 -> ROW1 -> ROW2->ROW3->ROW4->ROW5->ROW6->ROW7->UPDATE->WAIT->IDLE	0xff
31:24	Reserved			

KEYSCAN_CONFIG2

Offset Address: 0x008

Bits	Name	Access	Description	Reset
15:0	reg_db_col	R/W	column input debounce counter (IP clock cycle)	0x64
31:16	Reserved			

KEYSCAN_CONFIG3

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
7:0	reg_wait_cntr	R/W	wait interval between each scan (unit is reg_slow_div count)	0x10
31:8	Reserved			

KEYSCAN_SNAPSHOT_ARRAY

Offset Address: 0x014

Bits	Name	Access	Description	Reset
63:0	reg_cpu_snapshot_array	RO	CPU snapshot array result (0 = press, 1 = not press) [0] = Row 0 , Col 0 [1] = Row 0 , Col 1 … [7] = Row 0, Col 7 [8] = Row 1, Col 0 [63] = Row 7, Col 7 [N] = Row Y, Col X (N = Y*8 + X)	

KEYSCAN_SNAPSHOT_TRIG

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
0	reg_cpu_snapshot_toggle	W1T	Write 1 to Trigger snapshot array to update When current result is different from snapshot result, irq happen To solve the IRQ, write 1 to trigger the snapshot array to copy from current array and start checking which bit is different from previous state	
31:1	Reserved			

KEYSCAN_FIFO_STATUS

Offset Address: 0x020

Bits	Name	Access	Description	Reset
3:0	reg_fifo_count	RO	FIFO content count 0 = Empty 1 = one content in FIFO N = N content in FIFO	
4	reg_fifo_not_empty	RO	FIFO not empty flag 0 = Empty 1 = Not empty	
31:5	Reserved			

KEYSCAN_FIFO

Offset Address: 0x024

Bits	Name	Access	Description	Reset
6:0	reg_fifo_rdata	ROC	read data from FIFO (Auto POP) - check FIFO empty-ness before read [6] 0 = press, 1 = not-press [5:0] = index Row = INT(index/8) Col = mod(index,8) 63 = Row 7 , Column 7 13 = Row 1 , Clumne 5	
31:7	Reserved			

KEYSCAN_IRQ_ENABLE

Offset Address: 0x028

Bits	Name	Access	Description	Reset
0	reg_irq_fifo_not_empty_enable	R/W	FIFO mode IRQ Enable 0 = Disable 1 = Enable	0x0
3:1	Reserved			
4	reg_irq_snapshot_change_enable	R/W	Snapshot mode IRQ Enable 0 = Disable 1 = Enable	0x0
31:5	Reserved			

KEYSCAN_IRQ_FLAG

Offset Address: 0x02c

Bits	Name	Access	Description	Reset
0	reg_irq_fifo_not_empty	RO	FIFO not empty IRQ flag 0 = Empty 1 = Not empty	
3:1	Reserved			
4	reg_irq_snapshot_change	RO	Snapshot change IRQ flag 0 = No change 1 = Change	
31:5	Reserved			

KEYSCAN_IRQ_CLEAR

Offset Address: 0x030

Bits	Name	Access	Description	Reset
0	reg_irq_fifo_not_empty_clear_w1t	W1T	FIFO not empty IRQ Clear (Write 1 clear)	
3:1	Reserved			
4	reg_irq_snapshot_change_clear_w1t	W1T	Snapshot Change IRQ Clear (Write 1 clear)	
31:5	Reserved			

12.11 Wiegand

12.11.1 Overview

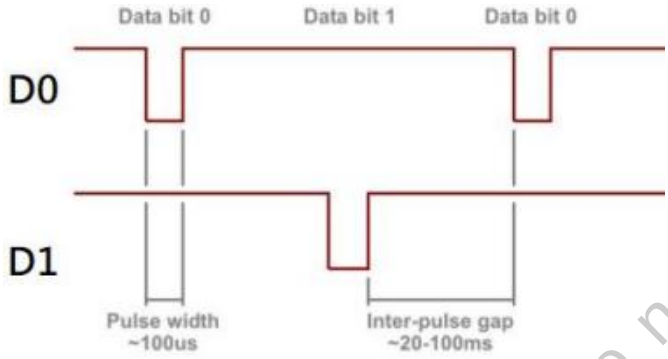


Figure 12- 45 the way wiegand signal bus transmits 0 / 1

The Wiegand interface uses two single ended signals, d0 / D1. When the bus idle both are high, a low pulse appears on D0, indicating that a "0" is transmitted. When a low pulse is found on D1, it means that a "1" is transmitted.

Wiegand is commonly used in access control system. There are two common formats, Wiegand 26 / 34, which represent the bit number of packets respectively. The brief introduction of these two formats is as follows.

12.11.1.1 Wiegand 26

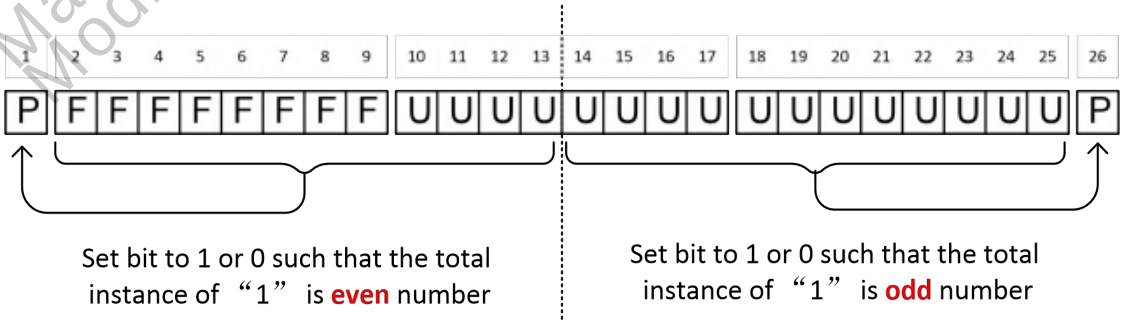


Figure 12- 9 Wiegand 26 Format

12.11.1.2 Wiegand 34

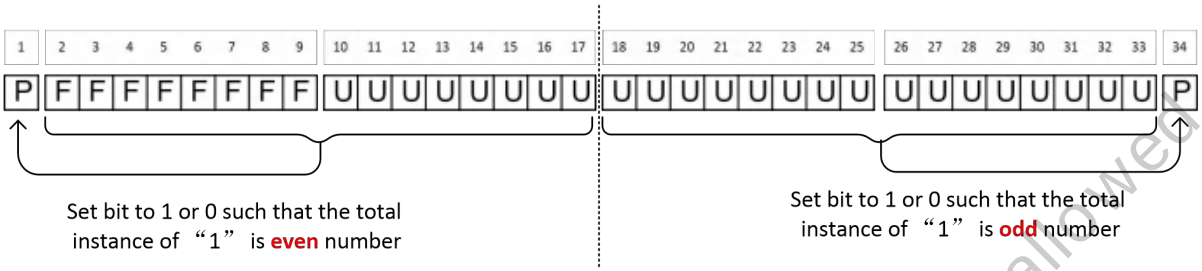


Figure 12- 47 Wiegand 34 format

F = Facility Code
U = User code

Some access card has a series of numbers on the back. After converting them to hex,
Dec 0002262506 Hex : 22_85_EA
0x22 is its Facility code (Dec = 34)
0x85EA is itsuser code (Dec = 34282)

Figure 12- 48 Common access card, the meaning those number

34, 34282 are facility code & user code in decimal system

PS. this IP TX RX does not handle the insertion or checking of parity. It is handled by software.

12.11.2 Working method

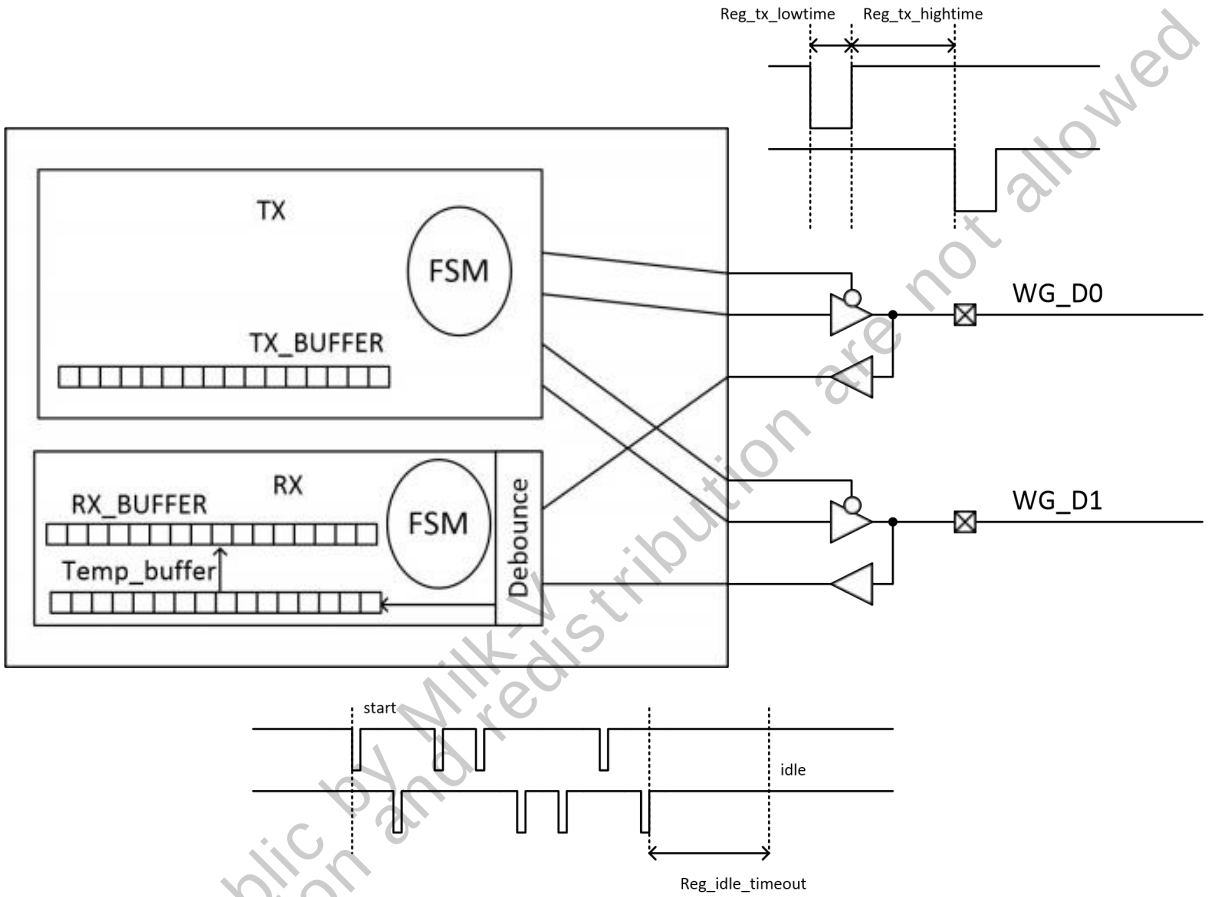


Figure 12- 49 wiegand architecture block

The Wiegand module contains TX and Rx, which can be used for single direction or bi-diretion used. During Transmitting, the RX will stop monitoring the bus to avoid received the message of its own tramission. Transmitter support push pull mode or open drain mode

12.11.2.1 TX

Before Transmit, set the high time and low time of TX, and transmit sequence is MSB 1st or LSB 1st. Then put the data in TX_BUFFER register, use TX_Trig, to start the transfer of the data.

After the transmission is complete, one can use TX_FINISH interrupt or by polling TX_BUSY status to determine when can transfer another packet.

12.11.2.2 RX

Before received, set the debounce time and the number of bits expected to receive in a packet. When low pulse appears in D0 D1, RX will start to push data into the temp buffer. When the number of received bits reaches the expected number of bits of a packet, it will push the temp buffer to RX_ BUFFER and sent out interrupt for software processing, while temp buffer continues to receive the next data.

If idle timeout occurs on D0 D1, even if the number of bits is not reached, it will be forced to be regarded as a packet

The high bit element of RX BUFFER will record the total number of bits received by this packet, whether it is caused by timeout or overflow before the software reads it

Every packet received can rely on rx_ buffer_ reciveived interrupt, or RX_ BUFFER_ VALID to determine is there any valid data in RX_BUFFER. After taking RX Data, trigger RX_ BUFFER_ CLEAR to clear RX_ BUFFER to receive the next packet.

12.11.3 Wiegand Register Overview

Name	Address Offset	Description
TX_CONFIG0	0x000	

Name	Address Offset	Description
TX_CONFIG1	0x004	
TX_CONFIG2	0x008	
TX_BUFFER	0x00c	
TX_TRIG	0x014	
TX_BUSY	0x018	
TX_DEBUG	0x01c	
RX_CONFIG0	0x020	
RX_CONFIG1	0x024	
RX_CONFIG2	0x028	
RX_BUFFER	0x02c	
RX_BUFFER_VALID	0x038	
RX_BUFFER_CLEAR	0x03c	
RX_DEBUG	0x040	
IRQ_ENABLE	0x044	
IRQ_FLAG	0x048	
IRQ_CLEAR	0x04c	

12.11.4 Wiegand Register Description

TX_CONFIG0

Offset Address: 0x000

Bits	Name	Access	Description	Reset
23:0	reg_tx_lowtime	R/W	TX Low width , unit = cycle	0xff
31:24	Reserved			

TX_CONFIG1

Offset Address: 0x004

Bits	Name	Access	Description	Reset
23:0	reg_tx_hightime	R/W	TX High width , unit = cycle	0xff
31:24	Reserved			

TX_CONFIG2

Offset Address: 0x008

Bits	Name	Access	Description	Reset
6:0	reg_tx_bitcount	R/W	TX Frame bit count per transmit , unit = bit	0x18
7	Reserved			
8	reg_tx_msb1st	R/W	TX Transmit from MSB or LSB 0 : LSB 1st , from tx_buffer[0] --> tx_buffer[reg_tx_bitcount]	0x0

Bits	Name	Access	Description	Reset
			1 : MSB 1st , from tx_buffer[reg_tx_bitcount] --> tx_buffer[0]	
15:9	Reserved			
16	reg_tx_opendrain	R/W	TX using push-pull mode or opendrain mode 0 : push-pull mode 1 : opendrain mode	0x0
31:17	Reserved			

TX_BUFFER

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
63:0	reg_tx_buffer	R/W	TX buffer content	0x00

TX_TRIG

Offset Address: 0x014

Bits	Name	Access	Description	Reset
0	reg_tx_trig_w1t	W1T	Trigger transmittion Write 1 trigger (please check reg_tx_busy before transmit)	
31:1	Reserved			

TX_BUSY

Offset Address: 0x018

Bits	Name	Access	Description	Reset
0	reg_tx_busy	RO	0 = idle, allow to trigger transmission 1 = busy, do not trigger any more tranmission or it will be ignore.	
31:1	Reserved			

TX_DEBUG

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
2:0	reg_tx_fsm	RO	TX Finite State Machine current state 0 : idle 1 : wait bus idle 2 : tx_start 3 : transmit low 4 : transmit high 5 : tx_stop	
7:3	Reserved			
14:8	reg_tx_pointer	RO	TX pointer current position indicate how many bit is still not yet send	
31:15	Reserved			

RX_CONFIG0

Offset Address: 0x020

Bits	Name	Access	Description	Reset
15:0	reg_rx_debounce	R/W	RX input debounce time (unit is cycle)	0xff
31:16	Reserved			

RX_CONFIG1

Offset Address: 0x024

Bits	Name	Access	Description	Reset
31:0	reg_idle_timeout	R/W	Bus timeout cycle count When bus is idle for idle_timeout cycle, bus is expected to be back to idle If some bit has received but not yet accumulate to rx_bitcount, it will also treat as a complete packet.	0xfff

RX_CONFIG2

Offset Address: 0x028

Bits	Name	Access	Description	Reset
6:0	reg_rx_bitcount	R/W	RX Expected Frame bit count , unit = bit	0x18
7	Reserved			
8	reg_rx_msb1st	R/W	RX Received sequence 0 : LSB 1st, 1st data is put in reg_rx_buffer[0]->[1]->[2].... 1 : MSB 1st, 1st data is put in reg_rx_buffer[reg_rx_bitcount]->[0]	0x0
11:9	Reserved			
12	reg_rx_enable	R/W	RX Enable 0 : disable 1 : Enable	0x0
31:13	Reserved			

RX_BUFFER

Offset Address: 0x02c

Bits	Name	Access	Description	Reset
72:0	reg_rx_buffer	RO	RX Buffer [63:0] = fifo = Indicate received content [70:64] = fifo_bit_count = How many effective bit is in rx_buffer[63:0] [71] = idle_reach = This RX is terminate by bus idle timeout [72] = overflow = This RX just overwrite an un-read message	
95:73	Reserved			

RX_BUFFER_VALID

Offset Address: 0x038

Bits	Name	Access	Description	Reset
0	reg_rx_buffer_valid	RO	reg_rx_buffer validness 0 : not valid 1 : valid	
31:1	Reserved			

RX_BUFFER_CLEAR

Offset Address: 0x03c

Bits	Name	Access	Description	Reset
0	reg_rx_buffer_clear_w1t	W1T	reg_rx_buffer clear (write 1 clear)	
31:1	Reserved			

RX_DEBUG

Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	reg_businidle	RO	bus in idle indication 0 : bus is not in idle 1 : bus is in idle more than reg_rx_idle_timeout cycle	
31:1	Reserved			

IRQ_ENABLE

Offset Address: 0x044

Bits	Name	Access	Description	Reset
0	reg_irq_tx_finish_enable	R/W	TX Finish IRQ Enable (to inform all data has being transmit, ready for next) 0 : Disable 1 : Enable	0x0
3:1	Reserved			
4	reg_irq_rx_overflow_enable	R/W	RX Overflow IRQ Enable 0 : Disable 1 : Enable	0x0
7:5	Reserved			
8	reg_irq_rx_received_enable	R/W	RX Received IRQ Enable 0 : Disable 1 : Enable	0x0
31:9	Reserved			

IRQ_FLAG

Offset Address: 0x048

Bits	Name	Access	Description	Reset
0	reg_irq_tx_finish	RO	TX Finish IRQ Flag 0 : no IRQ 1 : IRQ (one tranmission has being completed)	
3:1	Reserved			
4	reg_irq_rx_overflow	RO	RX overflow IRQ Flag 0 : no IRQ 1 : IRQ (rx buffer is not pop and new data has overwrited)	
7:5	Reserved			
8	reg_irq_rx_received	RO	RX received IRQ Flag 0 : no IRQ 1 : RX buffer has new data	
31:9	Reserved			

IRQ_CLEAR

Offset Address: 0x04c

Bits	Name	Access	Description	Reset
0	reg_irq_tx_finish_clear_w1t	W1T	TX Finish IRQ Clear , Write 1 to clear reg_irq_tx_finish flag	
3:1	Reserved			
4	reg_irq_rx_overflow_clear_w1t	W1T	RX Overflow IRQ Clear . Write 1 to clear reg_irq_rx_overflow flag	
7:5	Reserved			
8	reg_irq_rx_received_clear_w1t	W1T	RX Received IRQ Clear . Write 1 to clear reg_irq_rx_received flag	
31:9	Reserved			

12.12 IRRX Infrared Interface

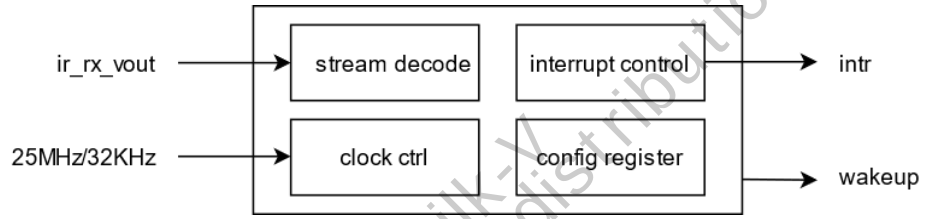
12.12.1 Overview

Receive external infrared data through IRRX Unit.

12.12.2 Characteristics

1. Support NEC encoding mode (including repeat code).
2. Support Philips RC5/RC6 encoding mode.
3. Support Sony encoding mode.
4. Support infrared wake-up function.

12.12.3 Working Mode



The software predefines the format of the received infrared data. When the IRRX module receives the infrared signal, it decodes it, and the encoded data that conforms to the predefined format is transmitted to the CPU via an interrupt. The CPU then performs corresponding operations based on the encoding.

12.12.4 IRRX Register Overview

Name	Address Offset	Description
IR_EN	0x000	
IR_MODE	0x004	
IR_CFG	0x008	
IR_FRAME	0x00c	
int_en	0x010	
int_clr	0x014	
int_msk	0x018	
int	0x01c	

Name	Address Offset	Description
int_raw	0x020	
IR_SYMBOL_CFG0	0x030	
IR_SYMBOL_CFG1	0x034	
IR_SYMBOL_CFG2	0x038	
IR_SYMBOL_CFG3	0x03c	
IR_SYMBOL_CFG4	0x040	
IR_SYMBOL_CFG5	0x044	
IR_SYMBOL_CFG6	0x048	
IR_SYMBOL_CFG7	0x04c	
IR_CLOCK_CTRL	0x050	
IR_DATA0	0x080	
IR_DATA1	0x084	
IR_DATA2	0x088	
IR_DATA3	0x08c	
IR_DATA4	0x090	
IR_NEC_DATA0	0x0a8	
IR_SONY_DATA0	0x0ac	
IR_SONY_DATA1	0x0b0	
IR_PHILIPS_DATA0	0x0b4	
IR_PHILIPS_DATA1	0x0b8	
IR_PRD_REC0	0x0e0	
IR_PRD_REC1	0x0e4	
IR_PRD_REC2	0x0e8	
IR_PRD_REC3	0x0ec	
IR_PRD_REC4	0x0f0	
IR_PRD_REC5	0x0f4	
SPARE_0	0xff0	
SPARE_1	0xff4	
SPARE_RO	0xff8	
DATA_CODE	0xffc	

12.12.5 IRRX Register Description

Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	reg_ir_rx_en	R/W	ir receiver enable	0x0
1	reg_ir_rx_rst	R/W	ir receiver reset	0x0
15:2	Reserved			
16	reg_ir_init_done	RO	ir receiver ready	
31:17	Reserved			

Offset Address: 0x004

Bits	Name	Access	Description	Reset
1:0	reg_ir_mode	R/W	ir receiver mode 0x0: pulse distance coding 0x2: bi-phase coding, RC5 0x3: bi-phase coding, RC6	0x0
7:2	Reserved			
8	reg_periodic_mode	R/W	periodic sample mode 0x1: periodic sample, reg_ic_mode is ignored	0x0
31:9	Reserved			

Offset Address: 0x008

Bits	Name	Access	Description	Reset
7:0	reg_tick_prd	R/W	tick period	0x18
11:8	reg_sample_prd	R/W	sample period	0x9
14:12	reg_debounce	R/W	input signal debounce control	0x4
15	Reserved			
16	reg_import_inv	R/W	input signal polarity control	0x1
17	reg_export_inv	R/W	output data polarity control	0x0
18	reg_sony_format	R/W	tx transmit stop burst at frame end 0x1: SONY formate 0x0: NEC formate	0x0

Bits	Name	Access	Description	Reset
19	reg_repeat_support	R/W	support simplfied repeat code	0x1
20	reg_bit_edge_sel	R/W	data bit selection used in pulse distance coding 0x1: SONY formate 0x0: NEC formate	0x0
31:21	Reserved			

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
7:0	reg_length	R/W	ir receiver data length	0x20
8	reg_slength	R/W	ir receiver repeate code length 0x1: TC9012 0x0: others	0x0
15:9	Reserved			
18:16	reg_lead_p_ratio	R/W	ratio of leading symbol period and strobe period, used in periodic sample mode	0x4
31:19	Reserved			

Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	reg_rx_done_int_en	R/W	rx_done interrupt enable	0x1
1	reg_frame_err_int_en	R/W	frame_err interrupt enable	0x1
2	reg_frame_ovf_int_en	R/W	frame_ovf interrupt enable	0x1
3	reg_release_int_en	R/W	release interrupt enable	0x1
4	reg_repeat_int_en	R/W	repeat interrupt enable	0x1
31:5	Reserved			

Offset Address: 0x014

Bits	Name	Access	Description	Reset
0	reg_rx_done_int_clr	W1T	rx_done interrupt clear	
1	reg_frame_err_int_clr	W1T	frame_err interrupt clear	
2	reg_frame_ovf_int_clr	W1T	frame_ovf interrupt clear	

Bits	Name	Access	Description	Reset
3	reg_release_int_clr	W1T	release interrupt clear	
4	reg_repeat_int_clr	W1T	repeat interrupt clear	
31:5	Reserved			

Offset Address: 0x018

Bits	Name	Access	Description	Reset
0	reg_rx_done_int_msk	R/W	rx_done interrupt mask	0x0
1	reg_frame_err_int_msk	R/W	frame_err interrupt mask	0x0
2	reg_frame_ovf_int_msk	R/W	frame_ovf interrupt mask	0x0
3	reg_release_int_msk	R/W	release interrupt mask	0x0
4	reg_repeat_int_msk	R/W	repeat interrupt mask	0x0
31:5	Reserved			

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
0	reg_rx_done_int	RO	rx_done interrpt	
1	reg_frame_err_int	RO	frame_err interrpt	
2	reg_frame_ovf_int	RO	frame_ovf interrpt	
3	reg_release_int	RO	release interrpt	
4	reg_repeat_int	RO	repeat interrpt	
31:5	Reserved			

Offset Address: 0x020

Bits	Name	Access	Description	Reset
0	reg_rx_done_int_raw	RO	rx_done interrupt raw value	
1	reg_frame_err_int_raw	RO	frame_err interrupt raw value	
2	reg_frame_ovf_int_raw	RO	frame_ovf interrupt raw value	
3	reg_release_int_raw	RO	release interrupt raw value	
4	reg_repeat_int_raw	RO	repeat interrupt raw value	
31:5	Reserved			

Offset Address: 0x030

Bits	Name	Access	Description	Reset

Bits	Name	Access	Description	Reset
11:0	reg_ir_rx_lead_p	R/W	lead symbol postive interval	0x383
15:12	Reserved			
23:16	reg_ir_rx_lead_p_tol	R/W	lead symbol postive interval tolerance reg_ir_rx_lead_p + reg_ir_rx_lead_p_tol <= 12'FFF reg_ir_rx_lead_p - reg_ir_rx_lead_p_tol >= 12'000	0x48
31:24	Reserved			

Offset Address: 0x034

Bits	Name	Access	Description	Reset
11:0	reg_ir_rx_lead_n	R/W	lead symbol negtive interval	0x1c1
15:12	Reserved			
23:16	reg_ir_rx_lead_n_tol	R/W	lead symbol negtive interval tolerance reg_ir_rx_lead_n + reg_ir_rx_lead_n_tol <= 12'FFF reg_ir_rx_lead_n - reg_ir_rx_lead_n_tol >= 12'000	0x24
31:24	Reserved			

Offset Address: 0x038

Bits	Name	Access	Description	Reset
7:0	reg_ir_rx_stop	R/W	stop symbol postive interval	0x37
11:8	reg_ir_rx_stop_tol	R/W	stop symbol postive interval tolerance reg_ir_rx_stop + reg_ir_rx_stop_tol <= 8'FF reg_ir_rx_stop - reg_ir_rx_stop_tol >= 8'00	0x3
15:12	Reserved			
23:16	reg_ir_rx_bit_p	R/W	data symbol postive interval	0x37
31:24	reg_ir_rx_bit_p_tol	R/W	data symbol postive interval	0x3

Bits	Name	Access	Description	Reset
			tolerance reg_ir_rx_bit_p + reg_ir_rx_bit_p_tol <= 8'FF reg_ir_rx_bit_p - reg_ir_rx_bit_p_tol >= 8'00	

Offset Address: 0x03c

Bits	Name	Access	Description	Reset
11:0	reg_ir_rx_bit_one	R/W	data one totol interval	0xe0
15:12	Reserved			
23:16	reg_ir_rx_bit_one_tol	R/W	data one total interval tolerance reg_ir_rx_bit_one + reg_ir_rx_bit_one_tol <= 12'FFF reg_ir_rx_bit_one - reg_ir_rx_bit_one_tol >= 12'000	0x18
31:24	Reserved			

Offset Address: 0x040

Bits	Name	Access	Description	Reset
11:0	reg_ir_rx_bit_zero	R/W	data zero totol interval	0x6f
15:12	Reserved			
23:16	reg_ir_rx_bit_zero_tol	R/W	data zero total interval tolerance reg_ir_rx_bit_zero + reg_ir_rx_bit_zero_tol <= 12'FFF reg_ir_rx_bit_zero - reg_ir_rx_bit_zero_tol >= 12'000	0x09
31:24	Reserved			

Offset Address: 0x044

Bits	Name	Access	Description	Reset
15:0	reg_ir_rx_release_time	R/W	time for wait repeat code, >108ms.	0x27de
31:16	Reserved			

Offset Address: 0x048

Bits	Name	Access	Description	Reset
11:0	reg_ir_rx_slead_p	R/W	repeate code lead symbol postive interval	0x383
15:12	Reserved			
23:16	reg_ir_rx_slead_p_tol	R/W	repeate code lead symbol postive interval tolerance reg_ir_rx_slead_p + reg_ir_rx_slead_p_tol <= 12'FFF reg_ir_rx_slead_p - reg_ir_rx_slead_p_tol >= 12'000	0x48
31:24	Reserved			

Offset Address: 0x04c

Bits	Name	Access	Description	Reset
11:0	reg_ir_rx_slead_n	R/W	repeate code lead symbol negtive interval	0x1c1
15:12	Reserved			
23:16	reg_ir_rx_slead_n_tol	R/W	repeate code lead symbol negtive interval tolerance reg_ir_rx_slead_n + reg_ir_rx_slead_n_tol <= 12'FFF reg_ir_rx_slead_n - reg_ir_rx_slead_n_tol >= 12'000	0x24
31:24	Reserved			

Offset Address: 0x050

Bits	Name	Access	Description	Reset
0	reg_pclock_auto_ctrl	R/W	pclk auto-gating control	0x0
1	reg_lpclock_switch_en	R/W	ip clock could be switch to 32KHz	0x0
7:2	Reserved			
8	reg_skip_lead_p	R/W	skip lead pulse check	0x0
31:9	Reserved			

Offset Address: 0x080

Bits	Name	Access	Description	Reset

Bits	Name	Access	Description	Reset
31:0	reg_ir_rx_data0	RO	recevier data[31:0]	

Offset Address: 0x084

Bits	Name	Access	Description	Reset
31:0	reg_ir_rx_data1	RO	recevier data[63:32]	

Offset Address: 0x088

Bits	Name	Access	Description	Reset
31:0	reg_ir_rx_data2	RO	recevier data[95:64]	

Offset Address: 0x08c

Bits	Name	Access	Description	Reset
31:0	reg_ir_rx_data3	RO	recevier data[127:96]	

Offset Address: 0x090

Bits	Name	Access	Description	Reset
31:0	reg_ir_rx_data4	RO	recevier data[159:128]	

Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
31:0	reg_ir_rx_nec_32bit	RO	receiver data, nec format	

Offset Address: 0x0ac

Bits	Name	Access	Description	Reset
11:0	reg_ir_rx_sony_12bit	RO	receiver data, sony D7C5 format	
15:12	Reserved			
30:16	reg_ir_rx_sony_15bit	RO	receiver data, sony D7C8 format	
31	Reserved			

Offset Address: 0x0b0

Bits	Name	Access	Description	Reset
19:0	reg_ir_rx_sony_20bit	RO	receiver data, sony D7C13 format	
31:20	Reserved			

Offset Address: 0x0b4

Bits	Name	Access	Description	Reset
11:0	reg_ir_rx_philips_rc5	RO	receiver data, RC5 format	
31:12	Reserved			

Offset Address: 0x0b8

Bits	Name	Access	Description	Reset
19:0	reg_ir_rx_philips_rc6	RO	receiver data, RC6 format	
31:20	Reserved			

Offset Address: 0x0e0

Bits	Name	Access	Description	Reset
11:0	reg_start_p0_min	RO	start phase0 minimul width	
15:12	Reserved			
27:16	reg_start_p0_max	RO	start phase0 maximul width	
31:28	Reserved			

Offset Address: 0x0e4

Bits	Name	Access	Description	Reset
11:0	reg_start_p1_min	RO	start phase1 minimul width	
15:12	Reserved			
27:16	reg_start_p1_max	RO	start phase1 maximul width	
31:28	Reserved			

Offset Address: 0x0e8

Bits	Name	Access	Description	Reset
11:0	reg_bit_p0_min	RO	bit phase0 minimul width	
15:12	Reserved			
27:16	reg_bit_p0_max	RO	bit phase0 maximul width	
31:28	Reserved			

Offset Address: 0x0ec

Bits	Name	Access	Description	Reset
11:0	reg_bit_p1_min	RO	bit phase1 minimul width	
15:12	Reserved			

Bits	Name	Access	Description	Reset
27:16	reg_bit_p1_max	RO	bit phase1 maximul width	
31:28	Reserved			

Offset Address: 0x0f0

Bits	Name	Access	Description	Reset
11:0	reg_end_min	RO	end phase minimuml width	
15:12	Reserved			
27:16	reg_end_max	RO	end phase maximuml width	
31:28	Reserved			

Offset Address: 0x0f4

Bits	Name	Access	Description	Reset
15:0	reg_frame_min	RO	frame minimul width	
31:16	reg_frame_max	RO	frame maximul width	

13 Security Subsystem Module

The chip provides independent security subsystem module which is responsible for providing specific security function.

The security subsystem module includes the following security function modules:

Crypto DMA

Secure Debug Protection

Crypto DMA provides hardware acceleration of symmetric key encryption, decryption and hardware acceleration of Hash. Secure eFuse unit is responsible for providing system security settings and secure keys for security subsystem.

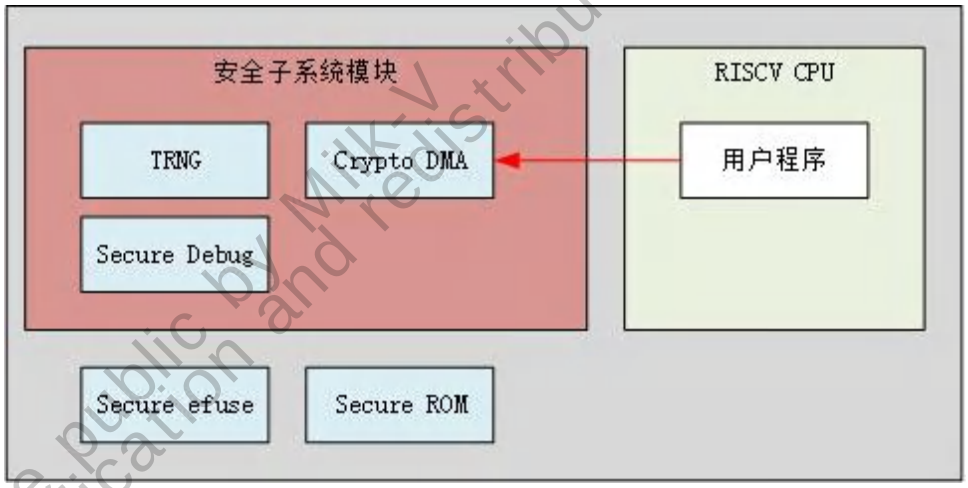


Figure 13- 1 Security Subsystem Module

13.1 CryptoDMA

13.1.1 Overview

CryptoDMA is a hardware accelerator for symmetric key algorithm, HASH algorithm and BASE64 conversion. It supports symmetric algorithm: AES 128 /

192 / 256, DES / TDES, SM4 and HASH algorithm: SHA-1 / SHA256. Through the instruction string of linked list, the function of key encryption and decryption or hash operation of data block can be directly accessed in memory.

The symmetric algorithm is suitable for hardware encryption and decryption of data, and supports a variety of block encryption and block series processing methods, including ECB, CBC and CTR.

The implementation of AES (Advanced Encryption Standard) algorithm conforms to FIPS 197 standard. The implementation of DES (data encryption standard) / TDES algorithm conforms to ISO / IEC 18033-3.

Hash algorithm is suitable for data integrity checking and digital signature operation acceleration. SHA1 and SHA256 meet FIPS180-2 standard.

13.1.2 Function Characteristics

The CryptoDMA module has the following features:

Support symmetric encryption and decryption algorithm AES and block encryption mode ECB / CBC / CTR. The key length supports 128 bits and 256 bits, and the key can be configured by secure operating system or linked list instruction.

Support symmetric encryption and decryption algorithm SM4 and block encryption mode ECB / CBC / CTR.

Support symmetric encryption and decryption algorithm DES / TDES and block encryption mode ECB / CBC / CTR.

Support hash algorithms SHA1 and SHA256.

Support CPU configuration to input PIO data and DMA mode to read active table instruction input data.

Support circular linked list structure, support splicing multiple linked list data.

Provide interrupt status query, interrupt mask and interrupt clear function.

13.1.3 DMA Function Description

CryptoDMA provides DMA function of direct memory access. The application program only needs to provide the linked list instruction to the target data block to start the CryptoDMA function until it receives the interrupt notification to

complete the block encryption and decryption or hash operation, and output the operation result to the target address.

13.1.4 Function description of symmetric key algorithm block encryption mode

Symmetric key algorithm AES / DES / SM4 all support ECC / CBC / CTR block encryption mode.

13.1.4.1 ECB Mode

In ECB (Electronic CodeBook) mode, encryption and decryption algorithms are directly applied to each packet data by the operation of each packet. This feature enables plaintext encryption and ciphertext decryption to be carried out independently by any group of block data.

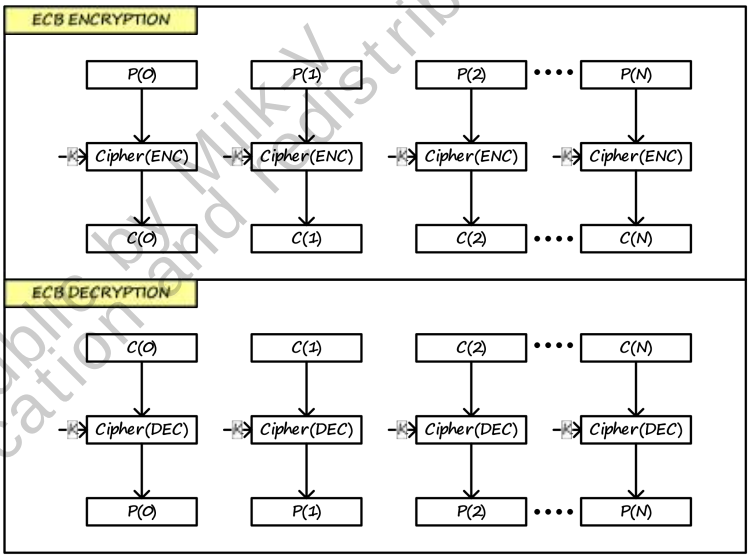


Figure 13- 2 ECB Mode

13.1.4.2 CBC Mode

CBC（Cipher Block Chaining）performs XOR operation between input plaintext group and input vector IV(Intialization Vector) or ciphertext result of previous

group before encryption operation. Encryption operation in CBC mode must start from the first block data group, and ciphertext obtained from previous group is required for subsequent encryption operation. During decryption, the plaintext can be obtained after decryption of the current ciphertext and XOR operation of the previous ciphertext.

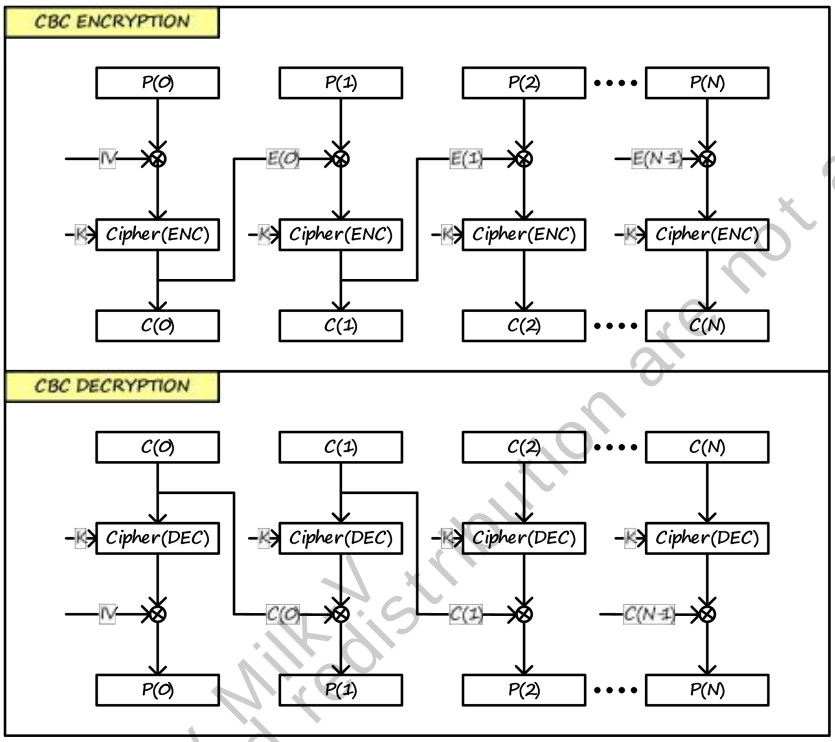


Figure 13- 3 CBC Mode

13.1.4.3 CTR Mode

CTR (Counter) is the use of encryption or decryption of a group of different sequences to ensure the independence and security of encrypted data processing. Generally, the XOR operation is carried out between encrypted accumulated sequence and plaintext.

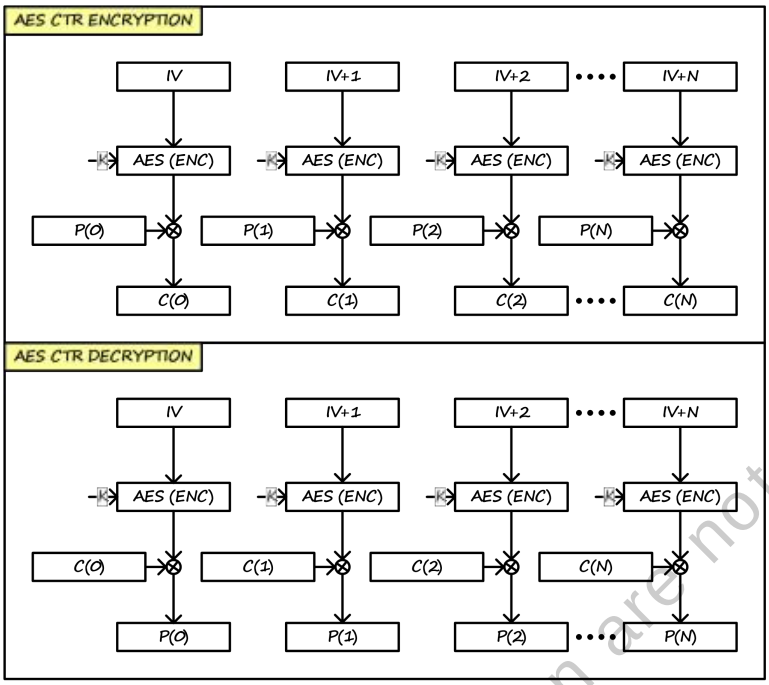


Figure 13- 4 CTR Mode

13.1.5 CryptoDMA Register Overview

Name	Address Offset	Description
dma_ctrl	0x000	DMA controll register
int_mask	0x004	interrupt mask
des_base_0	0x008	descriptor base low address
des_base_1	0x00c	descriptor base high address
spacc_int_raw	0x010	interrupt
secure_key_valid	0x014	key valid
des_addr_0	0x018	current descirptor low address
des_addr_1	0x01c	current descirptor high address
PIO_cmd_data_0	0x080	PIO command0
PIO_cmd_data_1	0x084	PIO command1
PIO_cmd_data_2	0x088	PIO command2
PIO_cmd_data_3	0x08c	PIO command3
PIO_cmd_data_4	0x090	PIO command4
PIO_cmd_data_5	0x094	PIO command5
PIO_cmd_data_6	0x098	PIO command6
PIO_cmd_data_7	0x09c	PIO command7
PIO_cmd_data_8	0x0a0	PIO command8
PIO_cmd_data_9	0x0a4	PIO command9
PIO_cmd_data_10	0x0a8	PIO command10
PIO_cmd_data_11	0x0ac	PIO command11

Name	Address Offset	Description
PIO_cmd_data_12	0x0b0	PIO command12
PIO_cmd_data_13	0x0b4	PIO command13
PIO_cmd_data_14	0x0b8	PIO command14
PIO_cmd_data_15	0x0bc	PIO command15
PIO_cmd_data_16	0x0c0	PIO command16
PIO_cmd_data_17	0x0c4	PIO command17
PIO_cmd_data_18	0x0c8	PIO command18
PIO_cmd_data_19	0x0cc	PIO command19
PIO_cmd_data_20	0x0d0	PIO command20
PIO_cmd_data_21	0x0d4	PIO command21
key_data_0	0x100	cipher key data 0
key_data_1	0x104	cipher key data 1
key_data_2	0x108	cipher key data 2
key_data_3	0x10c	cipher key data 3
key_data_4	0x110	cipher key data 4
key_data_5	0x114	cipher key data 5
key_data_6	0x118	cipher key data 6
key_data_7	0x11c	cipher key data 7
key_data_8	0x120	cipher key data 8
key_data_9	0x124	cipher key data 9
key_data_10	0x128	cipher key data 10
key_data_11	0x12c	cipher key data 11
key_data_12	0x130	cipher key data 12
key_data_13	0x134	cipher key data 13
key_data_14	0x138	cipher key data 14
key_data_15	0x13c	cipher key data 15
key_data_16	0x140	cipher key data 16
key_data_17	0x144	cipher key data 17
key_data_18	0x148	cipher key data 18
key_data_19	0x14c	cipher key data 19
key_data_20	0x150	cipher key data 20
key_data_21	0x154	cipher key data 21
key_data_22	0x158	cipher key data 22
key_data_23	0x15c	cipher key data 23
ini_data_0	0x180	initial vector data 0
ini_data_1	0x184	initial vector data 1
ini_data_2	0x188	initial vector data 2
ini_data_3	0x18c	initial vector data 3
ini_data_4	0x190	initial vector data 4
ini_data_5	0x194	initial vector data 5
ini_data_6	0x198	initial vector data 6
ini_data_7	0x19c	initial vector data 7
ini_data_8	0x1a0	initial vector data 8
ini_data_9	0x1a4	initial vector data 9
ini_data_10	0x1a8	initial vector data 10
ini_data_11	0x1ac	initial vector data 11
sha_data_0	0x1c0	SHA paramenter0
sha_data_1	0x1c4	SHA paramenter1
sha_data_2	0x1c8	SHA paramenter2
sha_data_3	0x1cc	SHA paramenter3
sha_data_4	0x1d0	SHA paramenter4
sha_data_5	0x1d4	SHA paramenter5
sha_data_6	0x1d8	SHA paramenter6
sha_data_7	0x1dc	SHA paramenter7

13.1.6 CryptoDMA Register Overview

(基址 0x02060000)

dma_ctrl

dma_ctrl

Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	dma_en	R/W	DMA Channel Enable Control 0：Channel Forbidden； 1：Channel Enable	0x0
1	decriptor_mode	R/W	Channel Command Mode 0: PIO Mode 1:Description Key List Mode	0x0
15:2	Reserved			
23:16	max_read_burst	R/W	Maximum Read Burst Value	0x0
31:24	max_write_burs	R/W	Maximum Write Burst Value	0x0

int_mask

int_mask

Offset Address: 0x004

Bits	Name	Access	Description	Reset
0	int_enc_mask	R/W	Encryption and decryption interrupt masking	0x0
1	int_hash_mask	R/W	Hash interrupt masking	0x0
31:2	Reserved		Reserved	

des_base_0

des_base_0

Offset Address: 0x008

Bits	Name	Access	Description	Reset
31:0	des_base_0	R/W	Description key table address_ Low address	0x0

des_base_1

des_base_1

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
31:0	des_base_1	R/W	Description key table address_ High address	0x0

spacc_int_raw

spacc_int_raw

Offset Address: 0x010

Bits	Name	Access	Description	Reset
31:0	spacc_int_raw	R/W	DMA Interrupted	0x0

secure_key_valid

secure_key_valid

Offset Address: 0x014

Bits	Name	Access	Description	Reset
31:0	secure_key_valid	R/W	Key Valid Value **One-Way:** Enabled	0x0

des_addr_0

des_addr_0

Offset Address: 0x018

Bits	Name	Access	Description	Reset
31:0	des_addr_0	R/W	Description key table address offset_ Low address	0x0

des_addr_1

des_addr_1

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
31:0	des_addr_1	R/W	Description key table address offset_ High address	0x0

PIO_cmd_data_0

PIO mode descriptor

Offset Address: 0x080

Bits	Name	Access	Description	Reset
0	PIO_cmd_data_0	R/W	PIO mode description key table, 0：invalid schema description key table 1：valid schema description key table	0x0
1	PIO_cmd_data_0_1	R/W	Last descriptor of description key table	0x0
2	PIO_cmd_data_0_2	R/W	PIO mode interrupt enable 0：interrupt and disable； 1：Interrupt and enable	0x0
3	PIO_cmd_data_0_3	R/W	Connection description key table address selection 0: Use the continuation address of the current address 1: Use the connection address register value	0x0
7:4	PIO_cmd_data_0_4	R/W	Reserved	0x0
8	PIO_cmd_data_0_8	R/W	DMA use bypass	0x0
9	PIO_cmd_data_0_9	R/W	Encryption and decryption using AES	0x0
10	PIO_cmd_data_0_10	R/W	Encryption and decryption usingDES	0x0
11	PIO_cmd_data_0_11	R/W	Reserved	0x0
12	PIO_cmd_data_0_12	R/W	Encryption and decryption usingSHA	0x0

Bits	Name	Access	Description	Reset
13	PIO_cmd_data_0_13	R/W	Reserved	0x0
15:14	Reserved			
19:16	PIO_cmd_data_0_16	R/W	Operation Key Selection AES / DES / SM4 is used: 1: Descriptor Key 2:Key2 4:Key1 8: Millan 0 Using sha 1: Descriptors 2:Reserved 4:Reserved 8: Using SHA parameter register The rest areReserved	0x0
23:20	PIO_cmd_data_0_20	R/W	First operation IV selection AES / DES / SM4 is used 1: Descriptor IV 2:IV2 4:IV1 8:IV0	0x0
26:24	PIO_cmd_data_0_24	R/W	Continued Operation IV Selection AES / DES / SM4 is used 1: Descriptor IV 2:IV2 4:IV1 8:IV0	0x0
27	PIO_cmd_data_0_27	R/W	Key selection enabling	0x0
31:28	PIO_cmd_data_0_28	R/W	Reserved	0x0

PIO_cmd_data_1

PIO mode descriptor

Offset Address: 0x084

Bits	Name	Access	Description	Reset
0	PIO_cmd_data_1_0	R/W	Select 1-encryption/0-decription for encryption and decryption Select 1-need parameter / 0-no need for hash parameter	0x0
1	PIO_cmd_data_1_1	R/W	CBC Mode 0-ECB/1-CBC	0x0
2	PIO_cmd_data_1_2	R/W	CTR Mode 1-CTR	0x0
5:3	PIO_cmd_data_1_3	R/W	Key Mode 100-128bit/010-192bit/001-256bit for aes 0-DES/1-TDES sha mode 0-SHA1/1-SHA256	0x0
31:6	PIO_cmd_data_1_6	R/W	Reserved	0x0

PIO_cmd_data_2

PIO mode descriptor

Offset Address: 0x088

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_2	R/W	Reserved	0x0

PIO_cmd_data_3

PIO mode descriptor

Offset Address: 0x08c

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_3	R/W	Continuous descriptor address_ Low address	0x0

PIO_cmd_data_4

PIO mode descriptor

Offset Address: 0x090

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_4	R/W	Continuous descriptor address_High address	0x0

PIO_cmd_data_5

PIO mode descriptor

Offset Address: 0x094

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_5	R/W	DMA source address_ Low address	0x0

PIO_cmd_data_6

PIO mode descriptor

Offset Address: 0x098

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_6	R/W	DMA source address_ High address	0x0

PIO_cmd_data_7

PIO mode descriptor

Offset Address: 0x09c

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_7	R/W	DMA target address_ Low address	0x0

PIO_cmd_data_8

PIO mode descriptor

Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_8	R/W	DMA target address_ High address	0x0

PIO_cmd_data_9

PIO mode descriptor

Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_9	R/W	SHA Information size	0x0

PIO_cmd_data_10

PIO mode descriptor

Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_10	R/W	Reserved	0x0

PIO_cmd_data_11

PIO mode descriptor

Offset Address: 0x0ac

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_11	R/W	BASE64 Target Information size	0x0

PIO_cmd_data_12

PIO mode descriptor

Offset Address: 0x0b0

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_12	R/W	Reserved	0x0

PIO_cmd_data_13

PIO mode descriptor

Offset Address: 0x0b4

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_13	R/W	Reserved	0x0

PIO_cmd_data_14

PIO mode descriptor

Offset Address: 0x0b8

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_14	R/W	Reserved	0x0

PIO_cmd_data_15

PIO mode descriptor

Offset Address: 0x0bc

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_15	R/W	Reserved	0x0

PIO_cmd_data_16

PIO mode descriptor

Offset Address: 0x0c0

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_16	R/W	Reserved	0x0

PIO_cmd_data_17

PIO mode descriptor

Offset Address: 0x0c4

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_17	R/W	Reserved	0x0

PIO_cmd_data_18

PIO mode descriptor

Offset Address: 0x0c8

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_18	R/W	Reserved	0x0

PIO_cmd_data_19

PIO mode descriptor

Offset Address: 0x0cc

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_19	R/W	Reserved	0x0

PIO_cmd_data_20

PIO mode descriptor

Offset Address: 0x0d0

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_20	R/W	Reserved	0x0

PIO_cmd_data_21

PIO mode descriptor

Offset Address: 0x0d4

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_21	R/W	Reserved	0x0

key_data_0

3key

Offset Address: 0x100

Bits	Name	Access	Description	Reset
31:0	key_data_0	RO	Key	

key_data_1

key

Offset Address: 0x104

Bits	Name	Access	Description	Reset
31:0	key_data_1	RO	Key	

key_data_2

key

Offset Address: 0x108

Bits	Name	Access	Description	Reset
31:0	key_data_2	RO	Key	

key_data_3

key

Offset Address: 0x10c

Bits	Name	Access	Description	Reset
31:0	key_data_3	RO	Key	

key_data_4

key

Offset Address: 0x110

Bits	Name	Access	Description	Reset
31:0	key_data_4	RO	Key	

key_data_5

key

Offset Address: 0x114

Bits	Name	Access	Description	Reset
31:0	key_data_5	RO	Key	

key_data_6

key

Offset Address: 0x118

Bits	Name	Access	Description	Reset
31:0	key_data_6	RO	Key	

key_data_7

key

Offset Address: 0x11c

Bits	Name	Access	Description	Reset
31:0	key_data_7	RO	Key	

key_data_8

key

Offset Address: 0x120

Bits	Name	Access	Description	Reset
31:0	key_data_8	RO	Key	

key_data_9

key

Offset Address: 0x124

Bits	Name	Access	Description	Reset
31:0	key_data_9	RO	Key	

key_data_10

key

Offset Address: 0x128

Bits	Name	Access	Description	Reset

Bits	Name	Access	Description	Reset
31:0	key_data_10	RO	Key	

key_data_11

key

Offset Address: 0x12c

Bits	Name	Access	Description	Reset
31:0	key_data_11	RO	Key	

key_data_12

key

Offset Address: 0x130

Bits	Name	Access	Description	Reset
31:0	key_data_12	RO	Key	

key_data_13

key

Offset Address: 0x134

Bits	Name	Access	Description	Reset
31:0	key_data_13	RO	Key	

key_data_14

key

Offset Address: 0x138

Bits	Name	Access	Description	Reset
31:0	key_data_14	RO	Key	

key_data_15

key

Offset Address: 0x13c

Bits	Name	Access	Description	Reset
31:0	key_data_15	RO	Key	

key_data_16

key

Offset Address: 0x140

Bits	Name	Access	Description	Reset
31:0	key_data_16	RO	Key	

key_data_17

key

Offset Address: 0x144

Bits	Name	Access	Description	Reset
31:0	key_data_17	RO	Key	

key_data_18

key

Offset Address: 0x148

Bits	Name	Access	Description	Reset
31:0	key_data_18	RO	Key	

key_data_19

key

Offset Address: 0x14c

Bits	Name	Access	Description	Reset
31:0	key_data_19	RO	Key	

key_data_20

key

Offset Address: 0x150

Bits	Name	Access	Description	Reset
31:0	key_data_20	RO	Key	

key_data_21

key

Offset Address: 0x154

Bits	Name	Access	Description	Reset
31:0	key_data_21	RO	Key	

key_data_22

key

Offset Address: 0x158

Bits	Name	Access	Description	Reset
31:0	key_data_22	RO	Key	

key_data_23

key

Offset Address: 0x15c

Bits	Name	Access	Description	Reset
31:0	key_data_23	RO	Key	

ini_data_0

3iv

Offset Address: 0x180

Bits	Name	Access	Description	Reset
31:0	ini_data_0	RO	Initial parameters	

ini_data_1

iv

Offset Address: 0x184

Bits	Name	Access	Description	Reset
31:0	ini_data_1	RO	Initial parameters	

ini_data_2

iv

Offset Address: 0x188

Bits	Name	Access	Description	Reset
31:0	ini_data_2	RO	Initial parameters	

ini_data_3

iv

Offset Address: 0x18c

Bits	Name	Access	Description	Reset
31:0	ini_data_3	RO	Initial parameters	

ini_data_4

iv

Offset Address: 0x190

Bits	Name	Access	Description	Reset
31:0	ini_data_4	RO	Initial parameters	

ini_data_5

iv

Offset Address: 0x194

Bits	Name	Access	Description	Reset
31:0	ini_data_5	RO	Initial parameters	

ini_data_6

iv

Offset Address: 0x198

Bits	Name	Access	Description	Reset
31:0	ini_data_6	RO	Initial parameters	

ini_data_7

iv

Offset Address: 0x19c

Bits	Name	Access	Description	Reset
31:0	ini_data_7	RO	Initial parameters	

ini_data_8

iv

Offset Address: 0x1a0

Bits	Name	Access	Description	Reset
1:0	ini_data_8	RO	Initial parameters	
31:2	Reserved			

ini_data_9

iv

Offset Address: 0x1a4

Bits	Name	Access	Description	Reset
1:0	ini_data_9	RO	Initial parameters	
31:2	Reserved			

ini_data_10

iv

Offset Address: 0x1a8

Bits	Name	Access	Description	Reset
31:0	ini_data_10	RO	Initial parameters	

ini_data_11

iv

Offset Address: 0x1ac

Bits	Name	Access	Description	Reset
31:0	ini_data_11	RO	Initial parameters	

sha_data_0

sha parameter

Offset Address: 0x1c0

Bits	Name	Access	Description	Reset
31:0	sha_data_0	RO	SHA Parameters	

sha_data_1

sha parameter

Offset Address: 0x1c4

Bits	Name	Access	Description	Reset
31:0	sha_data_1	RO	SHA Parameters	

sha_data_2

sha parameter

Offset Address: 0x1c8

Bits	Name	Access	Description	Reset
31:0	sha_data_2	RO	SHA Parameters	

sha_data_3

sha parameter

Offset Address: 0x1cc

Bits	Name	Access	Description	Reset
31:0	sha_data_3	RO	SHA Parameters	

sha_data_4

sha parameter

Offset Address: 0x1d0

Bits	Name	Access	Description	Reset
31:0	sha_data_4	RO	SHA Parameters	

sha_data_5

sha parameter

Offset Address: 0x1d4

Bits	Name	Access	Description	Reset
31:0	sha_data_5	RO	SHA Parameters	

sha_data_6

sha parameter

Offset Address: 0x1d8

Bits	Name	Access	Description	Reset
31:0	sha_data_6	RO	SHA Parameters	

sha_data_7

sha parameter

Offset Address: 0x1dc

Bits	Name	Access	Description	Reset
31:0	sha_data_7	RO	SHA Parameters	

13.2 Secure Debug Firewall

In order to read or control the internal functions of the chip during debugging or testing, the chip provides several debugging interfaces, such as JTAG, I2C and other external interfaces. Without proper protection mechanism, these interfaces can easily be used to attack chip security mechanism or read internal confidential information directly or indirectly. In order to protect and control these interfaces, the chip adopts a secure debugging firewall.

13.2.1 Overview

The chip supports three main debugging interfaces:

1. RISCV JTAG: RISCV processors have a built-in debug interface that allows users to access internal registers through the JTAG interface.
2. I2C: The chip provides debug interface
3. Test interface: The chip provides special debugging interface for production test

For JTAG/I2C interfaces, a secure debugging firewall provides specific protection controls for their access.

For the Test Interface, a secure debugging interface provides a separate class of protection controls for its access (Test Access).

For these debugging categories, the secure debugging firewall provides three connection control states:

Open: allow external access without additional control
Protected: no external connection is allowed until the corresponding password is input through I2C interface
Closed: no external connection is allowed through this interface, and it cannot be opened again by other methods.

13.2.2 Status Inquiry and Password Input Interface (I2C)

The secure debug firewall provides an independent I2C interface for the external chip to check the current status of the debug interface through I2C and input the corresponding password to restart the protected interface. External users need to specify the I2C ID of the main blockhouse to connect to the firewall interface.

The I2C interface register address is as follows:

Table 13- 1 Status Query I2C Interface Register Address

byte address	bitwidth	signal name	description
0	32	i2c_REE_password [31:0]	128-bit debug interface password
4	32	i2c_REE_password [63:32]	128-bit debug interface password
8	32	i2c_REE_password [95:64]	128-bit debug interface password
C	32	i2c_REE_password [127:96]	128-bit debug interface password
10	32	Reserved	Reserved
14	32	Reserved	Reserved
18	32	Reserved	Reserved
1C	32	Reserved	Reserved
20	32	Reserved	Reserved
24	32	Reserved	Reserved
28	32	Reserved	Reserved
2C	32	Reserved	Reserved
30	32	i2c_TST_password [31:0]	128-bit password entry field for test ports
34	32	i2c_TST_password [63:32]	128-bit test debug interface password
38	32	i2c_TST_password [95:64]	128-bit test debug interface password
3C	32	i2c_TST_password [127:96]	128-bit test debug interface password
40	1	REE_PW_update	Update the password compare result
44	1	Reserved	Reserved
48	1	Reserved	Reserved
4C	1	TST_PW_update	Update the compare result of test interface password
80	32	Chip_UID0 (LSB of ID)	Chip identification number (Device ID)
84	32	Chip_UID1 (MSB of ID)	Chip identification number (Device ID)
88	32	MKTSEG	Market segmentation number
8C	32	DBG_MODE	The status of the chip debugging interface protection settings [1:0] : Interface protection mode， 0:open;1:protected:2/3:closed [3:2] : Reserved [5:4] : Reserved [7:6] : test interface protection mode， 0:open;1:protected:2/3:closed [8]: Use HASH to make comparison
90	32	reserved	Reserved
94	4	DBG_PROT_STATUS	The current debug protection status [0]: the interface ptotection status，0:open 1:closed [1]: Reserved [2]: Reserved [3]::the test interface ptotection status，0:open 1:closed

13.2.3 Status Inquiry and Password Input Process

13.2.3.1 Status Inquiry Process

(step 1) control external I2C to send start signal
(step 2) I2C sends firewall I2C ID (default 0x56)
(step 3) the I2C reads the address 0x04001A94 to obtain the current protection status of the debugging interface

13.2.3.2 Password Input Process

(step 1) control external I2C to send start signal
(step 2) I2C sends the I2C ID of the debugging interface firewall (default: 0x56)
(step 3) I2C reads the address of 0x04001A80 / 0x04001A84 to get the Device ID, and reads the address of 0x04001A88 to get the market distinguishing number
(step 4) prepare the unlocking password corresponding to each category through the equipment serial number and market distinguishing number
(step 5) I2C reads the address 0x94 to get the current protection status of the debugging interface, and the blockhouse confirms whether it is locked or not
(step 6) take the non secure debug interface as an example, write the non secure password to the address of 0x04001A00 / 0x04001A04 / 0x04001A08 / 0x04001A0c by I2C
(step 7) I2C writes any value to the address 0x04001A10 and updates the password comparison value
(step 5) I2C reads the address 0x94 to get the current protection status of the debugging interface, and confirms whether it is unlock

13.3 Efuse Controller

13.3.1 Overview

4Kbit eFuse space is integrated in the chip, and eFuse is programmed and read by eFuse Ctrl.
The main functions of eFuse Ctrl include：
A double bit protection mechanism is provided, which consists of two entity eFuse bits to form a single bit logical effective value. It is equivalent to providing 2Kbit memory space to improve the robustness of eFuse burning or data maintenance

After power-on-reset, the eFuse content is automatically loaded into the register to provide the configuration settings required by the chip system, reduce the number of reading eFuse and improve the service life

Provide eFuse programming, read, verify read and power on / off instructions and content security protection mechanism.

The eFuse data register is divided into two areas, one is a non secure area, and the other is a secure area. The data in the non secure area is accessible to all modules, while the secure area is accessible only to the secure module. The non secure area stores system configuration and public information, and the secure area stores security configuration, key and password.

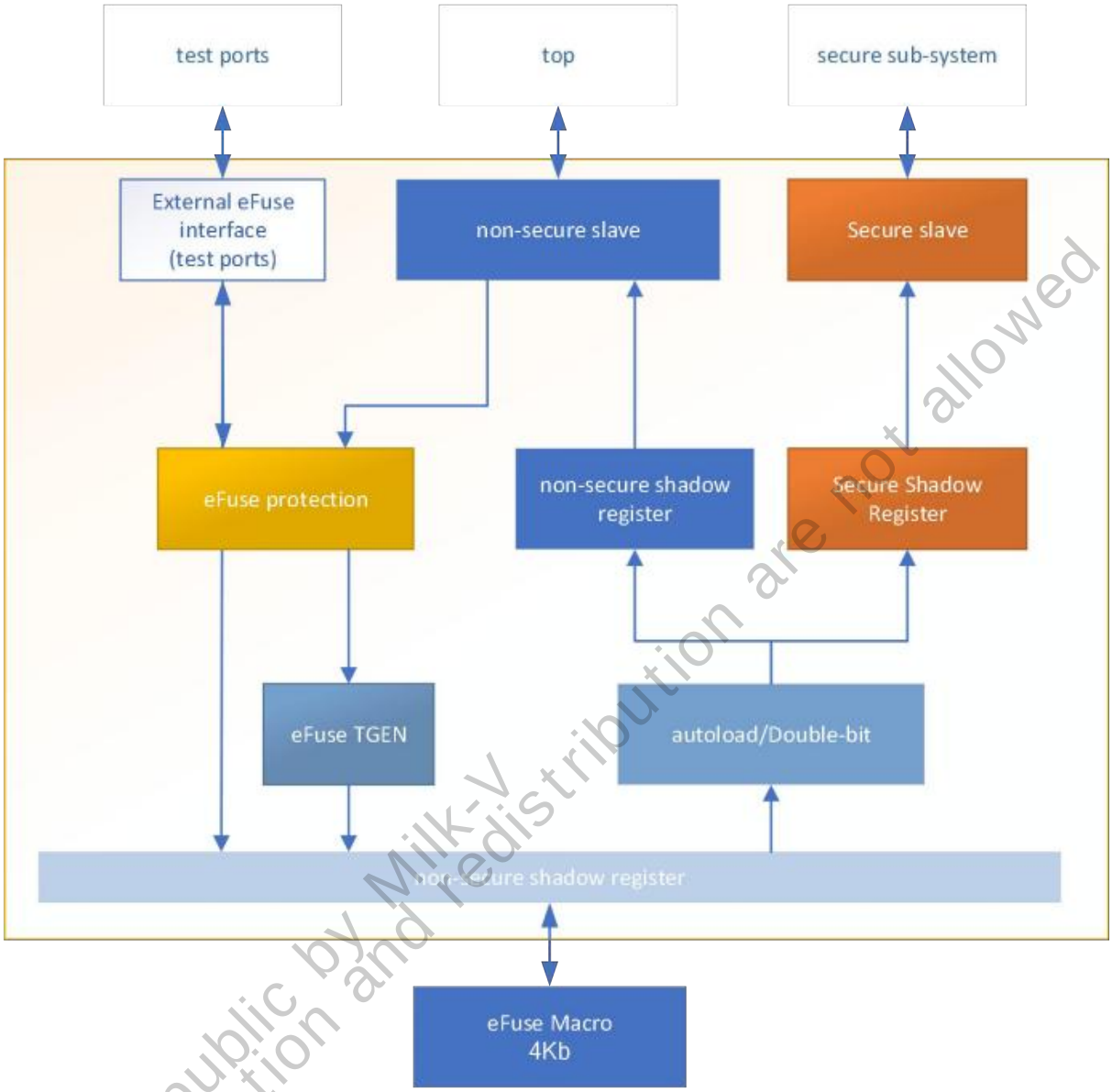


Figure 13- 5 eFuse CTRL Module architecture

13.3.2 Efuse entity address translation and virtual register address

The eFuse entity module consists of 128 lines, each line is 32bit. When reading the entity eFuse data, one line (32bit) can be read each time. When burning, one

bit can be burned each time. When reading or burning, the operation area should be specified through the 12bit entity address interface.
The arrangement is as follows:

Fuse physical address [11:0]: {bit address [11:7], row address [6:0]}

The double bit protection bar combines two adjacent different lines of entity data into one line of logical data, which is automatically loaded into the register after startup. The 4KB entity space is merged into a 64 line 32bit logical space. The system uses the logical space to provide corresponding functions except for the entity address used in eFuse programming data process.

The corresponding entity address is
Fuse physical address [11:0]: {logical address [11:7], logical row address [5:0], double address}

The double bit address is 0 or 1.

Figure 13- 2 eFuse entity (row) address and logical (row) address corresponding value

Logical row address	Macro unit entity row address		
0	0	FTSN0	Production serial number
	1	FTSN0	Production serial number
1	2	FTSN1	Production serial number
	3	FTSN1	Production serial number
2	4	FTSN2	Production serial number
	5	FTSN2	Production serial number
3	6	FTSN3	Production serial number
	7	FTSN3	Production serial number
4	8	FTSN4	Production serial number
	9	FTSN4	Production serial number

Logical row address	Macro unit entity row address		
5	10	Market Segment	Market Segment number
	11	Market Segment	Market Segment number
6	12	Analog0	Analog module calibration
	13	Analog0	Analog module calibration
7	14	Analog1	Analog module calibration
	15	Analog1	Analog module calibration
8	16	Analog2	Analog module calibration
	17	Analog2	Analog module calibration
9	18	Analog3	Analog module calibration
	19	Analog3	Analog module calibration
10	20	Bonding0	Configuration bonding settings
	21	Bonding0	Configuration bonding settings
11	22	SW_reserved	SoftwareReserved
	23	SW_reserved	SoftwareReserved
12	24	SW_reserved	SoftwareReserved
	25	SW_reserved	SoftwareReserved
13	26	SW_reserved	SoftwareReserved
	27	SW_reserved	SoftwareReserved
14	28	SW_reserved	SoftwareReserved
	29	SW_reserved	SoftwareReserved
15	30	SW_reserved	SoftwareReserved
	31	SW_reserved	SoftwareReserved
16	32	SW_reserved	SoftwareReserved
	33	SW_reserved	SoftwareReserved
17	34	SW_reserved	SoftwareReserved
	35	SW_reserved	SoftwareReserved
18	36	SW_reserved	SoftwareReserved
	37	SW_reserved	SoftwareReserved
19	38	SW_reserved	SoftwareReserved
	39	SW_reserved	SoftwareReserved
20	40	SW_reserved	SoftwareReserved
	41	SW_reserved	SoftwareReserved
21	42	SW_reserved	SoftwareReserved
	43	SW_reserved	SoftwareReserved

Logical row address	Macro unit entity row address		
22	44	SW_reserved	SoftwareReserved
	45	SW_reserved	SoftwareReserved
23	46	SW_reserved	SoftwareReserved
	47	SW_reserved	SoftwareReserved
24	48	SW_reserved	SoftwareReserved
	49	SW_reserved	SoftwareReserved
25	50	SW_reserved	SoftwareReserved
	51	SW_reserved	SoftwareReserved
26	52	SW_reserved	SoftwareReserved
	53	SW_reserved	SoftwareReserved
27	54	SW_reserved	SoftwareReserved
	55	SW_reserved	SoftwareReserved
28	56	SW_reserved	SoftwareReserved
	57	SW_reserved	SoftwareReserved
29	58	SW_reserved	SoftwareReserved
	59	SW_reserved	SoftwareReserved
30	60	SW_reserved	SoftwareReserved
	61	SW_reserved	SoftwareReserved
31	62	SW_reserved	SoftwareReserved
	63	SW_reserved	SoftwareReserved
32	64	SW_reserved	SoftwareReserved
	65	SW_reserved	SoftwareReserved
33	66	SW_reserved	SoftwareReserved
	67	SW_reserved	SoftwareReserved
34	68	SW_reserved	SoftwareReserved
	69	SW_reserved	SoftwareReserved
35	70	DID0	Chip serial number
	71	DID0	Chip serial number
36	72	DID1	Chip serial number
	73	DID1	Chip serial number
37	74	Nvcounter	Anti rollback number
	75	Nvcounter	Anti rollback number
38	76	eFuse_w_lock0	efuse Anti write options
	77	eFuse_w_lock0	efuse Anti write options

Logical row address	Macro unit entity row address		
39	78	eFuse_w_lock1	efuse Anti write options
	79	eFuse_w_lock1	efuse Anti write options
40	80	SCS_config	Security startup settings
	81	SCS_config	Security startup settings
41	82	DBG_mode	Debug interface protection settings
	83	DBG_mode	Debug interface protection settings
42	84	Kpub_Hash0	Public key hash value
	85	Kpub_Hash0	Public key hash value
43	86	Kpub_Hash1	Public key hash value
	87	Kpub_Hash1	Public key hash value
44	88	Kpub_Hash2	Public key hash value
	89	Kpub_Hash2	Public key hash value
45	90	Kpub_Hash3	Public key hash value
	91	Kpub_Hash3	Public key hash value
46	92	Kpub_Hash4	Public key hash value
	93	Kpub_Hash4	Public key hash value
47	94	Kpub_Hash5	Public key hash value
	95	Kpub_Hash5	Public key hash value
48	96	Kpub_Hash6	Public key hash value
	97	Kpub_Hash6	Public key hash value
49	98	Kpub_Hash7	Public key hash value
	99	Kpub_Hash7	Public key hash value
50	100	SecReserved	Security system reserved column
	101	SecReserved	Security system reserved column
51	102	SecReserved	Security system reserved column
	103	SecReserved	Security system reserved column
52	104	SecReserved	Security system reserved column
	105	SecReserved	Security system reserved column
53	106	SecReserved	Security system reserved column
	107	SecReserved	Security system reserved column
54	108	SecReserved	Security system reserved column
	109	SecReserved	Security system reserved column
55	110	SecReserved	Security system reserved column
	111	SecReserved	Security system reserved column
56	112	SecReserved	Security system reserved column

Logical row address	Macro unit entity row address		
	113	SecReserved	Security system reserved column
57	114	SecReserved	Security system reserved column
	115	SecReserved	Security system reserved column
58	116	SecReserved	Security system reserved column
	117	SecReserved	Security system reserved column
59	118	SecReserved	Security system reserved column
	119	SecReserved	Security system reserved column
60	120	SecReserved	Security system reserved column
	121	SecReserved	Security system reserved column
61	122	SecReserved	Security system reserved column
	123	SecReserved	Security system reserved column
62	124	SecReserved	Security system reserved column
	125	SecReserved	Security system reserved column
63	126	SecReserved	Security system reserved column
	127	SecReserved	Security system reserved column

13.3.3 Efuse Ctrl Register Overview

The eFuse Ctrl register provides two access interfaces, the insecure interface and the secure interface

Non secure interface provides

·Control interface of eFuse controller, read / write control field

eFuse register content, which can be used to read non-secure registers

Secure interface provides

Complete eFuse register content, only provide read function

The overview of non-secure registers of fuse Ctrl is shown in the table below:

Name	Address Offset	Description
EFUSE_MODE	0x000	EFUSE_MODE
EFUSE_ADR	0x004	EFUSE_ADR
EFUSE_DIR_CMD	0x008	EFUSE_DIR_CMD
EFUSE_RD_DATA	0x00c	EFUSE_RD_DATA

Name	Address Offset	Description
EFUSE_STATUS	0x010	EFUSE_STATUS
EFUSE_ONE_WAY	0x014	EFUSE_ONE_WAY
PGM_PLUSE_WIDTH	0x018	PGM_PLUSE_WIDTH
A_READ_WIDTH	0x01c	A_READ_WIDTH
M_READ_WIDTH	0x020	M_READ_WIDTH
FTSN0	0x100	Efuse Register contents
FTSN1	0x104	Efuse Register contents
FTSN2	0x108	Efuse Register contents
FTSN3	0x10c	Efuse Register contents
FTSN4	0x110	Efuse Register contents
eFuse_FT_Debug	0x114	Efuse Register contents
Analog0	0x118	Efuse Register contents
Analog1	0x11c	Efuse Register contents
Analog2	0x120	Efuse Register contents
Analog3	0x124	Efuse Register contents
Bonding0	0x128	Efuse Register contents
SW_info	0x12c	Efuse Register contents
SW_reserved30	0x130	Efuse Register contents
SW_reserved34	0x134	Efuse Register contents
SW_reserved38	0x138	Efuse Register contents
SW_reserved3c	0x13c	Efuse Register contents
SW_reserved40	0x140	Efuse Register contents
SW_reserved44	0x144	Efuse Register contents
SW_reserved48	0x148	Efuse Register contents
SW_reserved4c	0x14c	Efuse Register contents
SW_reserved50	0x150	Efuse Register contents
SW_reserved54	0x154	Efuse Register contents
SW_reserved58	0x158	Efuse Register contents
SW_reserved5c	0x15c	Efuse Register contents
SW_reserved60	0x160	Efuse Register contents
SW_reserved64	0x164	Efuse Register contents
SW_reserved68	0x168	Efuse Register contents
SW_reserved6c	0x16c	Efuse Register contents
SW_reserved70	0x170	Efuse Register contents
SW_reserved74	0x174	Efuse Register contents
SW_reserved78	0x178	Efuse Register contents
SW_reserved7c	0x17c	Efuse Register contents
SW_reserved80	0x180	Efuse Register contents
SW_reserved84	0x184	Efuse Register contents
SW_reserved88	0x188	Efuse Register contents
DID0	0x18c	Efuse Register contents
DID1	0x190	Efuse Register contents
MSID	0x194	Efuse Register contents
eFuse_w_lock0	0x198	Efuse Register contents
eFuse_w_lock1	0x19c	Efuse Register contents
SCS_config	0x1a0	Efuse Register contents
DBG_mode	0x1a4	Efuse Register contents
Kpub_Hash0	0x1a8	Efuse Register contents
Kpub_Hash1	0x1ac	Efuse Register contents
Kpub_Hash2	0x1b0	Efuse Register contents
Kpub_Hash3	0x1b4	Efuse Register contents
Kpub_Hash4	0x1b8	Efuse Register contents
Kpub_Hash5	0x1bc	Efuse Register contents
Kpub_Hash6	0x1c0	Efuse Register contents

Name	Address Offset	Description
Kpub_Hash7	0x1c4	Efuse Register contents

13.3.4 Efuse CTRL Register Overview

(Base Address 0x03050000)

EFUSE_MODE

Efuse operation mode

Offset Address: 0x000

Bits	Name	Access	Description	Reset
3:0	EFUSE_OP_MODE	R/W	eFuse firewall control register built in instructions: 0000: standby status / module start status 0001: reads eFuse instruction normally 0010: test pressure reading eFuse command 0100: burn instruction 1xxx: eFuse module shutdown	0x0
4	eFuse_A_Fire	W1T	Start eFuse built-in instruction action	
7:5	eFuse_REFRESH_MODE	R/W	001: Start shadow register update Others:Reserved	0x0
10:8	eFuse_DIRECT_MODE	R/W	Reserved	0x0
11	clear_eFuse_status	R/W	Clear error information 1: Clear 0: no action	0x0
31:12	EFUSE_MODE_reseved	R/W	Reserved	0x0

EFUSE_ADR

Efuse address for embedded opreation

Offset Address: 0x004

Bits	Name	Access	Description	Reset
11:0	EFUSE_ADR	R/W	Specifies the eFuse physical address used by the built-in instruction	0x0
31:12	Reserved			

EFUSE_DIR_CMD

direct bit-wise reg-control signals to eFuse macro IO

Offset Address: 0x008

Bits	Name	Access	Description	Reset
31:0	EFUSE_DIR_CMD	R/W	Reserved, Test interface	0x0

EFUSE_RD_DATA

eFuse Macro readback data for embedded read

Offset Address: 0x00c

Bits	Name	Access	Description	Reset
31:0	EFUSE_RD_DATA	RO	eFuse Module read value	

EFUSE_STATUS

eFuse_busy

Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	eFuse_busy	RO	eFuseController busy	
1	EFUSE_READ_err	RO	Read action error indication	
2	EFUSE_M_READ_err	RO	Pressure reading action error indication	
3	EFUSE_PGM_err	RO	Burning action error indication	
7:4	EfuseCTL_ST	RO	Controller status indication 0: Started 1: Auto read 2: Waiting 3: Reading 4: Burning and writing 5: Pressure reading 6: Test model test 7: Shutting down	
31:8	EFUSE_STATUS	RO	Reserved	

EFUSE_ONE_WAY

EFUSE_ONE_WAY

Offset Address: 0x014

Bits	Name	Access	Description	Reset
31:0	EFUSE_ONE_WAY	R/W	Reserved **One-Way:** Enabled	0x0

PGM_PLUSE_WIDTH

PGM_PLUSE_WIDTH

Offset Address: 0x018

Bits	Name	Access	Description	Reset
8:0	PGM_PLUSE_WIDTH	R/W	Reserved	0x0
31:9	Reserved			

A_READ_WIDTH

A_READ_WIDTH

Offset Address: 0x01c

Bits	Name	Access	Description	Reset
8:0	A_READ_WIDTH	R/W	Reserved	0x0
31:9	Reserved			

M_READ_WIDTH

M_READ_WIDTH

Offset Address: 0x020

Bits	Name	Access	Description	Reset
8:0	M_READ_WIDTH	R/W	Reserved	0x0
31:9	Reserved			

FTSN0

serial number for FT(function test)

Offset Address: 0x100

Bits	Name	Access	Description	Reset
31:0	FTSN0	RO	Production serial number	

FTSN1

serial number for FT(function test)

Offset Address: 0x104

Bits	Name	Access	Description	Reset
31:0	FTSN1	RO	Production serial number	

FTSN2

serial number for FT(function test)

Offset Address: 0x108

Bits	Name	Access	Description	Reset
31:0	FTSN2	RO	Production serial number	

FTSN3

serial number for FT(function test)

Offset Address: 0x10c

Bits	Name	Access	Description	Reset
31:0	FTSN3	RO	Production serial number	

FTSN4

serial number for FT(function test)

Offset Address: 0x110

Bits	Name	Access	Description	Reset
31:0	FTSN4	RO	Production serial number	

MRK_SEG

Market Segment

Offset Address: 0x114

Bits	Name	Access	Description	Reset
31:0	MRK_SEG	RO	Market differentiation number	

Analog0

Analog trimming data

Offset Address: 0x118

Bits	Name	Access	Description	Reset
31:0	Analog0	RO	Analog module calibration	

Analog1

Analog trimming data

Offset Address: 0x11c

Bits	Name	Access	Description	Reset
31:0	Analog1	RO	Analog module calibration	

Analog2

Analog trimming data

Offset Address: 0x120

Bits	Name	Access	Description	Reset
31:0	Analog2	RO	Analog module calibration	

Analog3

Analog trimming data

Offset Address: 0x124

Bits	Name	Access	Description	Reset
31:0	Analog3	RO	Analog module calibration	

Bonding0

Bonding option

Offset Address: 0x128

Bits	Name	Access	Description	Reset
31:0	Bonding0	RO	Configuration binding settings	

SW_info

reserved for SW or MBIST use

Offset Address: 0x12c

Bits	Name	Access	Description	Reset
31:0	SW_info	RO	SoftwareReserved	

SW_reserved30

reserved for SW or MBIST use

Offset Address: 0x130

Bits	Name	Access	Description	Reset
31:0	reserved30	RO	SoftwareReserved	

SW_reserved34

reserved for SW or MBIST use

Offset Address: 0x134

Bits	Name	Access	Description	Reset

Bits	Name	Access	Description	Reset
31:0	reserved34	RO	SoftwareReserved	

SW_reserved38

reserved for SW or MBIST use

Offset Address: 0x138

Bits	Name	Access	Description	Reset
31:0	reserved38	RO	SoftwareReserved	

SW_reserved3c

reserved for SW or MBIST use

Offset Address: 0x13c

Bits	Name	Access	Description	Reset
31:0	reserved3c	RO	SoftwareReserved	

SW_reserved40

reserved for SW or MBIST use

Offset Address: 0x140

Bits	Name	Access	Description	Reset
31:0	reserved40	RO	SoftwareReserved	

SW_reserved44

reserved for SW or MBIST use

Offset Address: 0x144

Bits	Name	Access	Description	Reset
31:0	reserved44	RO	SoftwareReserved	

SW_reserved48

reserved for SW or MBIST use

Offset Address: 0x148

Bits	Name	Access	Description	Reset
31:0	reserved48	RO	SoftwareReserved	

SW_reserved4c

reserved for SW or MBIST use

Offset Address: 0x14c

Bits	Name	Access	Description	Reset
31:0	reserved4c	RO	SoftwareReserved	

SW_reserved50

reserved for SW or MBIST use

Offset Address: 0x150

Bits	Name	Access	Description	Reset
31:0	reserved50	RO	SoftwareReserved	

SW_reserved54

reserved for SW or MBIST use

Offset Address: 0x154

Bits	Name	Access	Description	Reset
31:0	reserved54	RO	SoftwareReserved	

SW_reserved58

reserved for SW or MBIST use

Offset Address: 0x158

Bits	Name	Access	Description	Reset
31:0	reserved58	RO	SoftwareReserved	

SW_reserved5c

reserved for SW or MBIST use

Offset Address: 0x15c

Bits	Name	Access	Description	Reset
31:0	reserved5c	RO	SoftwareReserved	

SW_reserved60

reserved for SW or MBIST use

Offset Address: 0x160

Bits	Name	Access	Description	Reset
31:0	reserved60	RO	SoftwareReserved	

SW_reserved64

reserved for SW or MBIST use

Offset Address: 0x164

Bits	Name	Access	Description	Reset
31:0	reserved64	RO	SoftwareReserved	

SW_reserved68

reserved for SW or MBIST use

Offset Address: 0x168

Bits	Name	Access	Description	Reset
31:0	reserved68	RO	SoftwareReserved	

SW_reserved6c

reserved for SW or MBIST use

Offset Address: 0x16c

Bits	Name	Access	Description	Reset
31:0	reserved6c	RO	SoftwareReserved	

SW_reserved70

reserved for SW or MBIST use

Offset Address: 0x170

Bits	Name	Access	Description	Reset
31:0	reserved70	RO	SoftwareReserved	

SW_reserved74

reserved for SW or MBIST use

Offset Address: 0x174

Bits	Name	Access	Description	Reset
31:0	reserved74	RO	SoftwareReserved	

SW_reserved78

reserved for SW or MBIST use

Offset Address: 0x178

Bits	Name	Access	Description	Reset
31:0	reserved78	RO	SoftwareReserved	

SW_reserved7c

reserved for SW or MBIST use

Offset Address: 0x17c

Bits	Name	Access	Description	Reset
31:0	reserved7c	RO	SoftwareReserved	

SW_reserved80

reserved for SW or MBIST use

Offset Address: 0x180

Bits	Name	Access	Description	Reset
31:0	reserved80	RO	SoftwareReserved	

SW_reserved84

reserved for SW or MBIST use

Offset Address: 0x184

Bits	Name	Access	Description	Reset
31:0	reserved84	RO	SoftwareReserved	

SW_reserved88

reserved for SW or MBIST use

Offset Address: 0x188

Bits	Name	Access	Description	Reset
31:0	reserved88	RO	SoftwareReserved	

DID0

device ID

Offset Address: 0x18c

Bits	Name	Access	Description	Reset
31:0	DID0	RO	Chip serial number	

DID1

device ID

Offset Address: 0x190

Bits	Name	Access	Description	Reset
31:0	DID1	RO	Chip serial number	

MSID

Nvcounter

Offset Address: 0x194

Bits	Name	Access	Description	Reset
31:0	MSID	RO	Anti rollback number	

eFuse_w_lock0

eFuse_w_lock0

Offset Address: 0x198

Bits	Name	Access	Description	Reset
31:0	eFuse_w_lock0	RO	efuse Anti write options	

eFuse_w_lock1

eFuse_w_lock1

Offset Address: 0x19c

Bits	Name	Access	Description	Reset
31:0	eFuse_w_lock1	RO	efuse Anti write options	

SCS_config

SCS_config

Offset Address: 0x1a0

Bits	Name	Access	Description	Reset
31:0	SCS_enable	RO	Security startup settings	

DBG_mode

DBG_mode

Offset Address: 0x1a4

Bits	Name	Access	Description	Reset
31:0	REE_dbg_mode	RO	Debug interface protection settings	

Kpub_Hash0

Hash value of boot loader authentication public key

Offset Address: 0x1a8

Bits	Name	Access	Description	Reset
31:0	Kpub_Hash0	RO	Public key hash value	

Kpub_Hash1

Hash value of boot loader authentication public key

Offset Address: 0x1ac

Bits	Name	Access	Description	Reset
31:0	Kpub_Hash1	RO	Public key hash value	

Kpub_Hash2

Hash value of boot loader authentication public key

Offset Address: 0x1b0

Bits	Name	Access	Description	Reset
31:0	Kpub_Hash2	RO	Public key hash value	

Kpub_Hash3

Hash value of boot loader authentication public key

Offset Address: 0x1b4

Bits	Name	Access	Description	Reset
31:0	Kpub_Hash3	RO	Public key hash value	

Kpub_Hash4

Hash value of boot loader authentication public key

Offset Address: 0x1b8

Bits	Name	Access	Description	Reset
31:0	Kpub_Hash4	RO	Public key hash value	

Kpub_Hash5

Hash value of boot loader authentication public key

Offset Address: 0x1bc

Bits	Name	Access	Description	Reset
31:0	Kpub_Hash5	RO	Public key hash value	

Kpub_Hash6

Hash value of boot loader authentication public key

Offset Address: 0x1c0

Bits	Name	Access	Description	Reset
31:0	Kpub_Hash6	RO	Public key hash value	

Kpub_Hash7

Hash value of boot loader authentication public key

Offset Address: 0x1c4

Bits	Name	Access	Description	Reset
31:0	Kpub_Hash7	RO	Public key hash value	

13.3.5 eFuse CTRL Operation Process

The contents of eFuse will automatically load the eFuse contents into the internal registers during the boot process, and set the eFuse to a closed state after loading is completed.

13.3.5.1 eFuse Start Process

EFuse Ctrl will enter the closed state after the startup procedure is completed. Before further action, the controller must be started.
Step 1: Read eFuse_ Status [0], until the read back value is 0, indicating that the eFuse controller is idle and can initiate the next operation.
Step 2: Set eFuse Ctrl to write 0x40 to start the commandeFuse CTRL

13.3.5.2 eFuse Burning Process

eFuse CTRL has built-in burn command to do the burning of single bit eFuse.
Step 1: Read EFUSE_STATUS [0], until the read back value is 0, indicating that the eFuse controller is idle and can initiate the next operation.
Step 2: Convert the value to be burned into the entity address of eFuse and fill in EFUSE_ADR
Step 3: Set eFuse CTRL to write 0x14 and start the burning command
Step 4: Read EFUSE_STATUS [0], until the read back value is 0, indicating that the eFuse controller has completed the burning and writing, and can initiate the next operation.
Step 5: Set eFuse Ctrl to write 0x12 and start the read command.
Step 6: Check EFUSE_RD_DATA whether the burned address in data can be read back to the previously burned value.

13.3.5.3 eFuse Closing Process

After the eFuse is started manually, it can still control the eFuse to go back to the off state, which can save power consumption and avoid the eFuse's invalid action.

Step 1: Read EFUSE_STATUS [0], until the read back value is 0, indicating that the eFuse controller is idle and can initiate the next operation.
Step 2: Set eFuse Ctrl to write to 0xf0 to close the command

14 Intelligent Secure Operation Environment

In order to keep the operation environment of the blockhouse security system reliable and the security of intelligent programs and data not threatened, the operation system must establish a perfect intelligent security operation environment to provide comprehensive and complete protection for valuable assets. In particular, the intelligent program library and personal identification data must provide confidentiality, credibility and integrity to ensure the asset security of manufacturers and users.

According to the requirements of intelligent secure operation environment, the security system provides complete hardware and software protection functions from startup,

It includes: 1. The establishment of trust chain: providing the foundation of security environment, which is the foundation of trusted environment, such as hardware security setting, trust root, security startup; 2. Data encryption security: data encryption program, operation core encryption; 3. Execution isolation: Hardware isolation, software isolation; 4. Software and firmware verification: verifying the credibility and integrity of software, including Boot and load verification procedures, 5. Secure storage and transmission: protect external data storage and exchange, 6. Secure update: ensure a secure environment. These functions provide a systematic method and system to protect valuable program code and data.

14.1 Establishment of Trust Chain

When the device is started, the hardware system will start the security mechanism according to the security settings. Secure boot refers to the trusted platform startup sequence for security applications. After powering on, the initial instruction is read from the built-in read-only memory (ROM), which is the secure boot ROM of the system. This program contains the secure boot root public key for authorization authentication, which will verify the root signature of the underlying boot loader. Once it passes the verification, the system will start. Then it verifies subsequent boot loaders. After

verifying the legality of the signature, the entire boot program starts to load the driver, detect devices, and start the system daemon. If any component fails the check, that component will not be loaded, and the secure boot process will fail.

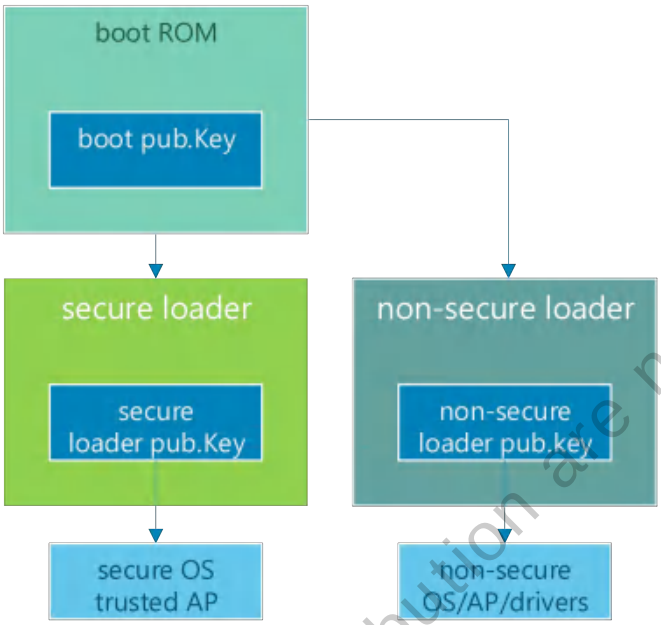


Figure 14- 1 Establishment of trust chain

14.2 Data Encryption Security

For confidential programs or data that need to be kept secret, including loaders, secure and non secure system programs required for startup, secure and non secure code and data required for application, such as AI model, security system can provide encryption protection programs, and use secure key operation to complete in the trusted execution environment. After decryption, the data of highly confidential programs or data are also isolated from the system .In a secure environment, non secure program access is not allowed. The system provides common public standard cryptographic algorithms, including symmetric and asymmetric encryption and decryption and signature verification algorithms, such as AES, des or RSA, ECC, etc. It also supports Chinese cryptographic algorithms, including SM2, SM3 and SM4, which meet a wide range of security standards. The key is managed by the security system, and the non security program is not allowed to use the security key.

14.3 Software and Hardware verification

The trust chain will ensure that all system components are officially written, signed and distributed, and can not come from other unknown organizations, such as malicious attackers from third parties. The trust chain is also used to check the signature when the application starts. All applications must be signed directly or indirectly by the official to ensure the credibility and integrity of the system program and prevent the program from being modified or implanted with malicious programs or backdoors. In the running stage, if the system wants to load the program library dynamically, such as the artificial intelligence model library, it still needs to be verified and decrypted by the security system to ensure the credibility of the program library.

14.4 Secure Storage and Transmission

Due to the application requirements, the system must transfer the security data into or out of the security environment to achieve data storage or exchange. For the storage or transmission of security data, the security operation environment only allows data transmission through the pre-defined security interface. The security interface includes secure storage, secure debugging and secure connection.

Secure storage requires that secure data cannot be transmitted to external storage media in plaintext. All secure data must be encrypted according to the device security key or private password corresponding to its security level before moving out of the security environment. The external storage data exists in ciphertext. Security debugging needs to use different security passwords to connect the debugging interface according to the security level of the running environment. Secure connection ensures the security of network equipment in the transmission process.

14.5 Security Update

In the face of ever-changing attacks, the security environment must maintain continuous security updates and patches. In addition to application security updates, the chip security environment also provides firmware updates that support secure

startup, as well as version control and anti rollback protection. The security update firmware needs to be verified by the authorized signature to start the security firmware update, and the version of the chip is consistent with the firmware through eFuse.

Through the comprehensive security protection mechanism, an intelligent and secure operation environment is established, which can effectively guarantee the data security of manufacturers and users, and meet the national security requirements.