



# H133

## User Manual

Revision 1.4  
September 20, 2022

## Revision History

Revision	Date	Author	Description
1.0	Jul 22, 2021	AWAXXXX	Initial release version
1.1	August 27, 2021	XRA0041	Refresh some specifications
1.2	March 23, 2022	KPA0570	Update section 2.2.7.4, section 3.2.3.3, section 3.2.3.4, section 3.2.5, section 3.2.6, section 7.3, section 8.7.5.17, table 8-20, table 8-24, table 9-21, section 9.7.5, section 2.2.3
1.3	July 22, 2022	KPA0570	Update Figure 3-5 and Table 3-3
1.4	September 20, 2022	KPA0570	<p><b>Chapter 2 Product Description</b> Updated section 2.2.6.9 I/O Memory Management Unit.</p> <p><b>Chapter 3 System</b></p> <ol style="list-style-type: none"><li>1. Updated 3.1 Memory Mapping and Table 3-12 Interrupt Sources.</li><li>2. Updated Figure 3-3 System Bus Tree, Figure 3-4 Module Clock Generation, Figure 3-5 PLL Distribution, and Table 3-3 PLL Typical Application.</li></ol> <p><b>Chapter 5 Video Output Interfaces</b></p> <ol style="list-style-type: none"><li>1. Updated section 5.1.1 Overview, deleted the section RGB Gamma Correction and the section HV Mode Configuration Process.</li><li>2. Added 5.1.6.14 0x0058 LCD HV Panel Interface Register.</li></ol>



# Contents

Revision History .....	i
Contents.....	iii
1 About This Document .....	1
1.1 Purpose and Scope.....	1
1.2 Intended Audience.....	1
1.3 Symbol Conventions.....	1
1.3.1 Symbol Conventions.....	1
1.3.2 Table Content Conventions.....	1
1.3.3 Reset Value Conventions .....	1
1.3.4 Register Attributes .....	2
1.3.5 Numerical System .....	2
Contents.....	3
2 Product Description .....	4
2.1 Overview .....	4
2.2 Features .....	4
2.2.1 CPU Architecture.....	4
2.2.2 Memory SubSystem .....	4
2.2.3 Video Engine .....	6
2.2.4 Video and Graphics .....	6
2.2.5 Video Output.....	7
2.2.6 System Peripherals.....	8
2.2.7 Audio Interfaces.....	12
2.2.8 Security System.....	14
2.2.9 External Peripherals .....	15
2.2.10 LEDC .....	20
2.2.11 Package .....	21
2.3 Block Diagram .....	21
Contents.....	23
Figures.....	26
Tables .....	28

---

3	System.....	29
3.1	Memory Mapping .....	29
3.2	CPU Configuration.....	32
3.2.1	Overview .....	32
3.2.2	Functional Descriptions.....	32
3.2.3	Cluster 0 Configuration Register List.....	34
3.2.4	Cluster 0 Configuration Register Description.....	35
3.2.5	CPU Subsystem Control Register List .....	43
3.2.6	CPU Subsystem Control Register Description .....	44
3.3	Clock Controller Unit (CCU).....	48
3.3.1	Overview .....	48
3.3.2	Block Diagram .....	48
3.3.3	Functional Description .....	49
3.3.4	Programming Guidelines.....	55
3.3.5	Register List .....	61
3.3.6	Register Description .....	64
3.4	BROM System.....	146
3.4.1	Overview .....	146
3.4.2	Functional Description .....	146
3.5	System Configuration.....	157
3.5.1	Overview .....	157
3.5.2	Register List .....	157
3.5.3	Register Description .....	157
3.6	Timer .....	165
3.6.1	Overview .....	165
3.6.2	Block Diagram .....	166
3.6.3	Functional Description .....	166
3.6.4	Programming Guidelines.....	170
3.6.5	Register List .....	171
3.6.6	Register Description .....	172
3.7	High Speed Timer .....	184
3.7.1	Overview .....	184
3.7.2	Block Diagram .....	184

---

3.7.3	Functional Description .....	184
3.7.4	Programming Guidelines.....	187
3.7.5	Register List.....	188
3.7.6	Register Description .....	189
3.8	Generic Interrupt Controller (GIC) .....	196
3.8.1	Overview .....	196
3.8.2	Functional Description .....	196
3.9	Direct Memory Access Controller (DMAC).....	204
3.9.1	Overview .....	204
3.9.2	Block Diagram .....	204
3.9.3	Functional Description .....	205
3.9.4	Programming Guidelines.....	214
3.9.5	Register List.....	217
3.9.6	Register Description .....	218
3.10	Thermal Sensor Controller (THS) .....	239
3.10.1	Overview .....	239
3.10.2	Block Diagram .....	239
3.10.3	Functional Description .....	239
3.10.4	Programming Guidelines.....	241
3.10.5	Register List.....	243
3.10.6	Register Description .....	243
3.11	IOMMU .....	249
3.11.1	Overview .....	249
3.11.2	Block Diagram .....	249
3.11.3	Functional Description .....	251
3.11.4	Programming Guidelines.....	261
3.11.5	Register List.....	262
3.11.6	Register Description .....	266
3.12	RTC .....	338
3.12.1	`Overview .....	338
3.12.2	Block Diagram .....	338
3.12.3	Functional Description .....	339
3.12.4	Programming Guidelines.....	343

---

---

3.12.5 Register List.....	345
3.12.6 Register Description.....	346
Contents.....	358
Figures.....	359
4    Video and Graphics .....	360
4.1  DE .....	360
4.2  DI .....	361
4.3  G2D.....	362
4.4  Video Decoding .....	363
4.4.1  Overview .....	363
4.4.2  Block Diagram .....	364
4.5  Video Encoding .....	365
4.5.1  Overview .....	365
4.5.2  Block Diagram .....	365
Contents.....	367
Figures.....	368
Tables .....	369
5    Video Output Interfaces.....	370
5.1  TCON LCD .....	370
5.1.1  Overview .....	370
5.1.2  Block Diagram .....	370
5.1.3  Functional Description .....	370
5.1.4  Programming Guidelines.....	376
5.1.5  Register List.....	380
5.1.6  Register Description .....	381
5.2  TCON TV .....	413
5.2.1  Overview .....	413
5.2.2  Functional Description .....	413
5.2.3  Register List.....	413
5.2.4  Register Description .....	414
5.3  MIPI DSI.....	426
5.3.1  Overview .....	426
5.4  HDMI .....	427

---

5.4.1 Overview .....	427
Contents.....	428
Figures.....	429
Tables .....	430
6 Memory.....	431
6.1 SDRAM Controller (DRAMC) .....	431
6.2 SD/MMC Host Controller (SMHC) .....	432
6.2.1 Overview .....	432
6.2.2 Block Diagram .....	432
6.2.3 Functional Description .....	433
6.2.4 Programming Guidelines.....	445
6.2.5 Register List.....	451
6.2.6 Register Description .....	452
Contents.....	484
Figures.....	485
Tables .....	487
7 Audio.....	488
7.1 I2S/PCM.....	488
7.1.1 Overview .....	488
7.1.2 Block Diagram .....	488
7.1.3 Functional Description .....	490
7.1.4 Programming Guidelines.....	496
7.1.5 Register List.....	498
7.1.6 Register Description .....	500
7.2 DMIC.....	548
7.2.1 Overview .....	548
7.2.2 Block Diagram .....	548
7.2.3 Functional Description .....	548
7.2.4 Register List.....	551
7.2.5 Register Description .....	551
7.3 OWA .....	566
7.3.1 Overview .....	566
7.3.2 Block Diagram .....	566

---

7.3.3	Functional Description .....	567
7.3.4	Register List .....	571
7.3.5	Register Description .....	572
7.4	Audio Codec .....	582
7.4.1	Overview .....	582
7.4.2	Block Diagram .....	582
7.4.3	Functional Description .....	583
7.4.4	Programming Guidelines.....	590
7.4.5	Register List .....	591
7.4.6	Register Description .....	593
Contents.....		630
Figures.....		633
Tables .....		637
8	Interfaces .....	639
8.1	TWI .....	639
8.1.1	Overview .....	639
8.1.2	Block Diagram .....	639
8.1.3	Functional Description .....	641
8.1.4	Programming Guidelines.....	646
8.1.5	Register List .....	649
8.1.6	Register Description .....	650
8.2	UART.....	665
8.2.1	Overview .....	665
8.2.2	Block Diagram .....	665
8.2.3	Functional Description .....	666
8.2.4	Programming Guidelines.....	672
8.2.5	Register List .....	675
8.2.6	Register Description .....	677
8.3	SPI.....	706
8.3.1	Overview .....	706
8.3.2	Block Diagram .....	706
8.3.3	Functional Description .....	708
8.3.4	Programming Guidelines.....	716

---

8.3.5 Register List.....	719
8.3.6 Register Description.....	720
8.4 SPI_DBI.....	741
8.4.1 Overview .....	741
8.4.2 Block Diagram .....	742
8.4.3 Functional Description .....	743
8.4.4 Programming Guidelines.....	763
8.4.5 Register List.....	771
8.4.6 Register Description.....	772
8.5 USB2.0 DRD .....	804
8.5.1 Overview .....	804
8.5.2 Block Diagram .....	805
8.5.3 Functional Description .....	805
8.6 USB2.0 HOST .....	807
8.6.1 Overview .....	807
8.6.2 Block Diagram .....	807
8.6.3 Functional Description .....	808
8.6.4 Register List.....	808
8.6.5 EHCI Register Description.....	810
8.6.6 OHCI Register Description.....	827
8.7 GPIO .....	849
8.7.1 Overview .....	849
8.7.2 Block Diagram .....	849
8.7.3 Functional Description .....	850
8.7.4 Register List.....	858
8.7.5 Register Description .....	860
8.8 GPADC.....	934
8.8.1 Overview .....	934
8.8.2 Block Diagram .....	935
8.8.3 Functional Description .....	935
8.8.4 Programming Guidelines.....	937
8.8.5 Register List.....	938
8.8.6 Register Description .....	939

---

8.9 PWM .....	946
8.9.1 Overview .....	946
8.9.2 Block Diagram .....	946
8.9.3 Functional Description .....	947
8.9.4 Programming Guidelines.....	957
8.9.5 Register List.....	958
8.9.6 Register Description.....	960
8.10 LEDC .....	987
8.10.1 Overview .....	987
8.10.2 Block Diagram .....	987
8.10.3 Functional Description .....	988
8.10.4 Programming Guidelines.....	993
8.10.5 Register List.....	998
8.10.6 Register Description.....	998
8.11 EMAC.....	1009
8.11.1 Overview .....	1009
8.11.2 Block Diagram .....	1009
8.11.3 Functional Description .....	1010
8.11.4 Programming Guidelines.....	1017
8.11.5 Register List.....	1018
8.11.6 Register Description.....	1019
8.12 CIR Receiver .....	1037
8.12.1 Overview .....	1037
8.12.2 Block Diagram .....	1037
8.12.3 Functional Description .....	1038
8.12.4 Programming Guidelines.....	1042
8.12.5 Register List.....	1043
8.12.6 Register Description.....	1043
8.13 CIR Transmitter .....	1049
8.13.1 Overview .....	1049
8.13.2 Block Diagram .....	1049
8.13.3 Functional Description .....	1050
8.13.4 Programming Guidelines.....	1052

---

---

8.13.5 Register List.....	1053
8.13.6 Register Description.....	1053
Contents.....	1061
Figures.....	1062
9 Security System.....	1063
9.1 Crypto Engine.....	1063
9.1.1 Overview .....	1063
9.1.2 Block Diagram .....	1064
9.1.3 Functional Description .....	1065
9.1.4 Programming Guidelines.....	1084
9.1.5 Register List.....	1087
9.1.6 Register Description .....	1088
9.2 Security ID .....	1092
Appendix: Glossary .....	1093

# 1 About This Document

## 1.1 Purpose and Scope

This document describes the features, logical structures, functions, operating modes, and related registers of each module about H133. For details about the interface timings and related parameters, the pins, pin usages, performance parameters, and package dimension, please refer to the [H133\\_Datasheet](#).

## 1.2 Intended Audience

The document is intended for:

- Design and maintenance personnel for electronics
- Programmers in writing code or modifying the Allwinner provided code

## 1.3 Symbol Conventions

### 1.3.1 Symbol Conventions

The symbols that may be found in this document are defined as follows.

Symbol	Description
 <b>WARNING</b>	Indicates potential risk of injury or death exists if the instructions are not obeyed.
 <b>CAUTION</b>	Indicates potential risk of equipment damage, data loss, performance degradation, or unexpected results exists if the instructions are not obeyed.
 <b>NOTE</b>	Provides additional information to emphasize or supplement important points of the main text.

### 1.3.2 Table Content Conventions

The table content conventions that may be found in this document are defined as follows.

Symbol	Description
/	The cell is blank.

### 1.3.3 Reset Value Conventions

In the register definition tables:

If other column value in a bit or multiple bits row is “/”, that this bit or these multiple bits are unused.

If the default value of a bit or multiple bits is “UDF”, that the default value is undefined.

#### 1.3.4 Register Attributes

The register attributes that may be found in this document are defined as follows.

Symbol	Description
R	Read Only
R/W	Read/Write
R/WAC	Read/Write-Automatic-Clear, clear the bit automatically when the operation of complete. Writing 0 has no effect
R/WC	Read/Write-Clear
R/W0C	Read/Write 0 to Clear. Writing 1 has no effect
R/W1C	Read/Write 1 to Clear. Writing 0 has no effect
R/W1S	Read/Write 1 to Set. Writing 0 has no effect
W	Write Only

#### 1.3.5 Numerical System

The expressions of data capacity, frequency, and data rate are described as follows.

Type	Symbol	Value
Data capacity	1 K	1024
	1 M	1,048,576
	1 G	1,073,741,824
Frequency, data rate	1 k	1000
	1 M	1,000,000
	1 G	1,000,000,000

The expressions of addresses and data are described as follows.

Symbol	Example	Description
0x	0x0200, 0x79	Address or data in hexadecimal
0b	0b010, 0b00 000 111	Data or sequence in binary (register description is excluded.)
X	00X, XX1	In data expression, X indicates 0 or 1. For example, 00X indicates 000 or 001 and XX1 indicates 001, 011, 101 or 111.

# Contents

2 Product Description .....	4
2.1 Overview .....	4
2.2 Features .....	4
2.2.1 CPU Architecture.....	4
2.2.2 Memory SubSystem .....	4
2.2.3 Video Engine .....	6
2.2.4 Video and Graphics .....	6
2.2.5 Video Output.....	7
2.2.6 System Peripherals.....	8
2.2.7 Audio Interfaces .....	12
2.2.8 Security System.....	14
2.2.9 External Peripherals .....	15
2.2.10 LEDC .....	20
2.2.11 Package .....	21
2.3 Block Diagram .....	21

## 2 Product Description

### 2.1 Overview

H133 is an advanced application processor designed for 4K decoding platform. It integrates the high-performance dual-core Cortex™-A7 to provide the most-efficient computing power. H133 supports full format decoding such as H.265, H.264, MPEG-1/2/4, JPEG, VC1, and so on. The independent encoder can encode in JPEG or MJPEG. Integrated multi ADCs/DACs and I2S/PCM/DMIC/OWA audio interfaces can work seamlessly with the CPU to accelerate multimedia algorithms and improve the user experience. H133 supports HDMI/LVDS/MIPI DSI display output interfaces to meet the requirements of the screen display. H133 comes with extensive connectivity and interfaces, such as USB, SDIO, EMAC, TWI, UART, SPI, PWM, GPADC, IR TX&RX, and so on. Besides, H133 can connect with other different peripherals like WiFi and BT via SDIO and UART.

### 2.2 Features

#### 2.2.1 CPU Architecture

- Dual-core ARM Cortex™-A7
- 32 KB L1 I-cache + 32 KB L1 D-cache per core, 256 KB L2 cache

#### 2.2.2 Memory SubSystem

##### 2.2.2.1 Boot ROM (BROM)

- On-chip memory
- Supports system boot from the following devices:
  - SD card
  - eMMC
  - SPI NOR Flash
  - SPI NAND Flash
- Supports mandatory upgrade process through USB and SD card
- Supports GPIO pin and eFuse module to select the boot media type

- Supports normal booting and secure booting
- Secure BROM loads only certified firmware
- Secure BROM ensures that the secure boot is a trusted environment

### 2.2.2.2 SDRAM

- Supports DDR2/DDR3 SDRAM
- Maximum capacity up to 2 GB
- Supports clock frequency up to 533 MHz for DDR2
- Supports clock frequency up to 800 MHz for DDR3

### 2.2.2.3 SMHC

- Three SD/MMC host controller (SMHC) interfaces
- The SMHC0 controls the devices that comply with the protocol Secure Digital Memory (SD mem-version 3.0)
- The SMHC1 controls the device that complies with the protocol Secure Digital I/O (SDIO-version 3.0)
- The SMHC2 controls the device that complies with the protocol Multimedia Card (eMMC-version 5.0)
- Maximum performance:
  - SDR mode 150 MHz@1.8 V IO pad
  - DDR mode 50 MHz@1.8 V IO pad
  - DDR mode 50 MHz@3.3 V IO pad
- Supports 1-bit or 4-bit data width
- Supports block size of 1 to 65535 bytes
- Internal 1024-Bytes RX FIFO and 1024-Bytes TX FIFO
- Supports card insertion and removal interrupt
- Supports hardware CRC generation and error detection
- Supports descriptor-based internal DMA controller

### 2.2.3 Video Engine

- Video decoding
  - H.265 MP@L5.0 up to 4K@30fps
  - H.264 BP/MP/HP@L5.0 up to 4K@24fps
  - H263 BP up to 1080p@60fps
  - MPEG-4 SP/ASP L5 up to 1080p@60fps
  - MPEG-2 MP/HL up to 1080p@60fps
  - MPEG-1 MP/HL up to 1080p@60fps
  - JPEG/Xvid/Sorenson Spark up to 1080p@60fps
  - MJPEG up to 1080p@30fps
- Video encoding
  - JPEG/MJPEG up to 1080p@60fps
  - Supports input picture scaler up/down

### 2.2.4 Video and Graphics

#### 2.2.4.1 Display Engine (DE)

- Output size up to 2048 x 2048
- Supports two alpha blending channels for main display and one channel for aux display
- Supports four overlay layers in each channel, and has an independent scaler
- Supports potter-duff compatible blending operation
- Supports LBC buffer decoder
- Supports dither output to TCON
- Supports input format Semi-planar YUV422/YUV420/YUV411 and Planar YUV422/YUV420/YUV411, ARGB8888/XRGB8888/RGB888/ARGB4444/ARGB1555/RGB565/palette
- Supports SmartColor2.0 for excellent display experience
  - Adaptive detail/edge enhancement
  - Adaptive color enhancement
  - Adaptive contrast enhancement and fresh tone rectify
- Supports write back for aux display

#### 2.2.4.2 De-interlacer (DI)

- Supports YUV420 (Planar/NV12/NV21) and YUV422 (Planar/NV16/NV61) data format
- Supports video resolution from 32x32 to 2048x1280 pixel
- Supports Inter-field interpolation/motion adaptive de-interlace method
- Performance: module clock 600M for 1080p@60Hz YUV420

#### 2.2.4.3 Graphic 2D (G2D)

- Supports layer size up to 2048 x 2048 pixels
- Supports pre-multiply alpha image data
- Supports color key
- Supports two pipes Porter-Duff alpha blending
- Supports multiple video formats 4:2:0, 4:2:2, 4:1:1 and multiple pixel formats (8/16/24/32 bits graphics layer)
- Supports memory scan order option
- Supports any format convert function
- Supports 1/16× to 32× resize ratio
- Supports 32-phase 8-tap horizontal anti-alias filter and 32-phase 4-tap vertical anti-alias filter
- Supports window clip
- Supports FillRectangle, BitBlit, StretchBlit and MaskBlit
- Supports horizontal and vertical flip, clockwise 0/90/180/270 degree rotate for normal buffer
- Supports horizontal flip, clockwise 0/90/270 degree rotate for LBC buffer

### 2.2.5 Video Output

#### 2.2.5.1 LVDS LCD

- Supports LVDS interface with single link, up to 1366 x 768@60fps

### 2.2.5.2 MIPI DSI

- Compliance with MIPI DSI v1.01
- Supports 4-lane MIPI DSI, up to 1280 x 720@60fps and 1920 x 1200@60fps resolution
- Supports non-burst mode with sync pulse/sync event and burst mode
- Supports pixel format: RGB888, RGB666, RGB666 loosely packed and RGB565
- Supports continuous lane clock modes
- Supports bidirectional communication of all generic commands in LP through data lane 0
- Supports low power data transmission
- Supports ULPS and escape modes
- Hardware checksum capabilities

### 2.2.5.3 HDMI

- Compatible with HDMI 1.4
- Supports DDC
- Integrated CEC hardware engine
- Optional color space converter (CSC): RGB (4:4:4) to/from YCbCr (4:4:4 or 4:2:2)
- Video formats:
  - Optional HDMI 1.4b video formats
  - All CEA-861-E video formats up to 1080p at 120 Hz
  - HDMI 1.4b 4K x 2K video formats
  - HDMI 1.4b 3D video modes with up to 340 MHz (TMDS clock)
- Audio formats:
  - Uncompressed audio formats: IEC60985 L-PCM audio samples, up to 192 kHz
  - Compressed audio formats: IEC61937 compressed audio, up to 768 kHz

## 2.2.6 System Peripherals

### 2.2.6.1 Timer

- The timer module implements the timing and counting functions, which includes timer0, timer1, watchdog, and audio video synchronization (AVS)

- The timer0/timer1 is a 32-bit down counter. The timer0 and timer1 are completely consistent
- The watchdog is used to transmit a reset signal to reset the entire system when an exception occurs in the system
- The AVS is used to synchronize the audio and video. The AVS sub-block includes AVS0 and AVS1, which are completely consistent

#### 2.2.6.2 High Speed Timer (HSTimer)

- The HSTimer module consists of HSTimer0 and HSTimer1. HSTimer0 and HSTimer1 are down counters that implement timing and counting functions. They are completely consistent.
- Configurable 56-bit down timer
- Supports 5 prescale factors
- The clock source is synchronized with AHB0 clock, much more accurate than other timers
- Supports 2 working modes: periodic mode and single counting mode
- Generates an interrupt when the count is decreased to 0

#### 2.2.6.3 GIC

- Supports 16 Software Generated Interrupts (SGIs), 16 Private Peripheral Interrupts (PPIs), and 192 Shared Peripheral Interrupts (SPIs)
- Software-configurable interrupts can be:
  - Enabled or disabled
  - Assigned to one of two groups: Group 0 or Group 1
  - Prioritized
  - Signaled to different processors in multiprocessor implementations
  - Either level-sensitive or edge-triggered
- GIC security extensions
  - Uses Group 0 interrupts as Secure interrupts, and Group 1 interrupts as Non-secure interrupts
  - Uses the FIQ interrupt request to signal Secure interrupts to a connected processor. The GIC-400 always signals Group 1 interrupts using the IRQ interrupt request

#### 2.2.6.4 DMAC

- Up to 16-ch DMA
- Provides 32 peripheral DMA requests for data reading and 32 peripheral DMA requests for data writing
- Flexible data width of 8/16/32/64-bit
- Programmable DMA burst length
- Supports linear and IO address modes
- Supports data transfer types with memory-to-memory, memory-to-peripheral, peripheral-to-memory, peripheral-to-peripheral
- Supports transfer with linked list
- DRQ response includes waiting mode and handshake mode
- DMA channel supports pause function
- Memory devices support non-aligned transform

#### 2.2.6.5 Clock Controller Unit (CCU)

- 8 PLLs
- One on-chip RC oscillator
- Supports one external 24 MHz DCXO and one external 32.768 kHz oscillator
- Supports clock configuration and clock generation for corresponding modules
- Supports software-controlled clock gating and software-controlled reset for corresponding modules

#### 2.2.6.6 Thermal Sensor Controller (THS)

- One thermal sensor located in CPU
- Temperature accuracy:  $\pm 3^{\circ}\text{C}$  from  $0^{\circ}\text{C}$  to  $+100^{\circ}\text{C}$ ,  $\pm 5^{\circ}\text{C}$  from  $-25^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$
- Averaging filter for thermal sensor reading
- Supports over-temperature protection interrupt and over-temperature alarm interrupt

### 2.2.6.7 LDO Power

- Integrated 2 LDOs (LDOA, LDOB)
- LDOA: 1.8 V power output, LDOB: 1.35 V/1.5 V/1.8 V power output
- LDOA for IO and analog module, LDOB for SDRAM
- Input voltage is 2.4 V to 3.6 V

### 2.2.6.8 RTC

- Implements time counter and timing wakeup
- Provides a 16-bit counter for counting day, 5-bit counter for counting hour, 6-bit counter for counting minute, 6-bit counter for counting second
- External connect a 32.768 kHz low-frequency oscillator for count clock
- Timer frequency is 1 kHz
- Configurable initial value by software anytime
- Supports timing alarm, and generates interrupt and wakeup the external devices
- 8 general purpose registers for storing power-off information

### 2.2.6.9 I/O Memory Management Unit (IOMMU)

- Supports virtual address to physical address mapping by hardware implementation
- Supports VE, DE, G2D, DI parallel address mapping
- Supports VE, DE, G2D, DI bypass function independently
- Supports VE, DE, G2D, DI pre-fetch independently
- Supports VE, DE, G2D, DI interrupt handing mechanism independently
- Supports 2 levels TLB (level1 TLB for special using, and level2 TLB for sharing)
- Supports TLB Fully cleared and Partially disabled
- Supports trigger PTW behavior when TLB miss
- Supports checking the permission

## 2.2.7 Audio Interfaces

### 2.2.7.1 Audio Codec

- Two audio digital-to-analog converter (DAC) channels
  - Supports 16-bit and 20-bit sample resolution
  - 8 kHz to 192 kHz DAC sample rate
  - $95 \pm 2$  dB SNR@A-weight,  $-85 \pm 3$  dB THD+N
- One audio output: One stereo headphone output: HPOUTL/R
- Three audio analog-to-digital converter (ADC) channels
  - Supports 16-bit and 20-bit sample resolution
  - 8 kHz to 48 kHz ADC sample rate
  - $95 \pm 3$  dB SNR@A-weight,  $-80 \pm 3$  dB THD+N
- Supports Dynamic Range Controller adjusting the DAC playback and ADC recording
- One 128x20-bits FIFO for DAC data transmit, one 128x20-bits FIFO for ADC data receive
- Programmable FIFO thresholds
- Supports interrupts and DMA
- Internal HPLDO output for HPVCC
- Internal ALDO output for AVCC

### 2.2.7.2 I2S/PCM

- Two I2S/PCM external interfaces (I2S1, I2S2) for connecting external power amplifier and MIC ADC
- Compliant with standard Philips Inter-IC sound (I2S) bus specification
  - Left-justified, Right-justified, PCM mode, and Time Division Multiplexing (TDM) format
  - Programmable PCM frame width: 1 BCLK width (short frame) and 2 BCLKs width (long frame)
- Transmit and Receive data FIFOs
  - Programmable FIFO thresholds
  - 128 depth x 32-bit width TXFIFO and 64 depth x 32-bit width RXFIFO
- Supports multiple function clock
  - Clock up to 24.576 MHz Data Output of I2S/PCM in Master mode (Only if the IO PAD and Peripheral I2S/PCM satisfy Timing Parameters)

- Clock up to 12.288 MHz Data Input of I2S/PCM in Master mode
- Supports TX/RX DMA slave interface
- Supports multiple application scenarios
  - Up to 16 channels ( $f_s = 48$  kHz) which has adjustable width from 8-bit to 32-bit
  - Sample rate from 8 kHz to 384 kHz (CHAN = 2)
  - 8-bit u-law and 8-bit A-law companded sample
- Supports master/slave mode

#### 2.2.7.3 DMIC

- Supports maximum 8 digital PDM microphones
- Supports sample rate from 8 kHz to 48 kHz

#### 2.2.7.4 One Wire Audio (OWA)

- One OWA TX
- Compliance with S/PDIF interface
- IEC-60958 transmitter functionality
- Supports 16-bit, 20-bit, and 24-bit data formats
- One 128×24bits TXFIFO for audio data transfer
- Programmable FIFO thresholds
- Supports TX DMA slave interface
- Function clock includes 24.576 MHz and 22.579 MHz frequency
- Hardware parity generation on the transmitter
- Supports channel status insertion for the transmitter
- Supports interrupts and DMA

## 2.2.8 Security System

### 2.2.8.1 Crypto Engine (CE)

- Supports Symmetrical algorithm for encryption and decryption: AES, DES, TDES
  - Supports ECB, CBC, CTS, CTR, CFB, OFB mode for AES
  - Supports 128/192/256-bit key for AES
  - Supports ECB, CBC, CTR mode for DES/TDES
- Supports Hash algorithm for tamper proofing: MD5, SHA, HMAC
  - Supports SHA1, SHA224, SHA256, SHA384, SHA512 for SHA
  - Supports HMAC-SHA1, HMAC-SHA256 for HMAC
  - Supports multi-package mode for MD5/SHA1/SHA224/SHA256/SHA384/SHA512
- Supports Asymmetrical algorithm for signature verification: RSA
  - RSA supports 512/1024/2048-bit width
- Supports 160-bit hardware PRNG with 175-bit seed
- Supports 256-bit hardware TRNG
- Internal DMA controller for data transfer with memory
- Supports secure and non-secure interfaces respectively

### 2.2.8.2 Security ID (SID)

- Supports 2 Kbits eFuse for chip ID and security application
- The eFuse has secure zone and non-secure zone
- Backup eFuse information by using SID\_SRAM
- Burning the key to the SID
- Reading the key use status in the SID
- Loading the key to the CE

### 2.2.8.3 Secure Memory Control (SMC)

- The SMC is always secure, only secure CPU can access the SMC

- Sets secure area of DRAM
- Sets secure property that Master accesses to DRAM

### 2.2.8.4 Secure Peripherals Control (SPC)

- The SPC is always secure, only secure CPU can access the SPC
- Sets secure property of peripherals

## 2.2.9 External Peripherals

### 2.2.9.1 USB DRD

- One USB 2.0 DRD (USB0), with integrated USB 2.0 analog PHY
- Complies with USB2.0 Specification
- Supports USB Host function
  - Compatible with Enhanced Host Controller Interface (EHCI) Specification, Version 1.0
  - Compatible with Open Host Controller Interface (OHCI) Specification, Version 1.0a
  - Supports High-Speed (HS, 480 Mbit/s), Full-Speed (FS, 12 Mbit/s), and Low-Speed (LS, 1.5 Mbit/s)
  - Supports only 1 USB Root port shared between EHCI and OHCI
- Supports USB Device function
  - Supports High-Speed (HS, 480 Mbit/s), Full-Speed (FS, 12 Mbit/s)
  - Supports bi-directional endpoint0 (EP0) for Control transfer
  - Up to 10 user-configurable endpoints (EP1+, EP1-, EP2+, EP2-, EP3+, EP3-, EP4+, EP4-, EP5+, EP5-) for Bulk transfer, Isochronous transfer and Interrupt transfer
  - Up to (8 KB + 64 Bytes) FIFO for all EPs (including EP0)
  - Supports interface to an external Normal DMA controller for every EP
- Supports an internal DMA controller for data transfer with memory
- Supports High-Bandwidth Isochronous & Interrupt transfers
- Automated splitting/combining of packets for Bulk transfers
- Supports point-to-point and point-to-multipoint transfer in both Host and Peripheral modes
- Includes automatic ping capabilities

- Soft connect/disconnect function
- Performs all transaction scheduling in hardware
- Power optimization and power management capabilities
- Device and host controller share a 8K SRAM and a physical PHY

### 2.2.9.2 USB HOST

- One USB 2.0 HOST (USB1), with integrated USB 2.0 analog PHY
- Complies with USB2.0 Specification
- Supports USB2.0 Host function
  - Compatible with Enhanced Host Controller Interface (EHCI) Specification, Version 1.0
  - Compatible with Open Host Controller Interface (OHCI) Specification, Version 1.0a
  - Supports High-Speed (HS, 480 Mbit/s), Full-Speed (FS, 12 Mbit/s) and Low-Speed (LS, 1.5 Mbit/s) Device
  - Supports only 1 USB Root port shared between EHCl and OHCI
- An internal DMA Controller for data transfer with memory

### 2.2.9.3 EMAC

- One EMAC interface for connecting external Ethernet PHY
- 10/100/1000 Mbit/s Ethernet port with RGMII and RMII interfaces
- Compliant with IEEE 802.3-2002 standard
- Supports both full-duplex and half-duplex operation
- Provides the management data input/output (MDIO) interface for PHY device configuration and management with configurable clock frequencies
- Programmable frame length to support Standard or Jumbo Ethernet frames with sizes up to 16 KB
- Supports a variety of flexible address filtering modes
- Separate 32-bit status returned for transmission and reception packets
- Optimization for packet-oriented DMA transfers with frame delimiters
  - Supports linked-list descriptor list structure

- Descriptor architecture, allowing large blocks of data transfer with minimum CPU intervention; each descriptor can transfer up to 4 KB of data
- Comprehensive status reporting for normal operation and transfers with errors
- 4 KB TXFIFO for transmission packets and 16 KB RXFIFO for reception packets
- Programmable interrupt options for different operational conditions

#### 2.2.9.4 UART

- Up to 6 UART controllers (UART0, UART1, UART2, UART3, UART4, UART5)
- UART0, UART4, UART5: 2-wire; UART1, UART2, UART3: 4-wire
- Compatible with industry-standard 16450/16550 UARTs
- Supports IrDA-compatible slow infrared (SIR) format
- Two separate FIFOs: one is RX FIFO, and the other is TX FIFO
  - Each of them is 64 bytes (For UART0)
  - Each of them is 256 bytes (For UART1, UART2, UART3, UART4, and UART5)
- The working reference clock is from the APB bus clock
  - Speed up to 4 Mbit/s with 64 MHz APB clock
  - Speed up to 1.5 Mbit/s with 24 MHz APB clock
- 5 to 8 data bits for RS-232 characters, or 9 bits RS-485 format
- 1, 1.5 or 2 stop bits
- Programmable parity (even, odd, or no parity)
- Supports TX/RX DMA slave controller interface
- Supports software/hardware flow control
- Supports RX DMA Master interface (Only for UART1)
- Supports auto-flow by using CTS & RTS (Only for UART1/2/3)

#### 2.2.9.5 Two Wire Interface (TWI)

- Up to 4 TWI controllers (TWI0, TWI1, TWI2, TWI3)
- Compliant with I2C bus standard

- Supports standard mode (up to 100 kbit/s) and fast mode (up to 400 kbit/s)
- Supports 7-bit and 10-bit device addressing modes
- Supports master mode or slave mode
- Master mode features:
  - Supports the bus arbitration in the case of multiple master devices
  - Supports clock synchronization and bit and byte waiting
  - Supports packet transmission and DMA
- Slave mode features:
  - Interrupt on address detection
- The TWI controller includes one TWI engine and one TWI driver. And the TWI driver supports packet transmission and DMA mode when TWI works in master mode

### 2.2.9.6 SPI and SPI\_DBI

- Up to 2 SPI controllers (SPI0, SPI1)
- The SPI0 only supports SPI mode; The SPI1 supports SPI mode and display bus interface (DBI) mode
- SPI mode:
  - Full-duplex synchronous serial interface
  - Master/slave configurable
  - Mode0 to Mode3 are supported for both transmit and receive operations
  - 8-bit wide by 64-entry FIFO for both transmit and receive data
  - Polarity and phase of the Chip Select (SPI-CS) and SPI Clock (SPI-CLK) are configurable
  - Supports 3-wire/4-wire SPI
  - Supports programmable serial data frame length: 1-bit to 32-bit
  - Supports Standard SPI, Dual-Output/Dual-Input SPI, Dual IO SPI, Quad-Output/Quad-Input SPI
- DBI mode:
  - Supports DBI Type C 3 Line/4 Line Interface Mode
  - Supports 2 Data Lane Interface Mode
  - Supports RGB111/444/565/666/888 video format
  - Maximum resolution of RGB666 240 x 320@30Hz with single data lane

- Maximum resolution of RGB888 240 x 320@60Hz or 320 x 480@30Hz with dual data lane
- Supports Tearing effect
- Supports software flexible control video frame rate

#### 2.2.9.7 CIR Receiver (CIR\_RX)

- One CIR\_RX interface (IR-RX)
- Full physical layer implementation
- Supports NEC format infra data
- Supports CIR for remote control or wireless keyboard
- 64x8 bits FIFO for data buffer
- Sample clock up to 1 MHz

#### 2.2.9.8 CIR Transmitter (CIR\_TX)

- One CIR\_TX interface (IR-TX)
- Supports arbitrary wave generator
- Configurable carrier frequency
- Supports handshake mode and waiting mode of DMA
- 128 bytes FIFO for data buffer

#### 2.2.9.9 PWM

- Supports 8 independent PWM channels (PWM0 to PWM7)
  - Supports PWM continuous mode output
  - Supports PWM pulse mode output, and the pulse number is configurable
  - Output frequency range: 0 to 24 MHz or 100 MHz
  - Various duty-cycle: 0% to 100%
  - Minimum resolution: 1/65536
- Supports 4 complementary pairs output

- PWM01 pair (PWM0 + PWM1), PWM23 pair (PWM2 + PWM3), PWM45 pair (PWM4 + PWM5), PWM67 pair (PWM6 + PWM7)
- Supports dead-zone generator, and the dead-zone time is configurable
- Supports 4 group of PWM channel output for controlling stepping motors
  - Supports any plural channels to form a group, and output the same duty-cycle pulse
  - In group mode, the relative phase of the output waveform for each channel is configurable
- Supports 8 channels capture input
  - Supports rising edge detection and falling edge detection for input waveform pulse
  - Supports pulse-width measurement for input waveform pulse

#### 2.2.9.10 General Purpose ADC (GPADC)

- 1-ch Successive approximation register (SAR) analog-to-digital converter (ADC)
- 12-bit sampling resolution and 8-bit precision
- 64 FIFO depth of data register
- Power reference voltage: AVCC, analog input voltage range: 0 to AVCC
- Maximum sampling frequency up to 1 MHz
- Supports three operation modes: single conversion mode, continuous conversion mode, burst conversion mode

#### 2.2.10 LEDC

- LEDC is used to control the external intelligent control LED lamp
- Configurable LED output high/low level width
- Configurable LED reset time
- LEDC data supports DMA configuration mode and CPU configuration mode
- Maximum 1024 LEDs serial connect
- LED data transfer rate up to 800 kbit/s

### 2.2.11 Package

LFBGA 196 balls, 10 mm x 10 mm body size, 0.65 mm ball pitch, 0.35 mm ball size

## 2.3 Block Diagram

Figure 2-1 shows the system block diagram of the H133.

**Figure 2-1 H133 System Block Diagram**

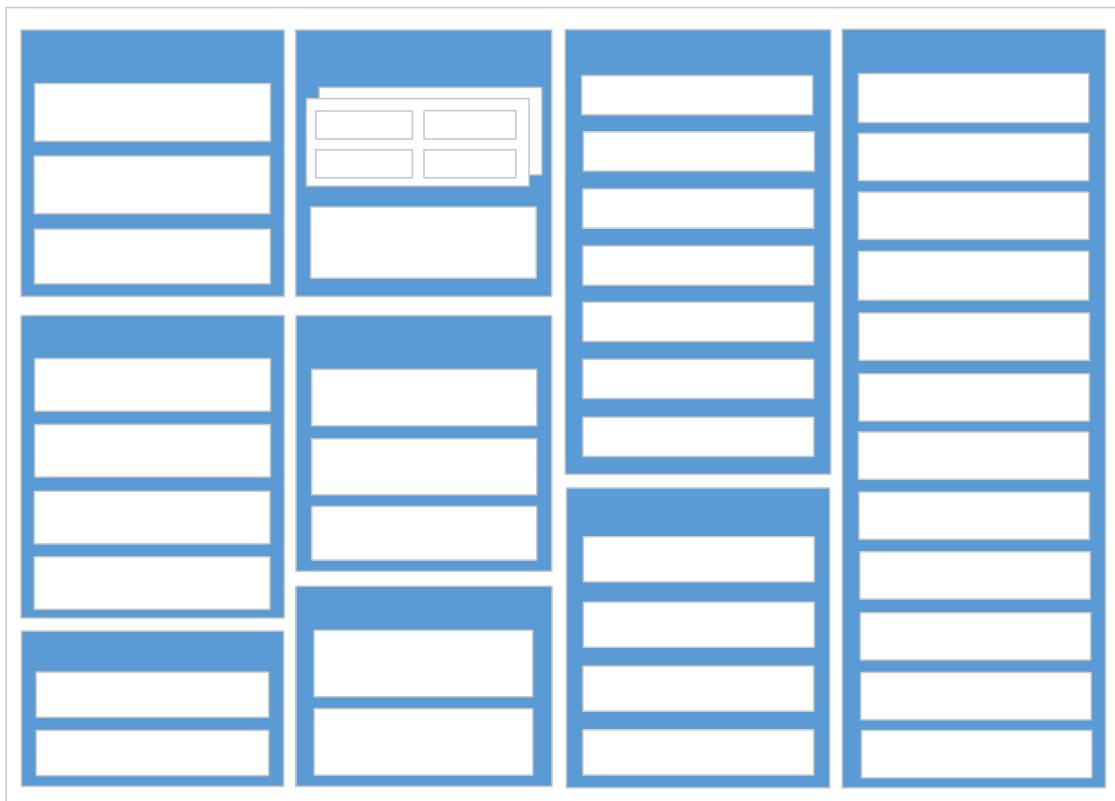
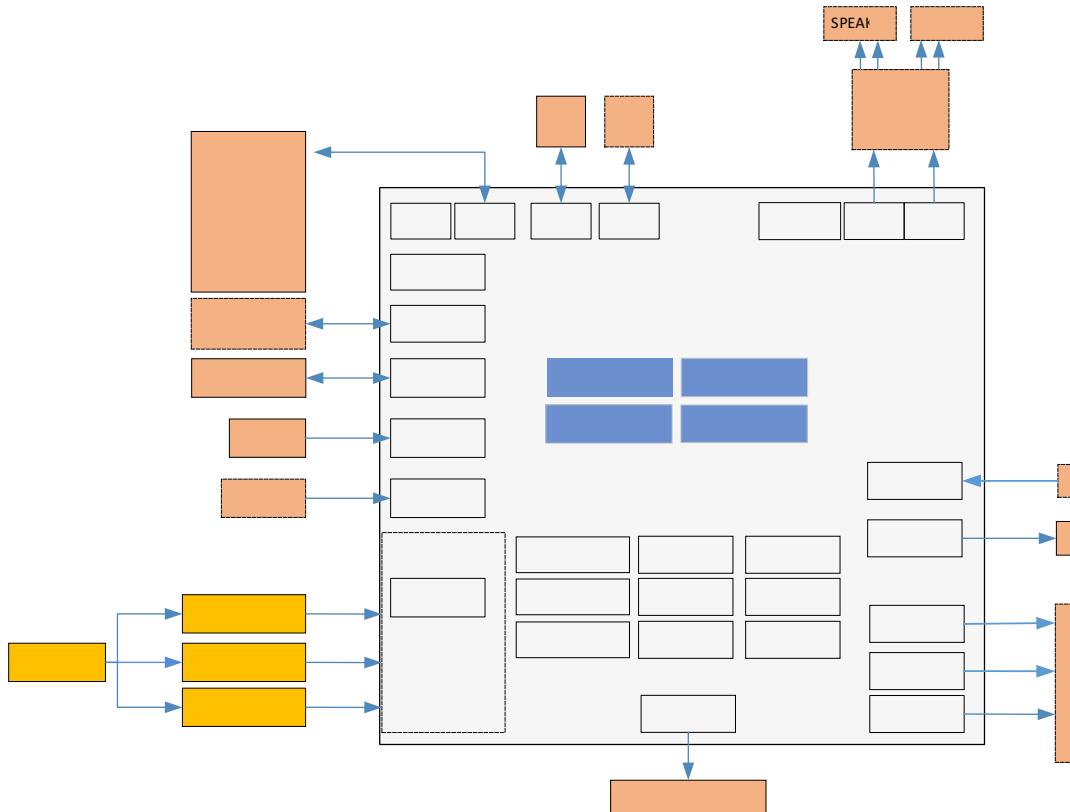


Figure 2-2 shows the typical solution diagram of the H133.

**Figure 2-2 Intelligent Speaker with Screen Solution of the H133**



# Contents

3	System.....	29
3.1	Memory Mapping .....	29
3.2	CPU Configuration.....	32
3.2.1	Overview .....	32
3.2.2	Functional Descriptions.....	32
3.2.3	Cluster 0 Configuration Register List.....	34
3.2.4	Cluster 0 Configuration Register Description.....	35
3.2.5	CPU Subsystem Control Register List .....	43
3.2.6	CPU Subsystem Control Register Description .....	44
3.3	Clock Controller Unit (CCU).....	48
3.3.1	Overview .....	48
3.3.2	Block Diagram .....	48
3.3.3	Functional Description .....	49
3.3.4	Programming Guidelines.....	55
3.3.5	Register List.....	61
3.3.6	Register Description .....	64
3.4	BROM System.....	146
3.4.1	Overview .....	146
3.4.2	Functional Description .....	146
3.5	System Configuration.....	157
3.5.1	Overview .....	157
3.5.2	Register List.....	157
3.5.3	Register Description .....	157
3.6	Timer .....	165
3.6.1	Overview .....	165
3.6.2	Block Diagram .....	166
3.7	High Speed Timer .....	184
3.7.1	Overview .....	184
3.7.2	Block Diagram .....	184

---

3.7.3	Functional Description .....	184
3.7.4	Programming Guidelines.....	187
3.7.5	Register List.....	188
3.7.6	Register Description .....	189
3.8	Generic Interrupt Controller (GIC) .....	196
3.8.1	Overview .....	196
3.8.2	Functional Description .....	196
3.9	Direct Memory Access Controller (DMAC).....	204
3.9.1	Overview .....	204
3.9.2	Block Diagram .....	204
3.9.3	Functional Description .....	205
3.9.4	Programming Guidelines.....	214
3.9.5	Register List.....	217
3.9.6	Register Description .....	218
3.10	Thermal Sensor Controller (THS) .....	239
3.10.1	Overview .....	239
3.10.2	Block Diagram .....	239
3.10.3	Functional Description .....	239
3.10.4	Programming Guidelines.....	241
3.10.5	Register List.....	243
3.10.6	Register Description .....	243
3.11	IOMMU .....	249
3.11.1	Overview .....	249
3.11.2	Block Diagram .....	249
3.11.3	Functional Description .....	251
3.11.4	Programming Guidelines.....	261
3.11.5	Register List.....	262
3.11.6	Register Description .....	266
3.12	RTC .....	338
3.12.1	`Overview .....	338
3.12.2	Block Diagram .....	338
3.12.3	Functional Description .....	339
3.12.4	Programming Guidelines.....	343

---

3.12.5 Register List.....	345
3.12.6 Register Description.....	346

# Figures

Figure 3-1 Default Power Domain of Cluster0 .....	34
Figure 3-2 CCU Block Diagram .....	48
Figure 3-3 System Bus Tree.....	49
Figure 3-4 Module Clock Generation .....	50
Figure 3-5 PLL Distribution.....	51
Figure 3-6 Boot Process in Normal BROM Mode.....	150
Figure 3-7 Boot Process in Secure BROM Mode.....	151
Figure 3-8 NON_CPU0 Boot Process Diagram .....	152
Figure 3-9 CPU0 Hotplug Process Diagram .....	153
Figure 3-10 Mandatory Upgrade Process .....	154
Figure 3-11 USB FEL Process .....	155
Figure 3-12 Timer Block Diagram.....	166
Figure 3-13 Block Diagram for the Timer.....	166
Figure 3-14 Block Diagram for the Watchdog.....	168
Figure 3-15 Block Diagram for the AVS.....	170
Figure 3-16 HSTimer Block Diagram .....	184
Figure 3-17 Typical Application for HSTimer.....	185
Figure 3-18 HSTimer Initialization Process.....	187
Figure 3-19 DMAC Block Diagram.....	204
Figure 3-20 DMAC Typical Application Diagram .....	206
Figure 3-21 DMA Descriptor .....	208
Figure 3-22 DMA Chain Transfer.....	210
Figure 3-23 Workflow of the DMAC Handshake Mode .....	213
Figure 3-24 DMAC Transfer Process .....	215
Figure 3-25 THS Block Diagram.....	239
Figure 3-26 Thermal Sensor Timing Requirement .....	240
Figure 3-27 Thermal Sensor Controller Interrupt Source .....	240
Figure 3-28 THS Initial Process .....	241
Figure 3-29 IOMMU Block Diagram .....	250
Figure 3-30 Internal Switch Process.....	255

---

Figure 3-31 VA-PA Switch Process .....	256
Figure 3-32 Invalid TLB Address Range .....	257
Figure 3-33 Level1 Page Table Format.....	259
Figure 3-34 Level1 Page Table Format.....	259
Figure 3-35 Read/Write Permission Control.....	260
Figure 3- 36 RTC Block Diagram .....	339
Figure 3-37 RTC Application Diagram .....	340
Figure 3-38 RTC Clock Tree .....	340
Figure 3-39 RTC Counter .....	341
Figure 3-40 RTC 1 kHz Counter Step Structure .....	342

# Tables

Table 3-1 Reset Signal Description.....	33
Table 3-2 Power Domain of Modules in the CPU Subsystem .....	33
Table 3-3 PLL Typical Application.....	52
Table 3-4 PLL Features .....	53
Table 3-5 The Usage Notes of PLL Frequencies .....	53
Table 3-6 Recommended Values for Configuration Factors of PLL_AUDIO1.....	56
Table 3-7 BOOT_MODE Setting.....	147
Table 3-8 GPIO Boot Select Configuration .....	147
Table 3-9 Groups of eFuse_Boot_Select.....	148
Table 3-10 eFuse Boot Select Setting.....	148
Table 3-11 Fast Boot Select Setting .....	156
Table 3-12 Interrupt Sources .....	196
Table 3-13 DMAC Sub-blocks.....	205
Table 3-14 DMA DRQ Type .....	206
Table 3-15 Source/Destination Address Distribution .....	209
Table 3-16 THS Information in the SID.....	241
Table 3-17 Correspondence Relation between Master and Module .....	250
Table 3-18 Relation between ACI and Domain .....	260
Table 3-19 RTC External Signals .....	339
Table 3-20 RTC Counter Changing Range.....	342

## 3 System

### 3.1 Memory Mapping

Module	Address (It is for Cluster CPU)	Size
<b>BROM &amp; SRAM</b>		
N-BROM	0x0000 0000—0x0000 BFFF	48 KB
S-BROM	0x0000 0000—0x0000 8FFF	36 KB
SRAM A1	0x0002 0000---0x0002 7FFF	32 KB
<b>VE_SYS Related</b>		
VE	0x01C0 E000---0x01C0 FFFF	8 KB
<b>SP0 (SYS Domain)</b>		
GPIO	0x0200 0000---0x0200 07FF	2 KB
SPC	0x0200 0800---0x0200 0BFF	1 KB
PWM	0x0200 0C00---0x0200 0FFF	1 KB
CCU	0x0200 1000---0x0200 1FFF	4 KB
IR_TX	0x0200 3000 – 0x0200 33FF	1 KB
TZMA	0x0200 4000---0x0200 43FF	1 KB
LEDC	0x0200 8000---0x0200 83FF	1 KB
GPADC	0x0200 9000---0x0200 93FF	1 KB
THS	0x0200 9400---0x0200 97FF	1 KB
IOMMU	0x0201 0000---0x0201 FFFF	64 KB
Audio Codec	0x0203 0000---0x0203 0FFF	4 KB
DMIC	0x0203 1000---0x0203 13FF	1 KB
I2S1	0x0203 3000---0x0203 3FFF	4 KB
I2S2	0x0203 4000---0x0203 4FFF	4 KB
OWA	0x0203 6000---0x0203 63FF	1 KB
TIMER	0x0205 0000---0x0205 0FFF	4 KB
<b>SP1 (SYS Domain)</b>		
UART0	0x0250 0000---0x0250 03FF	1 KB
UART1	0x0250 0400---0x0250 07FF	1 KB
UART2	0x0250 0800---0x0250 0BFF	1 KB
UART3	0x0250 0C00---0x0250 0FFF	1 KB
UART4	0x0250 1000---0x0250 13FF	1 KB

<b>Module</b>	<b>Address (It is for Cluster CPU)</b>	<b>Size</b>
UART5	0x0250 1400---0x0250 17FF	1 KB
TWI0	0x0250 2000---0x0250 23FF	1 KB
TWI1	0x0250 2400---0x0250 27FF	1 KB
TWI2	0x0250 2800---0x0250 2BFF	1 KB
TWI3	0x0250 2C00---0x0250 2FFF	1 KB
<b>SH0 (SYS Domain)</b>		
SYSCTRL	0x0300 0000---0x0300 0FFF	4 KB
DMAC	0x0300 2000---0x0300 2FFF	4 KB
SID	0x0300 6000---0x0300 6FFF	4 KB
SMC	0x0300 7000---0x0300 7FFF	4 KB
HSTIMER	0x0300 8000---0x0300 8FFF	4 KB
DCU	0x0301 0000---0x0301 FFFF	64 KB
CPU_GIC	0x0302 0000---0x0302 FFFF	64 KB
CE_NS	0x0304 0000---0x0304 07FF	2 KB
CE_S	0x0304 0800---0x0304 0FFF	2 KB
CE_KEY_SRAM	0x0304 1000---0x0304 1FFF	4 KB (only CE access)
MSI+MEMC	0x0310 2000---0x0330 1FFF	2 MB
<b>SH2 (SYS Domain)</b>		
SMHCO	0x0402 0000---0x0402 0FFF	4 KB
SMHC1	0x0402 1000---0x0402 1FFF	4 KB
SMHC2	0x0402 2000---0x0402 2FFF	4 KB
SPI0	0x0402 5000---0x0402 5FFF	4 KB
SPI1	0x0402 6000---0x0402 6FFF	4 KB
USB0	0x0410 0000---0x041F FFFF	1 MB
USB1	0x0420 0000---0x042F FFFF	1 MB
EMAC	0x0450 0000---0x0450 FFFF	64 KB
<b>VIDEO_OUT_SYS Related (SYS Domain)</b>		
DE	0x0500 0000---0x053F FFFF	4 MB
DI	0x0540 0000---0x0540 FFFF	64 KB
G2D	0x0541 0000---0x0544 FFFF	256 KB
DSI	0x0545 0000---0x0545 1FFF	8 KB
DISPLAY_TOP	0x0546 0000---0x0546 0FFF	4 KB
TCON_LCD0	0x0546 1000---0x0546 1FFF	4 KB

Module	Address (It is for Cluster CPU)	Size
TCON_TV0	0x0547 0000---0x0547 0FFF	4 KB
HDMI	0x0550 0000---0x055F FFFF	1M
<b>APBS0 (CPUS Domain)</b>		
R_CPUCFG	0x0700 0400---0x0700 0BFF	2 KB
R_PPU	0x0700 1000---0x0700 13FF	1 KB
R_SPC	0x0700 2000---0x0700 23FF	1 KB
R_PRCM	0x0701 0000---0x0701 FFFF	64 KB
R_TIMER	0x0702 0000 – 0x0702 03FF	1 KB
R_TWD	0x0702 0800 – 0x0702 0BFF	1 KB
IR_RX	0x0704 0000---0x0704 03FF	1 KB
<b>AHBS (CPUS Domain)</b>		
RTC	0x0709 0000---0x0709 03FF	1 KB
<b>CPUX Related</b>		
CPU_SYS_CFG	0x0810 0000---0x0810 03FF	1 KB
TimeStamp_STA	0x0811 0000---0x0811 0FFF	4 KB
TimeStamp_CTRL	0x0812 0000---0x0812 0FFF	4 KB
IDC	0x0813 0000---0x0813 0FFF	4 KB
C0_CPUX_CFG	0x0901 0000---0x0901 03FF	1 KB
C0_CPUX_MBIST	0x0902 0000---0x0902 0FFF	4 KB
<b>DRAM Space (SYS Domain)</b>		
DRAM SPACE	0x4000 0000---0xBFFF FFFF	2 GB

## 3.2 CPUX Configuration

### 3.2.1 Overview

CPUX indicates the ARM CPU subsystem. You can configure the CPUX via the C0\_CPUX\_CFG and CPU\_SUBSYS\_CTRL modules.

The C0\_CPUX\_CFG module is used for configuring Cluster0. Cluster0 is the only cluster which consists of dual-core Cortex™-A7, 32 KB I-cache and 32 KB D-cache per core, and the shared 256 KB L2 cache. The C0\_CPUX\_CFG module includes the following features:

- CPU reset system: core reset, debug circuit reset, and other functional modules reset
- CPU related control: interface control, CP15 control, and power-on/off control
- CPU status check: idle status, SMP status, interrupt status, and so on
- CPU debug-related register for control and status

The CPU\_SUBSYS\_CTRL module is used for controlling CPU subsystem resources, such as GIC-400 and JTAG.

### 3.2.2 Functional Descriptions

#### 3.2.2.1 L2 Idle Mode

Enter the L2 idle mode by configuring the L2 cache of Cluster to enter the wait for interrupt (WFI) mode.

Follow the steps below:

1. Make sure CPU[1:0] of Cluster has entered the WFI mode. You can check it through bit[17:16] of [Cluster0 CPU Status Register](#).
2. Pull high the [ACINACTM](#) of Cluster by writing 1 to the bit0 of [Cluster 0 Control Register1](#).
3. Check whether L2 enters idle status by checking whether the [STANDBYWFL2](#) is high.

To exit the L2 idle mode, set the [ACINACTM](#) to low.

#### 3.2.2.2 CPU Reset System

There are three levels of CPUX reset, with the scope size from small to large: core Reset, power-on Reset, and H\_Reset.

The following table describes all the reset signals in the CPUX reset system.

**Table 3-1 Reset Signal Description**

Reset Signal	Description
CORE_RST	This is the primary reset signal which resets the corresponding core logic, including NEON, VFP, Debug, ETM, breakpoint, and watchpoint logic. It maps to a warm reset that covers the reset of the processor logic.
PWRON_RST	This power-on reset signal resets all the processor logic, including the Debug, ETM trace unit, breakpoint, watchpoint logic, and performance monitors logic. It maps to a cold reset that covers the reset of the processor logic and the integrated debug functionality. It does not reset debug logic in the debug power domain. PWRON_RST includes CORE_RST, ETM_RST, and DBG_RST.
H_RST	H_RST includes PWRON_RST, L2_RST, MBIST_RST, SOC_DBG_RST, and CO_CPUX_CFG.
AXI2MBUS_RST	This reset signal resets the AXI2MBUS interface logic circuit.
L2_RST	This single, cluster-wide signal resets the L2 memory system and the logic in the SCU.
ETM_RST	This reset signal resets ETM debug logic circuit.
DBG_RST	In the processor power domain, this reset signal resets only the debug, breakpoint, and watchpoint logic. In the debug power domain, it resets the debug logic for each processor.
SOC_DBG_RST	This reset signal resets all the debug logic. It includes DBG_RST.
MBIST_RST	This reset signal resets all resettable registers in the cluster, for entry into and exit from MBIST mode.
CPU_SUBSYS_RST	CPU_SUBSYS_RST includes CO_H_RST, GIC-400, and CPU_SUBSYS_CTRL.

### 3.2.2.3 CPUX Power Block Diagram

The following table shows the power domain of the modules in the CPU subsystem.

**Table 3-2 Power Domain of Modules in the CPU Subsystem**

Power Domain	Modules	Description
Cluster0	Cluster0, CO_CPUX_CFG, and CO_MBIST	Cluster0 circuit, CO_CPUX_CFG module, and CPU reset/power/mbist circuits
System	Timestamp, GIC, CPU_SUBSYS_CTRL, and Clock	Provides system source of CPU sub-system

As shown in the table, CO\_CPUX\_CFG belongs to Cluster0 power domain and CPU\_SUBSYS\_CTRL belongs to the system power domain.

Because CO\_CPUX\_CFG and Cluster0 belong to the same power domain, CO\_CPUX\_CFG holds in the default state when Cluster0 restarts. You need to initialize CO\_CPUX\_CFG via software after CO\_H\_RST is de-asserted.

The following figure shows the default power domain of Cluster0.

**Figure 3-1 Default Power Domain of Cluster0**

For Cluster0, the power switch of all CPU cores are on, the PWRON\_RST of all CPU cores are de-asserted, the CORE\_RST of CPU0 is de-asserted, and the CORE\_RST of CPU1 is asserted.

#### 3.2.2.4 CPUX Power Block Diagram

The CPU-related operations, such as opening/closing cores, clustering switches, and querying status, require proper configurations of C0\_CPUX\_CFG module and the related system control resources like the bus and clock.

#### 3.2.3 Cluster 0 Configuration Register List

Module Name	Base Address
C0_CPUX_CFG	0x09010000

Register Name	Offset	Description
C0_RST_CTRL	0x0000	Cluster 0 Reset Control Register
C0_CTRL_REG0	0x0010	Cluster 0 Control Register0
C0_CTRL_REG1	0x0014	Cluster 0 Control Register1
C0_CTRL_REG2	0x0018	Cluster 0 Control Register2
CACHE_CFG_REG	0x0024	Cache Configuration Register
C0_CPU_STATUS	0x0080	Cluster 0 CPU Status Register
L2_STATUS_REG	0x0084	Cluster 0 L2 Status Register
DBG_REG0	0x00C0	Cluster 0 Debug Control Register0
DBG_REG1	0x00C4	Cluster 0 Debug Control Register1

Register Name	Offset	Description
AXI_MNT_CTRL_REG	0x00D0	AXI Monitor Control Register
AXI_MNT_PRD_REG	0x00D4	AXI Monitor Period Register
AXI_MNT_RLTCY_REG	0x00D8	AXI Monitor Read Total Latency Register
AXI_MNT_WLTCY_REG	0x00DC	AXI Monitor Write Total Latency Register
AXI_MNT_RREQ_REG	0x00E0	AXI Monitor Read Request Times Register
AXI_MNT_WREQ_REG	0x00E4	AXI Monitor Write Request Times Register
AXI_MNT_RBD_REG	0x00E8	AXI Monitor Read Bandwidth Register
AXI_MNT_WBD_REG	0x00EC	AXI Monitor Write Bandwidth Register

### 3.2.4 Cluster 0 Configuration Register Description

#### 3.2.4.1 0x0000 Cluster 0 Reset Control Register (Default Value: 0x13FF\_0101)

Offset: 0x0000			Register Name: C0_RST_CTRL
Bit	Read/Write	Default/Hex	Description
31:29	/	/	/
28	R/W	0x1	DDR_RST AXI2MBUS Logic Circuit Reset 0: Assert 1: De-assert
27:26	/	/	/
25	R/W	0x1	MBIST_RST CPUBIST Reset The reset signal is for test. 0: Assert 1: De-assert
24	R/W	0x1	SOC_DBG_RST Cluster SoC Debug Reset 0: Assert 1: De-assert
23:20	R/W	0xF	Reserved
19:16	R/W	0xF	DBG_RST Cluster CPU[1:0] Debug Reset Assert (The bit[19:18] is not used) The meaning of each bit: 0: Assert 1: De-assert

Offset: 0x0000			Register Name: C0_RST_CTRL
Bit	Read/Write	Default/Hex	Description
15:9	/	/	/
8	R/W	0x1	L2_RST Cluster L2 Cache Reset 0: Assert 1: De-assert
7:4	/	/	/
3:0	R/W	0x1	CORE_RESET Cluster CPU[1:0] Reset Assert (The bit[3:2] is not used) The meaning of each bit: 0: Assert 1: De-assert

### 3.2.4.2 0x0010 Cluster 0 Control Register0 (Default Value: 0x8000\_0000)

Offset: 0x0010			Register Name: C0_CTRL_REG0
Bit	Read/Write	Default/Hex	Description
31	R/W	0x1	SYSBAR_DISABLE Disable broadcasting barriers to the system bus. 0: Barriers are broadcasted to the system bus. This requires an AMBA4 interconnection. 1: Barriers are not broadcasted to the system bus. This is compatible with an AXI3 interconnection.
30	R/W	0x0	BROADCAST_INNER Enable broadcasting inner shareable transactions. 0: Inner shareable transactions are not broadcasted externally. 1: Inner shareable transactions are broadcasted externally.
29	R/W	0x0	BROADCAST_OUTER Enable broadcasting outer shareable transactions. 0: Outer Shareable transactions are not broadcasted externally. 1: Outer Shareable transactions are broadcasted externally.

Offset: 0x0010			Register Name: C0_CTRL_REG0
Bit	Read/Write	Default/Hex	Description
28	R/W	0x0	<p>BROADCAST_CACHE_MAINT Enable broadcasting cache maintenance operations to downstream caches.</p> <p>0: Cache maintenance operations are not broadcasted to downstream caches.</p> <p>1: Cache maintenance operations are broadcasted to downstream caches.</p>
27:12	/	/	/
11:8	R/W	0x0	<p>CP15S_DISABLE Disable write access to some secure CP15 registers</p> <p>The meaning of each bit:</p> <p>0: Enable write access to CP15 register</p> <p>1: Disable write access to CP15 register</p>
7:5	/	/	/
4	R/W	0x0	<p>L2_RST_DISABLE Disable automatic L2 cache invalidation at reset.</p> <p>0: L2 cache is reset by hardware.</p> <p>1: L2 cache is not reset by hardware.</p>
3:0	R/W	0x0	<p>L1_RST_DISABLE Disable automatic Cluster CPU[1:0] L1 cache invalidation at reset. (The bit[3:2] is not used)</p> <p>The meaning of each bit:</p> <p>0: L1 cache is reset by hardware.</p> <p>1: L1 cache is not reset by hardware.</p>

### 3.2.4.3 0x0014 Cluster 0 Control Register1 (Default Value: 0x0000\_0000)

Offset: 0x0014			Register Name: C0_CTRL_REG1
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1	R/W	0x0	<p>CRM_LowFreq_AutoSelect Enable CRM to automatically select low frequency.</p> <p>0: Disabled</p> <p>1: Enabled</p>

Offset: 0x0014			Register Name: C0_CTRL_REG1
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	ACINACTM Deactivate snoop interface 0: Snoop interface is active 1: Snoop interface is inactive and no longer accepts requests

### 3.2.4.4 0x0018 Cluster 0 Control Register2 (Default Value: 0x0000\_0010)

Offset: 0x0018			Register Name: C0_CTRL_REG2
Bit	Read/Write	Default/Hex	Description
31:25	/	/	/
24	R/W	0x0	EVENTI Event input for processor to wake up from WFE state. This bit must remain high for at least one clock cycle to be visible by the cores.
23:20	R/W	0x0	EXM_CLR[1:0] Clear the status of the corresponding interface. (The bit[23:22] is not used) The meaning of each bit: 0: Not to clear 1: Clear
19:0	/	/	/

### 3.2.4.5 0x0024 Cache Configuration Register (Default Value: 0x0018\_001A)

Offset: 0x0024			Register Name: CACHE_CFG_REG
Bit	Read/Write	Default/Hex	Description
31:22	/	/	/
21:19	R/W	0x3	EMA_L2D Indicates the port number of L2 Cache SRAM EMA control port.
18:17	R/W	0x0	EMAW_L2D Indicates the port number of L2 Cache SRAM EMAW control port.
16	R/W	0x0	EMAS_L2D Indicates the port number of L2 Cache SRAM EMAS control port.
15:12	/	/	/

Offset: 0x0024			Register Name: CACHE_CFG_REG
Bit	Read/Write	Default/Hex	Description
11:8	R/W	0x0	DVS FARADAY SRAM delay signal
7	R/W	0x0	DVSE FARADAY SRAM delay enable signal
6	R/W	0x0	STOV
5:3	R/W	0x3	EMA Indicates the port number of Cache SRAM EMA control port.
2:1	R/W	0x1	EMAW Indicates the port number of Cache SRAM EMAW control port.
0	R/W	0x0	EMAS Indicates the port number of Cache SRAM EMAS control port.

### 3.2.4.6 0x0080 Cluster0 CPU Status Register (Default Value: 0x000E\_0000)

Offset: 0x0080			Register Name: C0_CPU_STATUS
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27:24	R	0x0	SMP_AMP Configure the operating mode for CPU[1:0] as Symmetric Multiprocessing mode or Asymmetric Multiprocessing mode. (The bit[27:26] is not used) 0: AMP mode 1: SMP mode
23:20	/	/	/
19:16	R	0xE	STANDBYWFI Indicates if CPU[1:0] of Cluster0 is in the WFI standby mode. (The bit[19:18] is not used) The meaning of each bit: 0: The processor is not in the WFI standby mode. 1: The processor is in the WFI standby mode.
15:12	/	/	/

Offset: 0x0080			Register Name: C0_CPU_STATUS
Bit	Read/Write	Default/Hex	Description
11:8	R	0x0	<p>STANDBYWFE Indicates if CPU[1:0] of Cluster0 is in the WFE standby mode. (The bit[11:10] is not used)</p> <p>The meaning of each bit: 0: The processor is not in the WFE standby mode. 1: The processor is in the WFE standby mode.</p>
7:1	/	/	/
0	R	0x0	<p>STANDBYWFL2 Indicates if the L2 memory system of Cluster0 is in the WFI standby mode. 0: Cluster0 L2 is not in the WFI standby mode. 1: Cluster0 L2 is in the WFI standby mode.</p>

### 3.2.4.7 0x0084 L2 Status Register (Default Value: 0x0000\_0000)

Offset: 0x0084			Register Name: L2_STATUS_REG
Bit	Read/Write	Default/Hex	Description
31:10	/	/	/
9	R	0x0	<p>EVENTO Event output. This bit is asserted high for 3 clock cycles when any core in the cluster executes an SEV instruction.</p>
8:0	/	/	/

### 3.2.4.8 0x00C0 Cluster 0 Debug Control Register0 (Default Value: 0x0000\_000F)

Offset: 0x00C0			Register Name: DBG_REG0
Bit	Read/Write	Default/Hex	Description
31:12	/	/	/
11:8	R/W	0x0	<p>DBGRESTART[1:0] Indicates if there are external restart requests for CPU[1:0]. (The bit[11:10] is not used)</p> <p>The meaning of each bit: 0: There are no such requests. 1: There are such requests.</p>

Offset: 0x00C0			Register Name: DBG_REG0
Bit	Read/Write	Default/Hex	Description
7:4	/	/	/
3:0	R/W	0xF	<p>C_DBGPWRDUP                      Indicates if CPU[1:0] is powered-up.                      (The bit[3:2] is not used)</p> <p>The meaning of each bit:                      0: The core is powered down.                      1: The core is powered up.</p>

### 3.2.4.9 0x00C4 Cluster 0 Debug Control Register1 (Default Value: 0x0000\_0000)

Offset: 0x00C4			Register Name: DBG_REG1
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:12	R/W	0x0	<p>DBGRESTARTED[1:0] (The bit[15:14] is not used)                      Handshake for DBGRESTART.</p>
11:8	/	/	/
7:4	R	0x0	<p>C_DBGNOPWRDWN                      Do Not Power Down Request.                      Indicates if the debugger requests CPU[1:0] not to power down.                      (The bit[7:6] is not used)</p> <p>The meaning of each bit:                      0: There are no such requests.                      1: There are such requests.</p>
3:0	R	0x0	<p>C_DBGPWRUPREQ                      Power Up Request.                      Indicates if the debugger requests CPU[1:0] to power up.                      (The bit[3:2] is not used)</p> <p>The meaning of each bit:                      0: There are no such requests.                      1: There are such requests.</p>

### 3.2.4.10 0x00D0 CPU AXI Monitor Control Register (Default Value: 0x0000\_0000)

Offset: 0x00D0			Register Name: AXI_MNT_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1	R/W	0x0	AXI_MONITOR function clear 0: Not to clear 1: Clear the AXI_MONITOR related registers with the offset address from 0x00D8 to 0x00EC.
0	R/W	0x0	Enable AXI_MONITOR function 0: Disabled 1: Enabled

### 3.2.4.11 0x00D4 CPU AXI Monitor Period Register (Default Value: 0x0000\_0000)

Offset: 0x00D4			Register Name: AXI_MNT_PRD_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	The monitoring period of AXI_MONITOR function (unit: cycles)

### 3.2.4.12 0x00D8 CPU AXI Monitor Read Latency Register (Default Value: 0x0000\_0000)

Offset: 0x00D8			Register Name: AXI_MNT_RLTCY_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	The total latency of read channel in the AXI_MONITOR module (unit: cycles)

### 3.2.4.13 0x00DC CPU AXI Monitor Write Latency Register (Default Value: 0x0000\_0000)

Offset: 0x00DC			Register Name: AXI_MNT_WLTCY_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	The total latency of write channel in the AXI_MONITOR module (unit: cycles)

### 3.2.4.14 0x00E0 CPU AXI Monitor Read Request Register (Default Value: 0x0000\_0000)

Offset: 0x00E0			Register Name: AXI_MNT_RREQ_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	The total times of read channel requests in the AXI_MONITOR module.

### 3.2.4.15 0x00E4 CPU AXI Monitor Write Request Register (Default Value: 0x0000\_0000)

Offset: 0x00E4			Register Name: AXI_MNT_WREQ_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	The total times of write channel requests in the AXI_MONITOR module.

### 3.2.4.16 0x00E8 CPU AXI Monitor Read Bandwidth Register (Default Value: 0x0000\_0000)

Offset: 0x00E8			Register Name: AXI_MNT_RBD_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	The bandwidth of read channel in the AXI_MONITOR module(Unit: KB)

### 3.2.4.17 0x00EC CPU AXI Monitor Write Bandwidth Register (Default Value: 0x0000\_0000)

Offset: 0x00EC			Register Name: AXI_MNT_WBD_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	The bandwidth of write channel in the AXI_MONITOR module (Unit: KB)

## 3.2.5 CPU Subsystem Control Register List

Module Name	Base Address
CPU_SUBSYS_CTRL	0x08100000

Register Name	Offset	Description
GENER_CTRL_REG0	0x0000	General Control Register0
GIC_JTAG_RST_CTRL	0x000C	GIC and JTAG Reset Control Register
C0_INT_EN	0x0010	Cluster0 Interrupt Enable Control Register
IRQ_FIQ_STATUS	0x0014	IRQ/FIQ Status Register
GENER_CTRL_REG2	0x0018	General Control Register2
DBG_STATE	0x001C	Debug State Register

### 3.2.6 CPU Subsystem Control Register Description

#### 3.2.6.1 0x0000 General Control Register0 (Default Value: 0x0000\_0000)

Offset: 0x0000			Register Name: GENER_CTRL_REG0
Bit	Read/Write	Default/Hex	Description
31:3	/	/	/
2	R/W	0x0	Cluster 0 Corepll Select 0: CCU clock 1: Disp PLL clock
1	R/W	0x0	IDC Clock Enable 0: IDC clock is disabled 1: IDC clock is enabled
0	R/W	0x0	GIC_CFGSDISABLE Disable write access to some secure GIC registers. 0: The write access to some secure GIC registers is enabled. 1: The write access to some secure GIC registers is disabled.

#### 3.2.6.2 0x000C GIC and JTAG Reset Control Register (Default Value: 0x0000\_0F07)

Offset: 0x000C			Register Name: GIC_JTAG_RST_CTRL
Bit	Read/Write	Default/Hex	Description
31:20	/	/	/

Offset: 0x000C			Register Name: GIC_JTAG_RST_CTRL
Bit	Read/Write	Default/Hex	Description
19:16	R/W	0x0	<p>EXM_CLR[1:0] (The bit[19:18] is not used)</p> <p>Clear the status of interface for debug.</p> <p>The meaning of each bit:</p> <p>0: Not clear</p> <p>1: Clear</p>
15:12	/	/	/
11	R/W	0x1	<p>CS_RST</p> <p>CoreSight reset</p> <p>0: Assert</p> <p>1: De-assert</p>
10	R/W	0x1	<p>DAP_RST</p> <p>DAP reset</p> <p>0: Assert</p> <p>1: De-assert</p>
9	R/W	0x1	<p>PORTRST</p> <p>DAP reset signal control</p> <p>0: Assert</p> <p>1: De-assert</p>
8	R/W	0x1	<p>TRST</p> <p>JTAG RTST signal</p> <p>0: Assert</p> <p>1: De-assert</p>
7:2	/	/	/
1	R/W	0x1	<p>IDC_RST</p> <p>Interrupt delay controller reset</p> <p>0: Assert</p> <p>1: De-assert</p>
0	R/W	0x1	<p>GIC_RST</p> <p>GIC_reset_cpu_reg</p> <p>0: Assert</p> <p>1: De-assert</p>

### 3.2.6.3 0x0010 Cluster 0 Interrupt Enable Register (Default Value: 0x0000\_FFFF)

Offset: 0x0010			Register Name: C0_INT_EN
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xFFFF	<p>C0_GIC_EN</p> <p>Mask the FIQ_OUT[15:0] and IRQ_OUT[15:0] signals to system domain.</p> <p>The meaning of each bit:</p> <p>0: Mask the FIQ_OUT and IRQ_OUT signals to system domain.</p> <p>1: Not to mask the FIQ_OUT and IRQ_OUT signals to system domain.</p>

### 3.2.6.4 0x0014 GIC IRQ/FIQ Status Register (Default Value: 0x0000\_0000)

Offset: 0x0014			Register Name: IRQ_FIQ_STATUS
Bit	Read/Write	Default/Hex	Description
31:16	R/W	0x0000	FIQ_OUT[15:0]
15:0	R/W	0x0000	IRQ_OUT[15:0]

### 3.2.6.5 0x0018 General Control Register2 (Default Value: 0x0000\_0000)

Offset: 0x0018			Register Name: GENER_CTRL_REG2
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	<p>CDBGRSTACK</p> <p>Debug Reset ACK</p> <p>0: There are no debug reset ACKs.</p> <p>1: There are debug reset ACKs.</p>
15:1	/	/	/
0	R/W	0x0	<p>C0_TSCLKCHANGE</p> <p>Cluster 0 time stamp change.</p> <p>0: There are no time stamp changes.</p> <p>1: There are time stamp changes.</p>

### 3.2.6.6 0x001C Debug State Register (Default Value: 0x0000\_0000)

Offset: 0x001C			Register Name: DBG_STATE
Bit	Read/Write	Default/Hex	Description
31:24	R	0x0	CLU_PWRSW_STA
23:1	/	/	/
0	R	0x0	<p>C0_DBG_STATE Indicates if Cluster0 is in debug mode or normal mode. 0: Normal mode 1: Debug mode</p>

## 3.3 Clock Controller Unit (CCU)

### 3.3.1 Overview

The clock controller unit (CCU) controls the PLL configurations and most of the clock generation, division, distribution, synchronization, and gating. The input signals of the CCU include the external clock for the reference frequency (24 MHz). The outputs from the CCU are mostly clocks to other blocks in the system.

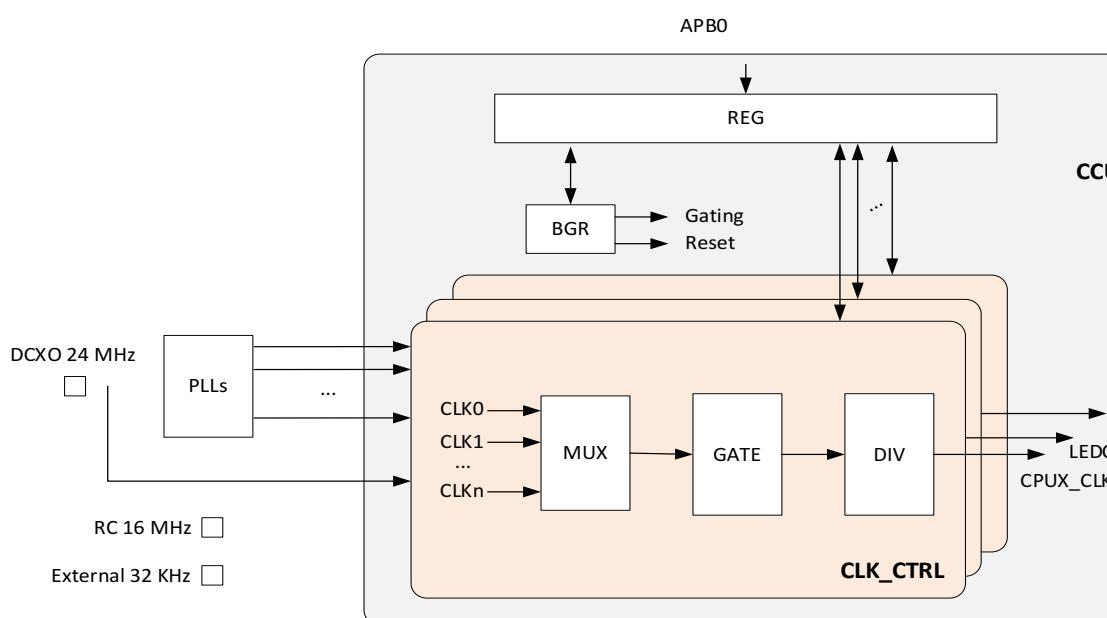
The CCU includes the following features:

- 8 PLLs
- Bus source and divisions
- Clock output control
- Configuring modules clock
- Bus clock gating
- Bus software reset

### 3.3.2 Block Diagram

The following figure shows the functional block diagram of the CCU.

Figure 3-2 CCU Block Diagram



### 3.3.3 Functional Description

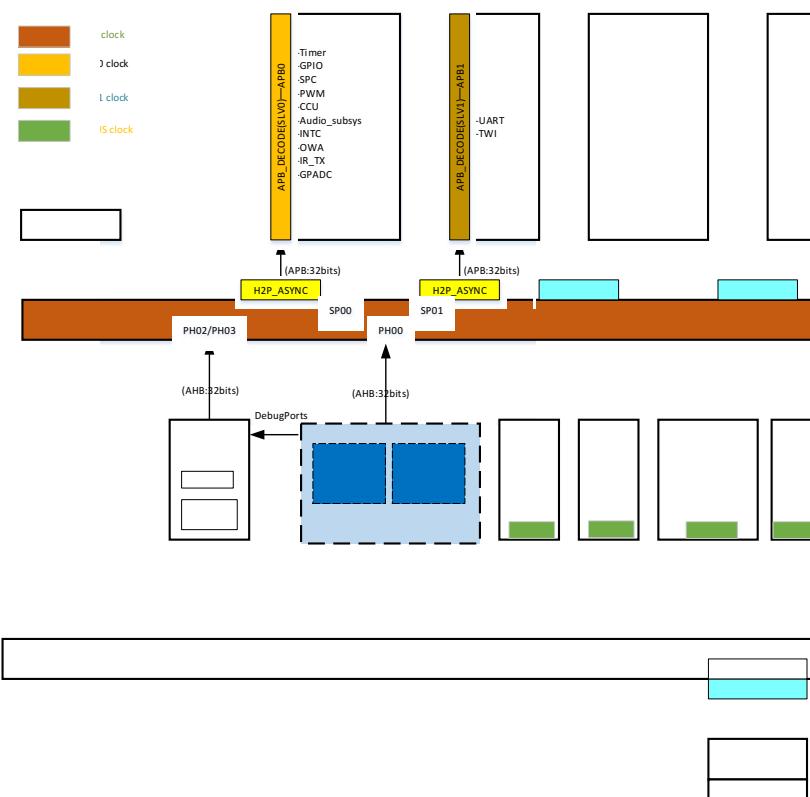
#### 3.3.3.1 System Bus Tree

The system buses include advanced high-performance buses (AHBs), advanced peripheral buses (APBs), and MBUS.

All devices mounted at the bus should use the related bus clocks, and the gating signals for the bus are from the CCU module.

The following figure shows the diagram of the System Bus Tree.

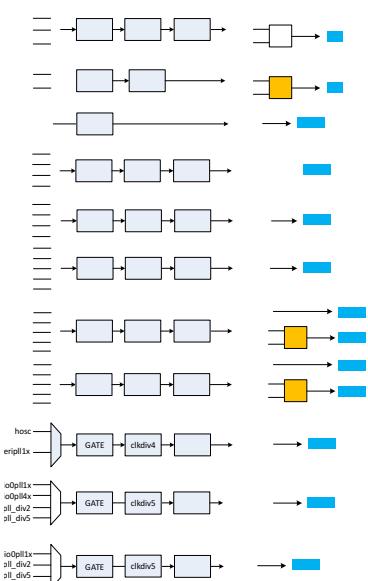
**Figure 3-3 System Bus Tree**



### 3.3.3.2 Module Clock Generation

The following figure describes module clock generation.

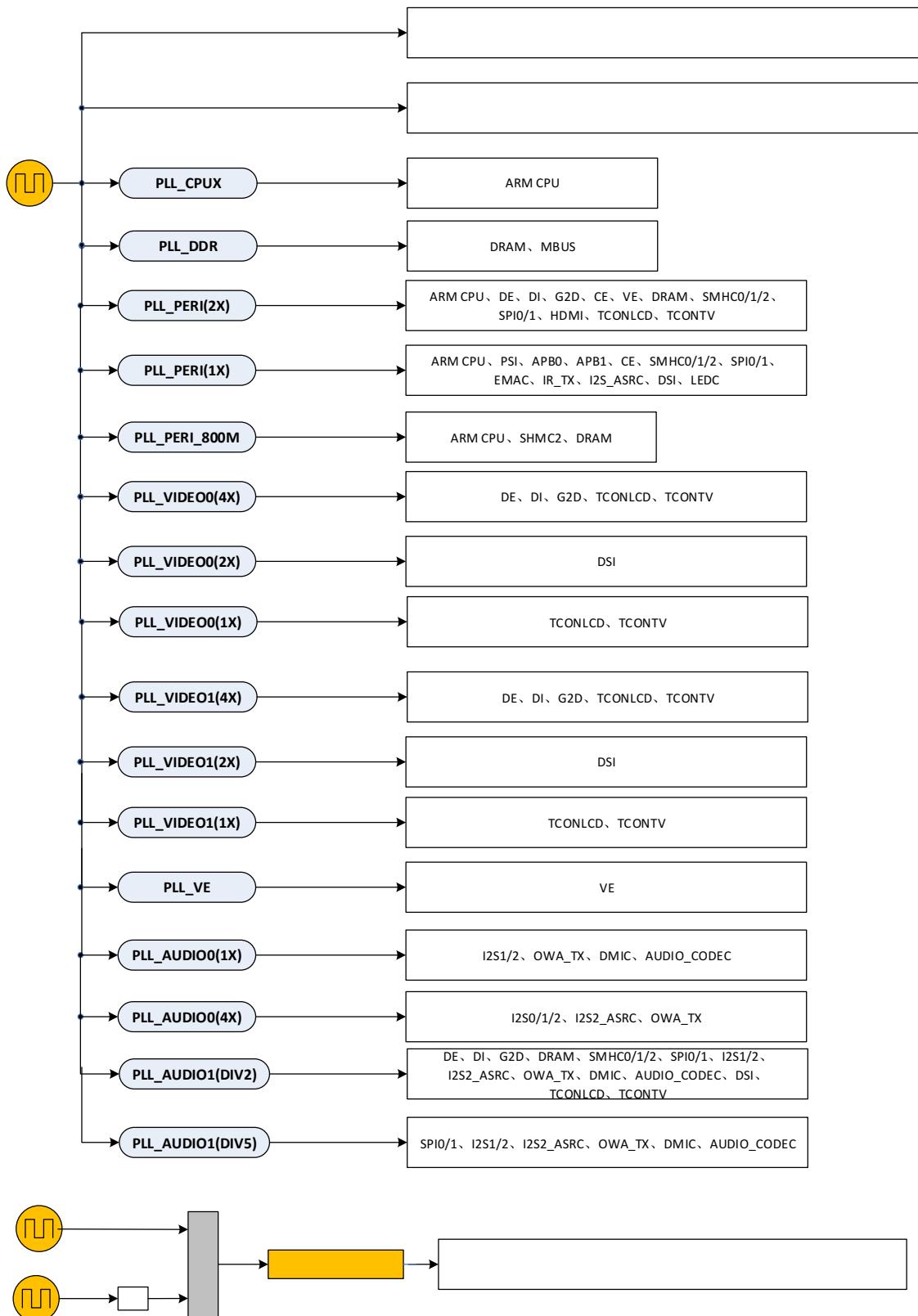
**Figure 3-4 Module Clock Generation**



### 3.3.3.3 PLL Distribution

The following figure shows the block diagram of the PLL distribution.

**Figure 3-5 PLL Distribution**



**Table 3-3 PLL Typical Application**

<b>PLL Type</b>	<b>Application Module</b>	<b>Notes</b>
PLL_CPU	ARM CPU	Support DVFS
PLL_DDR	MBUS, SDRAM	Support spread spectrum No support linear FM
PLL_PERI(2X)	ARM CPU, DE, DI, G2D, CE, VE, DRAM, SMHCO/1/2, SPI0/1, HDMI, TCONLCD, TCONTV	No support dynamic FM
PLL_PERI(1X)	ARM CPU, PSI, APBO, APB1, CE, SMHCO/1/2, SPI0/1, EMAC, IR_TX, I2S_ASRC, DSI, LEDC	No support dynamic FM
PLL_PERI(800M)	ARM CPU, SHMC2, DRAM	No support dynamic FM
PLL_VIDEO0(4X)	DE, DI, G2D, TCONLCD, TCONTV	No support DVFS
PLL_VIDEO0(2X)	DSI	No support DVFS
PLL_VIDEO0(1X)	TCONLCD, TCONTV	No support DVFS
PLL_VIDEO1(4X)	DE, DI, G2D, TCONLCD, TCONTV	No support DVFS
PLL_VIDEO1(2X)	DSI	No support DVFS
PLL_VIDEO1(1X)	TCONLCD, TCONTV	No support DVFS
PLL_VE	VE	No support dynamic FM
PLL_AUDIO0	I2S1/2, OWA_TX, DMIC, AUDIO_CODEC	No support DVFS
PLL_AUDIO0(4X)	I2S0/1/2, I2S2_ASRC, OWA_TX	No support DVFS
PLL_AUDIO1(DIV2)	DE, DI, G2D, DRAM, SMHCO/1/2, SPI0/1, I2S1/2, I2S2_ASRC, OWA_TX, DMIC, AUDIOCODEC, DSI, TCONLCD, TCONTV	No support DVFS
PLL_AUDIO1(DIV5)	SPI0/1, I2S1/2, I2S2_ASRC, OWA_TX, DMIC, AUDIO_CODEC	No support DVFS

### 3.3.3.4 PLL Features

The following table shows the PLL features.

**Table 3-4 PLL Features**

PLL	Stable Operating Frequency	Actual Operating Frequency	Default Frequency	Spread Spectrum	Linear FM	Rate Control	Pk-Pk	Lock Time
PLL_CPU	288 MHz to 3.0 GHz (24*N)	288 MHz to 1.8 GHz	408 MHz	No	Yes	No	< 200 ps	1.5 ms
PLL_AUDIO0	180 MHz to 3.0 GHz (24MHz*N.x/M1/M0/P)	24.576 MHz 22.5792 MHz 24.576*4 MHz 22.5792*4 MHz	24.576 MHz	Yes	No	No	< 200 ps	500 us
PLL_AUDIO1	180 MHz to 3.0 GHz	Fvco/Div (Div: 1~8)	Integer mode: 1/2x: 1.536 GHz 1/5x: 614.4 MHz  Decimal mode: 1/2x: 1.1179648 GHz 1/5x: 471.8592 MHz	Yes	No	No	< 200 ps	500 us
PLL_PERI	180 MHz to 3.2 GHz (24*N/M1/M0)	Fvco/Div (Div:1~8)	1/2x: 1.2 GHz 1/3x: 800 MHz	Yes	No	No	< 200ps	500 us
PLL_VIDEO0(4X)	252 MHz to 3.0 GHz (24*N/M)	192 MHz to 2400 MHz	1x: 297 MHz 2x: 594 MHz 4x: 1188 MHz	Yes	No	No	< 200ps	500 us
PLL_VIDEO1(4X)	252 MHz to 3.0 GHz (24*N/M)	192 MHz to 2400 MHz	1x: 297 MHz 2x: 594 MHz 4x: 1188 MHz	Yes	No	No	< 200ps	500 us
PLL_VE	180 MHz to 3.0 GHz (24*N/M1/M0)	192 MHz to 600 MHz	432 MHz	Yes	No	No	< 200ps	500 us
PLL_DDR	180 MHz to 3.0 GHz (24*N/M1/M0)	192 MHz to 2.0 GHz	432 MHz	Yes	No	No	200 MHz to 800 MHz (< 200 ps) 800 MHz to 1.3 GHz (< 140 ps) 1.3 GHz to 2.0 GHz (< 100 ps)	2 ms

### 3.3.3.5 The Usage Notes of PLL Frequencies

**Table 3-5 The Usage Notes of PLL Frequencies**

PLL Type	PLL Calculate Formula	PLL Output Frequency Range	Post Frequency-Divider of Module	Typical Working-Frequency of Module	Recommended PLL Parameter Configuration
PLL_CPU	24*N	200 MHz to 3 GHz	/P (1/2/4)	Based on the requirement of CPU	When the output clock is less than 288 MHz, the clock frequency can be outputted by dividing P.
PLL_PERI(1X)	24MHz*N/M/P0/2	600 MHz (P0=2)	1 to 64	600 MHz/300 MHz/ 200 MHz/150 MHz/ 100 MHz	N=100, M=1, P0=2, P1=3

PLL Type	PLL Calculate Formula	PLL Output Frequency Range	Post Frequency-Divider of Module	Typical Working-Frequency of Module	Recommended PLL Parameter Configuration
PLL_PERI(2X)	24MHz*N/M/P0	1.2 GHz (P0=2)	1 to 64	1.2 GHz/600 MHz/ 400 MHz/300 MHz/ 240 MHz/200 MHz/ 150 MHz/100 MHz	
PLL_PERI(800M)	24MHz*N/M/P1	800 MHz (P1=3)	1 to 64	800 MHz	
PLL_VIDEO0	24MHz*N/M	4x:0<=1188 MHz 1x: 0<=297 MHz	1 to 64	1188 MHz/585 MHz/ 537 MHz/ 501 MHz/ 468 MHz/462 MHz/ 426 MHz/390 MHz/ 357 MHz/297 MHz...	N=99, M=2
PLL_VIDEO1	24MHz*N/M	4x:0<=1188 MHz 1x: 0<=297 MHz	1 to 64	1188 MHz/585 MHz/ 537 MHz/ 501 MHz/ 468 MHz/462 MHz/ 426 MHz/390 MHz/ 357 MHz/297 MHz...	N=99, M=2,
PLL_VE	24MHz*N/M1/M0	200 MHz to 600 MHz	1 to 64	600 MHz/450 MHz/ 360 MHz/270 MHz	N=25, M1=1, M0=2
PLL_DDR	24MHz*N/M1/M0	192 MHz to 2.0 GHz	1 to 64	200 MHz/400 MHz/ 800 MHz	N=100, M1=M0=1, P=3
PLL_AUDIO0(1X) PLL_AUDIO0(4X) Decimal	24MHz*N.x/M1/M0/P/4 24MHz*N.x/M1/M0/P	24.576/22.5792 MHz (24.576/22.5792)*4 MHz	1 to 64	24.576/22.5792 MHz (24.576/22.5792)*4 MHz	24.571:N.x=14.336, P=14, M0=1, M1=1 22.5792:N.x=7.5264, P=8, M0=1, M1=1
PLL_AUDIO1(DIV2) Integral	24MHz*N/M/P0	1536 MHz (P0=2)	4 5 6 8 10 16 .....	384 MHz 307 MHz 256 MHz 192 MHz 154 MHz 96 MHz .....	N=128, M=1, P0=2, P1=5
PLL_AUDIO1(DIV5) Integral	24MHz*N/M/P1	614.4 MHz (P1=5)	25	24.576 MHz The jitter of this frequency point is better. The frequency point is used for audio module.	
PLL_AUDIO0(DIV2) Decimal	24MHz*N.x/M/P0	1179.648 MHz (P0=2)	2 3 4 5 6 8 12 48 .....	600 MHz 393 MHz 295 MHz 236 MHz 197 MHz 147 MHz 98 MHz 24.576 MHz The jitter of this frequency point is relatively poor. The frequency point is used for audio module. .....	N.x=98.304, M=1, P0=2, P1=1 to 8
PLL_AUDIO0(DIV5) Decimal	24MHz*N.x/M/P1	295 MHz to 2400 MHz	1 to 64	Select based on peripherals	

### 3.3.4 Programming Guidelines

#### 3.3.4.1 Configuring the Frequency of PLL\_CPU

The frequency configuration formula of PLL\_CPU:

$$\text{PLL\_CPU} = 24 \text{ MHz} * N / P$$

The parameter N is the frequency-doubling factor of PLL. The next parameter configuration can proceed after the PLL relock. The parameter P ([CPU\\_AXI\\_CFG\\_REG\[17:16\]](#)) is a digital post-frequency-division factor, which can be dynamically switched in real-time, without affecting the normal work of PLL.



PLL\_CPU supports dynamic frequency adjustment (modifying the value of N). However, for the system stability, to configure the frequency of the PLL\_CPU from a higher value to a lower one, switch the clock source of the CPU to another clock whose frequency is not higher than the current one first, and configure PLL\_CPU to the target low frequency, and then switch the clock source of the CPU back to PLL\_CPU.

Follow the steps below to adjust the frequency of PLL\_CPU:

- Step 1** Before you configure PLL\_CPU, switch the clock source of CPU to PLL\_PERI(1X).
- Step 2** Modify the parameters N and P of PLL\_CPU.
- Step 3** Write the PLL Lock Enable bit (bit[29]) of [PLL\\_CPU\\_CTRL\\_REG](#) to 0 and then to 1.
- Step 4** Wait for the Lock bit (bit[28]) of [PLL\\_CPU\\_CTRL\\_REG](#) to change to 1.
- Step 5** Switch the clock source of the CPU to PLL\_CPU.

#### 3.3.4.2 Configuring the Frequency of PLL\_AUDIO



PLL\_AUDIO includes PLL\_AUDIO0 and PLL\_AUDIO1.

The frequency configuration formula of PLL\_AUDIO:

$$\text{PLL\_AUDIO0} = 24 \text{ MHz} * N / M_0 / M_1 / P / 4$$

$$\text{PLL\_AUDIO1} = 24 \text{ MHz} * N / M$$

PLL\_AUDIO does not support dynamic adjustment because changing any parameter of N, M0, M1, and P will affect the normal work of PLL, and the PLL will need to be relocked.

Generally, PLL\_AUDIO only needs two frequency points: 24.576\*4 MHz or 22.5792\*4 MHz. For these two frequencies, there are usually special recommended matching factors. To implement the desired frequency point of PLL\_AUDIO, you need to use the decimal frequency-division function, so follow the steps below:

- Step 1** Configure the N, M0, M1 and P factors.
- Step 2** Configure [PLL\\_AUDIOx Control Register](#)[PLL\_SDM\_ENABLE] to 1.
- Step 3** Configure [PLL\\_AUDIOx Pattern0 Control Register](#) to enable the digital spread spectrum.
- Step 4** Write [PLL\\_AUDIOx Control Register](#)[Lock Enable] to 0 and then to 1.
- Step 5** Wait for [PLL\\_AUDIOx Control Register](#)[Lock] to 1.



When the P factor of PLL\_AUDIO is an odd number, the clock output is an unequal-duty-cycle signal.

The recommended values for configuration factors of PLL\_AUDIO1 are as follows.

**Table 3-6 Recommended Values for Configuration Factors of PLL\_AUDIO1**

Mode	Clock Source (MHz)	Frequency-doubling N	VCO (MHz)	Post Frequency-Division	PLL Output (MHz)	Divisor	Actual Operating Frequency (MHz)	Description
Integer divider	24	128	3072	2	1536	4	384	Provide clock source for peripherals
						8	192	
						16	96	
						.....		
				5	614.4	25	24.576	For audio-related modules
Decimal divider	24	98.304	2359.296	2	1179.648	2	589.824	Provides clock source for peripheral devices
						3	393.216	
						6	196.608	
						12	98.304	

Mode	Clock Source (MHz)	Frequency-doubling N	VCO (MHz)	Post Frequency-Division	PLL Output (MHz)	Divisor	Actual Operating Frequency (MHz)	Description
						48	24.576	For audio-related modules
				5	471.8592			

### 3.3.4.3 Configuring the Frequency of General PLLs

- Step 1** Make sure the PLL is enabled. If not, refer to section 3.3.4.4 [Enabling the PLL](#) to enable the PLL.
- Step 2** Configure the PLL\_OUTPUT\_ENABLE bit (bit[27]) of the PLL control register as 0 to disable the output gate of the PLL because general PLLs are unavailable in the process of frequency modulation.
- Step 3** Configure the N and M factors. (It is not suggested to configure M1 factor)
- Step 4** Write the LOCK\_ENABLE bit (bit[29]) of the PLL control register to 0 and then to 1.
- Step 5** Wait for the LOCK bit (bit[28]) of the PLL control register to 1.
- Step 6** Configure PLL\_OUTPUT\_ENABLE bit (bit[27]) of the PLL control register to 1.

### 3.3.4.4 Enabling the PLL

Follow the steps below to enable the PLL:

- Step 1** Configure the N, M, and P factors of the PLL control register.
- Step 2** Write the PLL\_ENABLE bit and the LDO\_EN bit of the PLL control register to 1, write the PLL\_OUTPUT\_GATE bit of the PLL control register to 0.
- Step 3** Write the LOCK\_ENABLE bit of the PLL control register to 1.
- Step 4** Wait for the status of the Lock to change to 1.
- Step 5** Delay 20 us.
- Step 6** Write the PLL\_OUTPUT\_GATE bit of the PLL control register to 1 and then the PLL will be available.

### 3.3.4.5 Disabling the PLL

Follow the steps below to disable the PLL:

**Step 1** Write the PLL\_ENABLE bit (bit[31]) and the LDO\_EN bit of the PLL control register to 0.

**Step 2** Write the LOCK\_ENABLE bit (bit[29]) of the PLL control register to 0.



#### CAUTION

In the normal use of PLLs, it is unsuggested to enable and disable the PLLs frequently. Turning on and off the PLLs will cause mutual interference between PLLs, which will affect the stability of the system. When the clock is unnecessary, you can write 0 to the PLL\_OUTPUT\_EN bit of the PLL control register to disable the output gate of the PLL, instead of writing 0 to the Enable bit to disable the PLL.

#### 3.3.4.6 Configuring Bus Clock

The bus clock supports dynamic switching, but the process of switching needs to follow the following two rules.

- From a higher frequency to a lower frequency: switch the clock source first, and then set the frequency division factor;
- From a lower frequency to a higher frequency: configure the frequency division factor first, and then switch the clock source.

#### 3.3.4.7 Configuring Module Clocks

For the Bus Gating Reset register of a module, the reset bit is de-asserted first, and then the clock gating bit is enabled to avoid potential problems caused by the asynchronous release of the reset signal.

For all module clocks except the DDR clock, configure the clock source and frequency division factor first, and then release the clock gating (that is, set to 1). For the configuration order of the clock source and frequency division factor, follow the rules below:

- With the increasing of the clock source frequency, configure the frequency division factor before the clock source;
- With the decreasing of the clock source frequency, configure the clock source before the frequency division factor.

#### 3.3.4.8 Implementing Spread Spectrum

The spread spectrum technology is to convert a narrowband signal into a wideband signal. It helps to reduce the effect of electromagnetic interference (EMI) associated with the fundamental frequency of the signal.

For the general PLL frequency, the calculation formula is as follows:

$$f = \frac{N+1+X}{P \cdot (M0+1) \cdot (M1+1)} \cdot 24MHz, 0 < X < 1$$

Where,

P is the frequency division factor of module or PLL;

M0 is the post-frequency division factor of PLL;

M1 is the pre-frequency division factor of PLL;

N is the frequency doubling factor of PLL;

X is the amplitude coefficient of the spread spectrum.

The parameters N, P, M1, and M0 are for the frequency division.

When M1 = 0, M0 = 0, and P = 1 (no frequency division), the calculation formula of PLL frequency can be simplified as follows:

$$f = (N + 1 + X) \cdot 24MHz, 0 < X < 1$$

$$[f_1, f_2] = (N + 1 + [X_1, X_2]) \cdot 24MHz$$

$$SDM\_BOT = 2^{17} * X_1$$

$$WAVE\_STEP = 2^{17} * (X_2 - X_1) / (24MHz/PREQ)*2$$

Where, SDM\_BOT and WAVE\_STEP are bits of the PLL pattern control register, and PREQ is the frequency of the spread spectrum.



#### NOTE

Different PLLs have different calculate formulas, refer to the CTRL register of the corresponding PLL in section 3.3.6 [Register Description](#).

## Configuration Procedure

Follow the steps below to implement the spread spectrum:

### Step 1 Configure the control register of the corresponding PLL

- a) Calculate the factor N and decimal value X according to the PLL frequency and PLL frequency formula. Refer to the control register of the corresponding PLL (named PLL\_xxx\_CTRL\_REG, where xxx is the module name) in 3.3.6 [Register Description](#) for the corresponding PLL frequency formula.
- b) Write M0, M1, N, and PLL frequency to the PLL control register.
- c) Configure the SDM\_Enable bit (bit[24]) of the PLL control register to 1 to enable the spread spectrum function.

**Step 2 Configure the pattern control register of the corresponding PLL**

- a) Calculate the SDM\_BOT and WAVE\_STEP of the pattern control register according to decimal value X and spread spectrum frequency (the bit[18:17] of the PLL\_PAT register)
- b) Configure the spread spectrum mode (SPR\_FREQ\_MODE) to 2 or 3.
- c) If the bit PLL\_INPUT\_DIV2 of the PLL control register is 1, configure the spread spectrum clock source select bit (SDM\_CLK\_SEL) of the PLL pattern control register to 1. Otherwise, configure SDM\_CLK\_SEL to the default value 0.
- d) Write SDM\_BOT, WAVE\_STEP, PREQ, SPR\_FREQ\_MODE, and SDM\_CLK\_SEL to the PLL pattern control register, and configure the SIG\_DELT\_PAT\_EN bit (bit[31]) of this register to 1.

**Step 3 Delay 20 us.****Configuration Example**

The following example shows how to configure the spread spectrum frequency as 605.3 MHz to 609.7 MHz.

If M1 = 0, M0 = 0, P = 1, according to the formula  $[f_1, f_2] = (N + 1 + [X_1, X_2]) \cdot 24\text{MHz}$ , you can get:

$$\begin{aligned} N + 1 + [X_1, X_2] &= \frac{[605.3, 609.7]}{24} \\ &= \frac{600 + [5.3, 9.7]}{24} \\ &= 24 + 1 + [5.3/24, 9.7/24] \end{aligned}$$

Obviously,

$$N = 24, X_1 = 5.3/24, X_2 = 9.7/24$$

$$\text{SDM\_BOT} = 2^{17} * X_1 = 0x7111$$

$$\text{WAVE\_STEP} = 2^{17} * (X_2 - X_1) / (24\text{M}/\text{PREQ}) * 2 = 0x3f; \text{PREQ} = 31.5 \text{ kHz}$$

If M0 = 1, M1=0, P = 1, then total frequency division factor is  $(M0 + 1) * 1 = 2$ , so the actual output frequency of PLL is 1212.1 MHz to 1219.4 MHz.

Similarly, you can get:

$$N = 49, X_1 = 12.1/24, X_2 = 19.4/24$$

Then calculate the values of SDM\_BOT and WAVE\_STEP according to the formulas, and follow the steps described in

**Configuration Procedure**

### 3.3.5 Register List

Module Name	Base Address
CCU	0x02001000

Register Name	Offset	Description
PLL_CPU_CTRL_REG	0x0000	PLL_CPU Control Register
PLL_DDR_CTRL_REG	0x0010	PLL_DDR Control Register
PLL_PERI_CTRL_REG	0x0020	PLL_PERI Control Register
PLL_VIDEO0_CTRL_REG	0x0040	PLL_VIDEO0 Control Register
PLL_VIDEO1_CTRL_REG	0x0048	PLL_VIDEO1 Control Register
PLL_VE_CTRL_REG	0x0058	PLL_VE Control Register
PLL_AUDIO0_CTRL_REG	0x0078	PLL_AUDIO0 Control Register
PLL_AUDIO1_CTRL_REG	0x0080	PLL_AUDIO1 Control Register
PLL_DDR_PAT0_CTRL_REG	0x0110	PLL_DDR Pattern0 Control Register
PLL_DDR_PAT1_CTRL_REG	0x0114	PLL_DDR Pattern1 Control Register
PLL_PERI_PAT0_CTRL_REG	0x0120	PLL_PERI Pattern0 Control Register
PLL_PERI_PAT1_CTRL_REG	0x0124	PLL_PERI Pattern1 Control Register
PLL_VIDEO0_PAT0_CTRL_REG	0x0140	PLL_VIDEO0 Pattern0 Control Register
PLL_VIDEO0_PAT1_CTRL_REG	0x0144	PLL_VIDEO0 Pattern1 Control Register
PLL_VIDEO1_PAT0_CTRL_REG	0x0148	PLL_VIDEO1 Pattern0 Control Register
PLL_VIDEO1_PAT1_CTRL_REG	0x014C	PLL_VIDEO1 Pattern1 Control Register
PLL_VE_PAT0_CTRL_REG	0x0158	PLL_VE Pattern0 Control Register
PLL_VE_PAT1_CTRL_REG	0x015C	PLL_VE Pattern1 Control Register
PLL_AUDIO0_PAT0_CTRL_REG	0x0178	PLL_AUDIO0 Pattern0 Control Register
PLL_AUDIO0_PAT1_CTRL_REG	0x017C	PLL_AUDIO0 Pattern1 Control Register
PLL_AUDIO1_PAT0_CTRL_REG	0x0180	PLL_AUDIO1 Pattern0 Control Register
PLL_AUDIO1_PAT1_CTRL_REG	0x0184	PLL_AUDIO1 Pattern1 Control Register
PLL_CPU_BIAS_REG	0x0300	PLL_CPU Bias Register
PLL_DDR_BIAS_REG	0x0310	PLL_DDR Bias Register
PLL_PERI_BIAS_REG	0x0320	PLL_PERI Bias Register
PLL_VIDEO0_BIAS_REG	0x0340	PLL_VIDEO0 Bias Register
PLL_VIDEO1_BIAS_REG	0x0348	PLL_VIDEO1 Bias Register
PLL_VE_BIAS_REG	0x0358	PLL_VE Bias Register
PLL_AUDIO0_BIAS_REG	0x0378	PLL_AUDIO0 Bias Register

Register Name	Offset	Description
PLL_AUDIO1_BIAS_REG	0x0380	PLL_AUDIO1 Bias Register
PLL_CPU_TUN_REG	0x0400	PLL_CPU Tuning Register
CPU_AXI_CFG_REG	0x0500	CPU_AXI Configuration Register
CPU_GATING_REG	0x0504	CPU_GATING Configuration Register
PSI_CLK_REG	0x0510	PSI Clock Register
APB0_CLK_REG	0x0520	APB0 Clock Register
APB1_CLK_REG	0x0524	APB1 Clock Register
MBUS_CLK_REG	0x0540	MBUS Clock Register
DE_CLK_REG	0x0600	DE Clock Register
DE_BGR_REG	0x060C	DE Bus Gating Reset Register
DI_CLK_REG	0x0620	DI Clock Register
DI_BGR_REG	0x062C	DI Bus Gating Reset Register
G2D_CLK_REG	0x0630	G2D Clock Register
G2D_BGR_REG	0x063C	G2D Bus Gating Reset Register
CE_CLK_REG	0x0680	CE Clock Register
CE_BGR_REG	0x068C	CE Bus Gating Reset Register
VE_CLK_REG	0x0690	VE Clock Register
VE_BGR_REG	0x069C	VE Bus Gating Reset Register
DMA_BGR_REG	0x070C	DMA Bus Gating Reset Register
HSTIMER_BGR_REG	0x073C	HSTIMER Bus Gating Reset Register
AVS_CLK_REG	0x0740	AVS Clock Register
DBGSYS_BGR_REG	0x078C	DBGSYS Bus Gating Reset Register
PWM_BGR_REG	0x07AC	PWM Bus Gating Reset Register
IOMMU_BGR_REG	0x07BC	IOMMU Bus Gating Reset Register
DRAM_CLK_REG	0x0800	DRAM Clock Register
MBUS_MAT_CLK_GATING_REG	0x0804	MBUS Master Clock Gating Register
DRAM_BGR_REG	0x080C	DRAM Bus Gating Reset Register
SMHCO_CLK_REG	0x0830	SMHCO Clock Register
SMHC1_CLK_REG	0x0834	SMHC1 Clock Register
SMHC2_CLK_REG	0x0838	SMHC2 Clock Register
SMHC_BGR_REG	0x084C	SMHC Bus Gating Reset Register
UART_BGR_REG	0x090C	UART Bus Gating Reset Register
TWI_BGR_REG	0x091C	TWI Bus Gating Reset Register
SPI0_CLK_REG	0x0940	SPI0 Clock Register

Register Name	Offset	Description
SPI1_CLK_REG	0x0944	SPI1 Clock Register
SPI_BGR_REG	0x096C	SPI Bus Gating Reset Register
EMAC_25M_CLK_REG	0x0970	EMAC_25M Clock Register
EMAC_BGR_REG	0x097C	EMAC Bus Gating Reset Register
IRTX_CLK_REG	0x09C0	IRTX Clock Register
IRTX_BGR_REG	0x09CC	IRTX Bus Gating Reset Register
GPADC_BGR_REG	0x09EC	GPADC Bus Gating Reset Register
THS_BGR_REG	0x09FC	THS Bus Gating Reset Register
I2S1_CLK_REG	0x0A14	I2S1 Clock Register
I2S2_CLK_REG	0x0A18	I2S2 Clock Register
I2S2_ASRC_CLK_REG	0x0A1C	I2S2_ASRC Clock Register
I2S_BGR_REG	0x0A20	I2S Bus Gating Reset Register
OWA_TX_CLK_REG	0x0A24	OWA_TX Clock Register
OWA_BGR_REG	0x0A2C	OWA Bus Gating Reset Register
DMIC_CLK_REG	0x0A40	DMIC Clock Register
DMIC_BGR_REG	0x0A4C	DMIC Bus Gating Reset Register
AUDIO_CODEC_DAC_CLK_REG	0x0A50	AUDIO_CODEC_DAC Clock Register
AUDIO_CODEC_ADC_CLK_REG	0x0A54	AUDIO_CODEC_ADC Clock Register
AUDIO_CODEC_BGR_REG	0x0A5C	AUDIO_CODEC Bus Gating Reset Register
USB0_CLK_REG	0x0A70	USB0 Clock Register
USB1_CLK_REG	0x0A74	USB1 Clock Register
USB_BGR_REG	0x0A8C	USB Bus Gating Reset Register
DPSS_TOP_BGR_REG	0x0ABC	DPSS_TOP Bus Gating Reset Register
HDMI_24M_CLK_REG	0x0B04	HDMI_24M Clock Register
HDMI_CEC_CLK_REG	0x0B10	HDMI CEC Clock Register
HDMI_BGR_REG	0x0B1C	HDMI Bus Gating Reset Register
DSI_CLK_REG	0x0B24	DSI Clock Register
DSI_BGR_REG	0x0B4C	DSI Bus Gating Reset Register
TCONLCD_CLK_REG	0x0B60	TCONLCD Clock Register
TCONLCD_BGR_REG	0x0B7C	TCONLCD Bus Gating Reset Register
TCONTV_CLK_REG	0x0B80	TCONTV Clock Register
TCONTV_BGR_REG	0x0B9C	TCONTV Bus Gating Reset Register
LVDS_BGR_REG	0x0BAC	LVDS Bus Gating Reset Register
LEDC_CLK_REG	0x0BF0	LEDC Clock Register

Register Name	Offset	Description
LEDC_BGR_REG	0x0BFC	LEDC Bus Gating Reset Register
TZMA_BGR_REG	0x0C6C	TZMA Bus Gating Reset Register
CCU_SEC_SWITCH_REG	0x0F00	CCU Security Switch Register
PLL_LOCK_DBG_CTRL_REG	0x0F04	PLL Lock Debug Control Register
FRE_DET_CTRL_REG	0x0F08	Frequency Detect Control Register
FRE_UP_LIM_REG	0x0F0C	Frequency Up Limit Register
FRE_DOWN_LIM_REG	0x0F10	Frequency Down Limit Register
CCU_FAN_GATE_REG	0x0F30	CCU FANOUT CLOCK GATE Register
CLK27M_FAN_REG	0x0F34	CLK27M FANOUT Register
PCLK_FAN_REG	0x0F38	PCLK FANOUT Register
CCU_FAN_REG	0x0F3C	CCU FANOUT Register

### 3.3.6 Register Description



#### NOTE

In secure mode, these registers from 0x0000 offset to 0x0540 offset are secure registers and can only be accessed by the secure host.

For 0x0000 offset to 0x0400 offset, the secure control bit is in the bit0 of [CCU SEC SWITCH REG](#).

For 0x0500 offset to 0x0524 offset, the secure control bit is in the bit1 of [CCU SEC SWITCH REG](#).

For 0x0540, 0x0804 offset, the secure control bit is in the bit2 of [CCU SEC SWITCH REG](#)

### 3.3.6.1 0x0000 PLL\_CPU Control Register (Default Value: 0x4A00\_1000)

Offset: 0x0000			Register Name: PLL_CPU_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<p>PLL_EN PLL Enable 0: Disable 1: Enable The PLL_CPU= InputFreq*N. <b>The PLL_CPU output frequency must be in the range from 200 MHz to 3 GHz. And the default value of PLL_CPU is 408 MHz when the crystal oscillator is 24 MHz.</b></p>
30	R/W	0x1	<p>PLL_LDO_EN LDO Enable 0: Disable 1: Enable</p>
29	R/W	0x0	<p>LOCK_ENABLE Lock Enable 0: Disable 1: Enable</p>
28	R	0x0	<p>LOCK PLL Lock Status 0: Unlocked 1: Locked (It indicates that the PLL has been stable.)</p>
27	R/W	0x1	<p>PLL_OUTPUT_GATE PLL Output Gating Enable 0: Disable 1: Enable The bit is used to control the output enable of the PLL.</p>
26:24	R/W	0x2	<p>PLL_LOCK_TIME. PLL Lock Time The bit indicates the step amplitude from one frequency to another.</p>
23:16	/	/	/
15:8	R/W	0x10	<p>PLL_N PLL N N= PLL_N +1 PLL_N is from 0 to 254. In application, PLL_N shall be more than or equal to 12.</p>

Offset: 0x0000			Register Name: PLL_CPU_CTRL_REG
Bit	Read/Write	Default/Hex	Description
7:6	R/W	0x0	PLL_UNLOCK_MDSEL PLL Unlock Level 00: 21-29 Clock Cycles 01: 22-28 Clock Cycles 1X: 20-30 Clock Cycles
5	R/W	0x0	PLL_LOCK_MDSEL PLL Lock Level 0: 24-26 Clock Cycles 1: 23-27 Clock Cycles
4:2	/	/	/
1:0	R/W	0x0	PLL_M PLL_M M = PLL_FACTOR_M + 1 PLL_FACTOR_M is from 0 to 3. <b>Note: The M factor is only for testing.</b>

### 3.3.6.2 0x0010 PLL\_DDR Control Register (Default Value: 0x4800\_2301)

Offset: 0x0010			Register Name: PLL_DDR_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	PLL_EN 0: Disable 1: Enable PLL_DDR = InputFreq*N/M1/M0. <b>The default value of PLL_DDR is 432 MHz when the crystal oscillator is 24 MHz.</b>
30	R/W	0x1	PLL_LDO_EN LDO ENABLE 0: Disable 1: Enable
29	R/W	0x0	LOCK_ENABLE Lock Enable 0: Disable 1: Enable

Offset: 0x0010			Register Name: PLL_DDR_CTRL_REG
Bit	Read/Write	Default/Hex	Description
28	R	0x0	<p>LOCK</p> <p>PLL Lock</p> <p>0: Unlocked</p> <p>1: Locked (It indicates that the PLL has been stable.)</p>
27	R/W	0x1	<p>PLL_OUTPUT_GATE</p> <p>PLL Output Gating Enable</p> <p>0: Disable</p> <p>1: Enable</p> <p>The bit is used to control the output enable of the PLL.</p>
26:25	/	/	/
24	R/W	0x0	<p>PLL_SDM_EN</p> <p>PLL SDM Enable</p> <p>0: Disable</p> <p>1: Enable</p>
23:16	/	/	/
15:8	R/W	0x23	<p>PLL_N</p> <p>PLL N</p> <p>N= PLL_N +1</p> <p>PLL_N is from 0 to 254.</p> <p>In application, PLL_N shall be more than or equal to 12.</p>
7:6	R/W	0x0	<p>PLL_UNLOCK_MDSEL</p> <p>PLL Unlock Level</p> <p>00: 21-29 Clock Cycles</p> <p>01: 22-28 Clock Cycles</p> <p>1X: 20-30 Clock Cycles</p>
5	R/W	0x0	<p>PLL_LOCK_MDSEL</p> <p>PLL Lock Level</p> <p>0: 24-26 Clock Cycles</p> <p>1: 23-27 Clock Cycles</p>
4:2	/	/	/
1	R/W	0x0	<p>PLL_INPUT_DIV2</p> <p>PLL Input Divider M1</p> <p>M1=PLL_INPUT_DIV2 + 1</p> <p>PLL_INPUT_DIV2 is from 0 to 1.</p>

Offset: 0x0010			Register Name: PLL_DDR_CTRL_REG
Bit	Read/Write	Default/Hex	Description
0	R/W	0x1	PLL_OUTPUT_DIV2 PLL Output Div M0 $M0 = PLL\_OUTPUT\_DIV2 + 1$ PLL_OUTPUT_DIV2 is from 0 to 1.

### 3.3.6.3 0x0020 PLL\_PERI Control Register (Default Value: 0x4821\_6300)

Offset: 0x0020			Register Name: PLL_PERI_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	PLL_EN PLL Enable 0: Disable 1: Enable $PLL\_PERI(2X) = 24 \text{ MHz} * N / M / P0$ $PLL\_PERI(1X) = 24 \text{ MHz} * N / M / P0 / 2$ $PLL\_PERI(800M) = 24 \text{ MHz} * N / M / P1.$ When the crystal oscillator is 24 MHz, the default frequency of PLL_PERI(2X) is 1.2 GHz, the default frequency of PLL_PERI(1X) is 600 MHz, and the default frequency of PLL_PERI(800M) is 800 MHz. <b>The output clock of PLL_PERI(2X) is fixed to 1.2 GHz and not suggested to change the parameter.</b>
30	R/W	0x1	PLL_LDO_EN LDO Enable 0: Disable 1: Enable
29	R/W	0x0	LOCK_ENABLE Lock Enable 0: Disable 1: Enable
28	R	0x0	LOCK PLL Lock 0: Unlocked 1: Locked (It indicates that the PLL has been stable.)

Offset: 0x0020			Register Name: PLL_PERI_CTRL_REG
Bit	Read/Write	Default/Hex	Description
27	R/W	0x1	<p>PLL_OUTPUT_GATE PLL Output Gating Enable 0: Disable 1: Enable The bit is used to control the output enable of PLL.</p>
26:25	/	/	/
24	R/W	0x0	<p>PLL_SDM_EN PLL SDM Enable 0: Disable 1: Enable Enable spread spectrum and decimal division.</p>
23	/	/	/
22:20	R/W	0x2	<p>PLL_P1 PLL Output Divider P1 P1=PLL_OUTPUT_DIV_P1 + 1 PLL_OUTPUT_DIV_P1 is from 0 to 7.</p>
19	/	/	/
18:16	R/W	0x1	<p>PLL_P0 PLL Output Divider P0 P0=PLL_OUTPUT_DIV_P0 + 1 PLL_OUTPUT_DIV_P0 is from 0 to 7.</p>
15:8	R/W	0x63	<p>PLL_N PLL N N= PLL_N +1 PLL_N is from 0 to 254. In application, PLL_N shall be more than or equal to 12.</p>
7:6	R/W	0x0	<p>PLL_UNLOCK_MDSEL PLL Unlock Level 00: 21-29 Clock Cycles 01: 22-28 Clock Cycles 1X: 20-30 Clock Cycles</p>
5	R/W	0x0	<p>PLL_LOCK_MDSEL PLL Lock Level 0: 24-26 Clock Cycles 1: 23-27 Clock Cycles</p>
4:2	/	/	/

Offset: 0x0020			Register Name: PLL_PERI_CTRL_REG
Bit	Read/Write	Default/Hex	Description
1	R/W	0x0	<p>PLL_INPUT_DIV2 PLL Input Divider M <math>M1 = \text{PLL\_INPUT\_DIV2} + 1</math> PLL_INPUT_DIV2 is from 0 to 1.</p>
0	/	/	/

### 3.3.6.4 0x0040 PLL\_VIDEO0 Control Register (Default Value: 0x4800\_6203)

Offset: 0x0040			Register Name: PLL_VIDEO0_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<p>PLL_EN PLL Enable 0: Disable 1: Enable For application, <math>\text{PLL\_VIDEO0}(4X) = \text{InputFreq} * N/M</math>. <math>\text{PLL\_VIDEO0}(2X) = \text{InputFreq} * N/M/2</math>. <math>\text{PLL\_VIDEO0}(1X) = \text{InputFreq} * N/M/4</math>. When the HOSC is 24 MHz, the default frequency of PLL_VIDEO0(4X) is 1188 MHz.</p>
30	R/W	0x1	<p>PLL_LDO_EN LDO Enable 0: Disable 1: Enable</p>
29	R/W	0x0	<p>LOCK_ENABLE Lock Enable 0: Disable 1: Enable</p>
28	R	0x0	<p>LOCK PLL Lock 0: Unlocked 1: Locked (It indicates that the PLL has been stable.)</p>

Offset: 0x0040			Register Name: PLL_VIDEO0_CTRL_REG
Bit	Read/Write	Default/Hex	Description
27	R/W	0x1	<p>PLL_OUTPUT_GATE PLL Output Gating Enable 0: Disable 1: Enable The bit is used to control the output enable of PLL.</p>
26:25	/	/	/
24	R/W	0x0	<p>PLL_SDM_EN PLL SDM Enable 0: Disable 1: Enable</p>
23:16	/	/	/
15:8	R/W	0x62	<p>PLL_N PLL N N= PLL_N +1 PLL_N is from 0 to 254. In application, PLL_N shall be more than or equal to 12.</p>
7:6	R/W	0x0	<p>PLL_UNLOCK_MDSEL PLL Unlock Level 00: 21-29 Clock Cycles 01: 22-28 Clock Cycles 1X: 20-30 Clock Cycles</p>
5	R/W	0x0	<p>PLL_LOCK_MDSEL PLL Lock Level 0: 24-26 Clock Cycles 1: 23-27 Clock Cycles</p>
4:2	/	/	/
1	R/W	0x1	<p>PLL_INPUT_DIV2 PLL Input Divider M M1=PLL_INPUT_DIV_M + 1 PLL_INPUT_DIV_M is from 0 to 1.</p>
0	R/W	0x1	<p>PLL_OUTPUT_DIV2 PLL Output Divider D (The factor is only for testing) M0=PLL_OUTPUT_DIV_D + 1 PLL_OUTPUT_DIV_D is from 0 to 1. For test, PLL_VIDEO0(4X) =24MHz*N/M/D</p>

## 3.3.6.5 0x0048 PLL\_VIDEO1 Control Register (Default Value: 0x4800\_6203)

Offset: 0x0048			Register Name: PLL_VIDEO1_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<p>PLL_EN PLL Enable 0: Disable 1: Enable For application, <math>\text{PLL\_VIDEO1}(4X) = \text{InputFreq} * N/M.</math> <math>\text{PLL\_VIDEO1}(2X) = \text{InputFreq} * N/M/2.</math> <math>\text{PLL\_VIDEO1}(1X) = \text{InputFreq} * N/M/4.</math> When the HOSC is 24 MHz, the default frequency of PLL_VIDEO1(4X) is 1188 MHz.</p>
30	R/W	0x1	<p>PLL_LDO_EN LDO Enable 0: Disable 1: Enable</p>
29	R/W	0x0	<p>LOCK_ENABLE Lock Enable 0: Disable 1: Enable</p>
28	R	0x0	<p>LOCK PLL Lock 0: Unlocked 1: Locked (It indicates that the PLL has been stable.)</p>
27	R/W	0x1	<p>PLL_OUTPUT_GATE PLL Output Gating Enable 0: Disable 1: Enable The bit is used to control the output enable of PLL.</p>
26:25	/	/	/
24	R/W	0x0	<p>PLL_SDM_EN PLL SDM Enable 0: Disable 1: Enable</p>
23:16	/	/	/

Offset: 0x0048			Register Name: PLL_VIDEO1_CTRL_REG
Bit	Read/Write	Default/Hex	Description
15:8	R/W	0x62	PLL_FACTOR_N PLL FACTOR N N= PLL_FACTOR_N +1 PLL_FACTOR_N is from 0 to 254. In application, PLL_FACTOR_N shall be more than or equal to 12.
7:6	R/W	0x0	PLL_UNLOCK_MDSEL PLL Unlock Level 00: 21-29 Clock Cycles 01: 22-28 Clock Cycles 1X: 20-30 Clock Cycles
5	R/W	0x0	PLL_LOCK_MDSEL PLL Lock Level 0: 24-26 Clock Cycles 1: 23-27 Clock Cycles
4:2	/	/	/
1	R/W	0x1	PLL_INPUT_DIV2 PLL Input Divider M M1=PLL_INPUT_DIV_M + 1 PLL_INPUT_DIV_M is from 0 to 1.
0	R/W	0x1	PLL_OUTPUT_DIV2 PLL Output Divider D(The factor is only for testing) M0=PLL_OUTPUT_DIV_D + 1 PLL_OUTPUT_DIV_D is from 0 to 1. For test, PLL_VIDEO1(4X) =24MHz*N/M/D

**3.3.6.6 0x0058 PLL\_VE Control Register (Default Value: 0x4800\_2301)**

Offset: 0x0058			Register Name: PLL_VE_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	PLL_EN PLL Enable 0: Disable 1: Enable PLL_VE = InputFreq*N/M1/M0. When the HOSC is 24 MHz, the default frequency of PLL_VE is 432 MHz.
30	R/W	0x1	PLL_LDO_EN LDO Enable 0: Disable 1: Enable
29	R/W	0x0	LOCK_ENABLE Lock Enable 0: Disable 1: Enable
28	R	0x0	LOCK PLL Lock 0: Unlocked 1: Locked (It indicates that the PLL has been stable.)
27	R/W	0x1	PLL_OUTPUT_GATE PLL Output Gating Enable 0: Disable 1: Enable The bit is used to control the output enable of PLL.
26:25	/	/	/
24	R/W	0x0	PLL_SDM_EN PLL SDM Enable 0: Disable 1: Enable
23:16	/	/	/

Offset: 0x0058			Register Name: PLL_VE_CTRL_REG
Bit	Read/Write	Default/Hex	Description
15:8	R/W	0x23	PLL_N PLL N N= PLL_N +1 PLL_N is from 0 to 254. In application, PLL_N shall be more than or equal to 12.
7:6	R/W	0x0	PLL_UNLOCK_MDSEL PLL Unlock Level 00: 21-29 Clock Cycles 01: 22-28 Clock Cycles 1X: 20-30 Clock Cycles
5	R/W	0x0	PLL_LOCK_MDSEL PLL Lock Level 0: 24-26 Clock Cycles 1: 23-27 Clock Cycles
4:2	/	/	/
1	R/W	0x0	PLL_INPUT_DIV2 PLL Input Divider M1 M1=PLL_INPUT_DIV2 + 1 PLL_INPUT_DIV2 is from 0 to 1.
0	R/W	0x1	PLL_OUTPUT_DIV2 PLL Output Divider M0 M0=PLL_OUTPUT_DIV2 + 1 PLL_OUTPUT_DIV2 is from 0 to 1.

### 3.3.6.7 0x0078 PLL\_AUDIO0 Control Register (Default Value: 0x4814\_5500)

Offset: 0x0078			Register Name: PLL_AUDIO0_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<p>PLL_EN PLL Enable 0: Disable 1: Enable</p> <p><math>\text{PLL\_AUDIO0(4X)} = 24\text{MHz} * \text{N} / \text{M1/M0/P}</math>  <math>\text{PLL\_AUDIO0(2X)} = (24\text{MHz} * \text{N} / \text{M1/M0}) / \text{P}/2</math>  <math>\text{PLL\_AUDIO0(1X)} = (24\text{MHz} * \text{N} / \text{M1/M0}) / \text{P}/4</math></p> <p><b>Note: <math>7.5 \leq \text{N/M0/M1} \leq 125</math> and <math>12 \leq \text{N}</math></b></p> <p>The working frequency range of <math>24\text{MHz} * \text{N} / \text{M0/M1}</math> is from 180 MHz to 3.0 GHz.</p> <p>The default frequency of <math>\text{PLL\_AUDIO0(1X)}</math> is 24.5714 MHz.</p>
30	R/W	0x1	<p>PLL_LDO_EN LDO Enable 0: Disable 1: Enable</p>
29	R/W	0x0	<p>LOCK_ENABLE Lock Enable 0: Disable 1: Enable</p>
28	R	0x0	<p>LOCK PLL Lock 0: Unlocked 1: Locked (It indicates that the PLL has been stable.)</p> <p><b>Note: The bit is only valid when the bit29 is set to 1.</b></p>
27	R/W	0x1	<p>PLL_OUTPUT_GATE 0: Disable 1: Enable</p> <p>The bit is used to control the output enable of PLL.</p>
26:25	/	/	/
24	R/W	0x0	<p>PLL_SDM_EN PLL SDM Enable 0: Disable 1: Enable</p> <p>Enable spread spectrum and decimal division.</p>
23:22	/	/	/

Offset: 0x0078			Register Name: PLL_AUDIO0_CTRL_REG
Bit	Read/Write	Default/Hex	Description
21:16	R/W	0x14	PLL_P PLL Post-divider P $P = \text{PLL\_POST\_DIV\_P} + 1$ PLL_POST_DIV_P is from 0 to 63.
15:8	R/W	0x55	PLL_N PLL N $N = \text{PLL\_N} + 1$ PLL_N is from 0 to 254. In application, PLL_N shall be more than or equal to 12.
7:6	R/W	0x0	PLL_UNLOCK_MDSEL PLL Unlock Level 00: 21-29 Clock Cycles 01: 22-28 Clock Cycles 1X: 20-30 Clock Cycles
5	R/W	0x0	PLL_LOCK_MDSEL PLL Lock Level 0: 24-26 Clock Cycles 1: 23-27 Clock Cycles
4:2	/	/	/
1	R/W	0x0	PLL_INPUT_DIV2 PLL Input Divider M1 $M1 = \text{PLL\_INPUT\_DIV2} + 1$ PLL_INPUT_DIV2 is from 0 to 1.
0	R/W	0x0	PLL_OUTPUT_DIV2 PLL Output Divider M0 $M0 = \text{PLL\_OUTPUT\_DIV2} + 1$ PLL_OUTPUT_DIV2 is from 0 to 1.

## 3.3.6.8 0x0080 PLL\_AUDIO1 Control Register (Default Value: 0x4841\_7F00)

Offset: 0x0080			Register Name: PLL_AUDIO1_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<p>PLL_EN PLL Enable 0: Disable 1: Enable</p> <p>PLL_AUDIO1 = 24MHz*N/M PLL_AUDIO1(DIV2) = 24MHz*N/M/P0 PLL_AUDIO1(DIV5) = 24MHz*N/M/P1</p> <p>The working frequency range of 24 MHz/M*N is from 180 MHz to 3.5 GHz.</p> <p>The default frequency of PLL_AUDIO1 is 3072 MHz.</p> <p>The default frequency of PLL_AUDIO1(DIV2) is 1536 MHz.</p> <p>The default frequency of PLL_AUDIO1(DIV5) is 614.4 MHz (24.576 MHz*25).</p>
30	R/W	0x1	<p>PLL_LDO_EN LDO Enable 0: Disable 1: Enable</p>
29	R/W	0x0	<p>LOCK_ENABLE Lock Enable 0: Disable 1: Enable</p>
28	R	0x0	<p>LOCK 0: Unlocked 1: Locked (It indicates that the PLL has been stable.)</p> <p><b>Note:</b> The bit is only valid when the bit29 is set to 1.</p>
27	R/W	0x1	<p>PLL_OUTPUT_GATE 0: Disable 1: Enable</p> <p>The bit is used to control the output enable of PLL.</p>
26:25	/	/	/
24	R/W	0x0	<p>PLL_SDM_EN 0: Disable 1: Enable</p> <p>Enable spread spectrum and decimal division.</p>
23	/	/	/

Offset: 0x0080			Register Name: PLL_AUDIO1_CTRL_REG
Bit	Read/Write	Default/Hex	Description
22:20	R/W	0x4	<p>PLL_P1</p> <p>PLL Output Divider P1</p> <p>P1=PLL_OUTPUT_DIV_P1 + 1</p> <p>PLL_OUTPUT_DIV_P1 is from 0 to 7.</p>
19	/	/	/
18:16	R/W	0x1	<p>PLL_P0</p> <p>PLL Output Divider P0</p> <p>P0=PLL_OUTPUT_DIV_P0 + 1</p> <p>PLL_OUTPUT_DIV_P0 is from 0 to 7.</p>
15:8	R/W	0x7F	<p>PLL_N</p> <p>PLL N</p> <p>N= PLL_N +1</p> <p>PLL_N is from 0 to 254.</p> <p>In application, PLL_N shall be more than or equal to 12.</p>
7:6	R/W	0x0	<p>PLL_UNLOCK_MDSEL</p> <p>PLL Unlock Level</p> <p>00: 21-29 Clock Cycles</p> <p>01: 22-28 Clock Cycles</p> <p>1X: 20-30 Clock Cycles</p>
5	R/W	0x0	<p>PLL_LOCK_MDSEL</p> <p>PLL Lock Level</p> <p>0: 24-26 Clock Cycles</p> <p>1: 23-27 Clock Cycles</p>
4:2	/	/	/
1	R/W	0x0	<p>PLL_INPUT_DIV2</p> <p>PLL Input Divider M</p> <p>M=PLL_INPUT_DIV_M + 1</p> <p>PLL_INPUT_DIV_M is from 0 to 1.</p>
0	/	/	/

### 3.3.6.9 0x0110 PLL\_DDR Pattern0 Control Register (Default Value: 0x0000\_0000)

Offset: 0x0110			Register Name: PLL_DDR_PAT0_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	SIG_DELTA_PAT_EN Sigma-Delta Pattern Enable 0: Disable 1: Enable
30:29	R/W	0x0	SPR_FREQ_MODE Spread Frequency Mode 00: DC=0 01: DC=1 10: Triangular(1-bit) 11: Triangular(n-bit)
28:20	R/W	0x0	WAVE_STEP Wave Step
19	R/W	0x0	SDM_CLK_SEL SDM Clock Select 0: 24 MHz 1: 12 MHz <b>Note: When the PLL_INPUT_DIV2 is 1, the bit needs to be set to 1.</b>
18:17	R/W	0x0	FREQ Frequency 00: 31.5 kHz 01: 32 kHz 10: 32.5 kHz 11: 33 kHz
16:0	R/W	0x0	WAVE_BOT Wave Bottom

### 3.3.6.10 0x0114 PLL\_DDR Pattern1 Control Register (Default Value: 0x0000\_0000)

Offset: 0x0114			Register Name: PLL_DDR_PAT1_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:25	/	/	/

Offset: 0x0114			Register Name: PLL_DDR_PAT1_CTRL_REG
Bit	Read/Write	Default/Hex	Description
24	R/W	0x0	DITHER_EN Dither Enable 0: Disable 1: Enable
23:21	/	/	/
20	R/W	0x0	FRAC_EN Fraction Enable 0: Disable 1: Enable
19:17	/	/	/
16:0	R/W	0x0	FRAC_IN Fraction In

### 3.3.6.11 0x0120 PLL\_PERI Pattern0 Control Register (Default Value: 0x0000\_0000)

Offset: 0x0120			Register Name: PLL_PERI_PAT0_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	SIG_DELT_PAT_EN Sigma-Delta Pattern Enable 0: Disable 1: Enable
30:29	R/W	0x0	SPR_FREQ_MODE Spread Frequency Mode 00: DC=0 01: DC=1 10: Triangular(1-bit) 11: Triangular(n-bit)
28:20	R/W	0x0	WAVE_STEP Wave Step
19	R/W	0x0	SDM_CLK_SEL SDM Clock Select 0: 24 MHz 1: 12 MHz <b>Note: When the PLL_INPUT_DIV2 is 1, the bit needs to be set to 1.</b>

Offset: 0x0120			Register Name: PLL_PERI_PATO_CTRL_REG
Bit	Read/Write	Default/Hex	Description
18:17	R/W	0x0	FREQ Frequency 00: 31.5 kHz 01: 32 kHz 10: 32.5 kHz 11: 33 kHz
16:0	R/W	0x0	WAVE_BOT Wave Bottom

### 3.3.6.12 0x0124 PLL\_PERI Pattern1 Control Register (Default Value: 0x0000\_0000)

Offset: 0x0124			Register Name: PLL_PERI_PAT1_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:25	/	/	/
24	R/W	0x0	DITHER_EN Dither Enable 0: Disable 1: Enable
23:21	/	/	/
20	R/W	0x0	FRAC_EN Fraction Enable 0: Disable 1: Enable
19:17	/	/	/
16:0	R/W	0x0	FRAC_IN Fraction In

**3.3.6.13 0x0140 PLL\_VIDEO0 Pattern0 Control Register (Default Value: 0x0000\_0000)**

Offset: 0x0140			Register Name: PLL_VIDEO0_PAT0_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	SIG_DELTA_PAT_EN Sigma-Delta Pattern Enable 0: Disable 1: Enable
30:29	R/W	0x0	SPR_FREQ_MODE Spread Frequency Mode 00: DC=0 01: DC=1 10: Triangular(1-bit) 11: Triangular(n-bit)
28:20	R/W	0x0	WAVE_STEP Wave Step
19	R/W	0x0	SDM_CLK_SEL SDM Clock Select 0: 24 MHz 1: 12 MHz <b>Note: When the PLL_INPUT_DIV2 is 1, the bit needs to be set to 1.</b>
18:17	R/W	0x0	FREQ Frequency 00: 31.5 kHz 01: 32 kHz 10: 32.5 kHz 11: 33 kHz
16:0	R/W	0x0	WAVE_BOT Wave Bottom

**3.3.6.14 0x0144 PLL\_VIDEO0 Pattern1 Control Register (Default Value: 0x0000\_0000)**

Offset: 0x0144			Register Name: PLL_VIDEO0_PAT1_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:25	/	/	/

Offset: 0x0144			Register Name: PLL_VIDEO0_PAT1_CTRL_REG
Bit	Read/Write	Default/Hex	Description
24	R/W	0x0	DITHER_EN Dither Enable 0: Disable 1: Enable
23:21	/	/	/
20	R/W	0x0	FRAC_EN Fraction Enable
19:17	/	/	/
16:0	R/W	0x0	FRAC_IN Fraction In

### 3.3.6.15 0x0148 PLL\_VIDEO1 Pattern0 Control Register (Default Value: 0x0000\_0000)

Offset: 0x0148			Register Name: PLL_VIDEO1_PAT0_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	SIG_DELT_PAT_EN Sigma-Delta Pattern Enable 0: Disable 1: Enable
30:29	R/W	0x0	SPR_FREQ_MODE Spread Frequency Mode 00: DC=0 01: DC=1 10: Triangular(1-bit) 11: Triangular(n-bit)
28:20	R/W	0x0	WAVE_STEP Wave Step
19	R/W	0x0	SDM_CLK_SEL SDM Clock Select 0: 24 MHz 1: 12 MHz <b>Note: When the PLL_INPUT_DIV2 is 1, the bit needs to be set to 1.</b>

Offset: 0x0148			Register Name: PLL_VIDEO1_PAT0_CTRL_REG
Bit	Read/Write	Default/Hex	Description
18:17	R/W	0x0	FREQ Frequency 00: 31.5 kHz 01: 32 kHz 10: 32.5 kHz 11: 33 kHz
16:0	R/W	0x0	WAVE_BOT Wave Bottom

### 3.3.6.16 0x014C PLL\_VIDEO1 Pattern1 Control Register (Default Value: 0x0000\_0000)

Offset: 0x014C			Register Name: PLL_VIDEO1_PAT1_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:25	/	/	/
24	R/W	0x0	DITHER_EN Dither Enable 0: Disable 1: Enable
23:21	/	/	/
20	R/W	0x0	FRAC_EN Fraction Enable
19:17	/	/	/
16:0	R/W	0x0	FRAC_IN Fraction In

### 3.3.6.17 0x0158 PLL\_VE Pattern0 Control Register (Default Value: 0x0000\_0000)

Offset: 0x0158			Register Name: PLL_VE_PATO_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	SIG_DELT_PAT_EN Sigma-Delta Pattern Enable 0: Disable 1: Enable

Offset: 0x0158			Register Name: PLL_VE_PATO_CTRL_REG
Bit	Read/Write	Default/Hex	Description
30:29	R/W	0x0	<p>SPR_FREQ_MODE Spread Frequency Mode 00: DC=0 01: DC=1 10: Triangular(1-bit) 11: Triangular(n-bit)</p>
28:20	R/W	0x0	<p>WAVE_STEP Wave Step</p>
19	R/W	0x0	<p>SDM_CLK_SEL SDM Clock Select 0: 24 MHz 1: 12 MHz <b>Note: When the PLL_INPUT_DIV2 is 1, the bit needs to be set to 1.</b></p>
18:17	R/W	0x0	<p>FREQ Frequency 00: 31.5 kHz 01: 32 kHz 10: 32.5 kHz 11: 33 kHz</p>
16:0	R/W	0x0	<p>WAVE_BOT Wave Bottom</p>

### 3.3.6.18 0x015C PLL\_VE Pattern1 Control Register (Default Value: 0x0000\_0000)

Offset: 0x015C			Register Name: PLL_VE_PAT1_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:25	/	/	/
24	R/W	0x0	<p>DITHER_EN Dither Enable 0: Disable 1: Enable</p>
23:21	/	/	/

Offset: 0x015C			Register Name: PLL_VE_PAT1_CTRL_REG
Bit	Read/Write	Default/Hex	Description
20	R/W	0x0	FRAC_EN Fraction Enable 0: Disable 1: Enable
19:17	/	/	/
16:0	R/W	0x0	FRAC_IN Fraction In

### 3.3.6.19 0x0178 PLL\_AUDIO0 Pattern0 Control Register (Default Value: 0x0000\_0000)

Offset: 0x0178			Register Name: PLL_AUDIO0_PAT0_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	SIG_DELT_PAT_EN Sigma-Delta Pattern Enable 0: Disable 1: Enable
30:29	R/W	0x0	SPR_FREQ_MODE Spread Frequency Mode 00: DC=0 01: DC=1 10: Triangular(1-bit) 11: Triangular(n-bit)
28:20	R/W	0x0	WAVE_STEP Wave Step
19	R/W	0x0	SDM_CLK_SEL SDM Clock Select 0: 24 MHz 1: 12 MHz <b>Note: When the PLL_INPUT_DIV2 is 1, the bit needs to be set to 1.</b>

Offset: 0x0178			Register Name: PLL_AUDIO0_PAT0_CTRL_REG
Bit	Read/Write	Default/Hex	Description
18:17	R/W	0x0	FREQ Frequency 00: 31.5 kHz 01: 32 kHz 10: 32.5 kHz 11: 33 kHz
16:0	R/W	0x0	WAVE_BOT Wave Bottom

### 3.3.6.20 0x017C PLL\_AUDIO0 Pattern1 Control Register (Default Value: 0x0000\_0000)

Offset: 0x017C			Register Name: PLL_AUDIO0_PAT1_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:25	/	/	/
24	R/W	0x0	DITHER_EN Dither Enable 0: Disable 1: Enable
23:21	/	/	/
20	R/W	0x0	FRAC_EN Fraction Enable 0: Disable 1: Enable
19:17	/	/	/
16:0	R/W	0x0	FRAC_IN Fraction In

### 3.3.6.21 0x0180 PLL\_AUDIO1 Pattern0 Control Register (Default Value: 0x0000\_0000)

Offset: 0x0180			Register Name: PLL_AUDIO1_PAT0_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	SIG_DELTA_PAT_EN Sigma-Delta Pattern Enable 0: Disable 1: Enable
30:29	R/W	0x0	SPR_FREQ_MODE Spread Frequency Mode 00: DC=0 01: DC=1 10: Triangular(1-bit) 11: Triangular(n-bit)
28:20	R/W	0x0	WAVE_STEP Wave Step
19	R/W	0x0	SDM_CLK_SEL SDM Clock Select 0: 24 MHz 1: 12 MHz <b>Note: When the PLL_INPUT_DIV2 is 1, the bit needs to be set to 1.</b>
18:17	R/W	0x0	FREQ Frequency 00: 31.5 kHz 01: 32 kHz 10: 32.5 kHz 11: 33 kHz
16:0	R/W	0x0	WAVE_BOT Wave Bottom

### 3.3.6.22 0x0184 PLL\_AUDIO1 Pattern1 Control Register (Default Value: 0x0000\_0000)

Offset: 0x0184			Register Name: PLL_AUDIO1_PAT1_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:25	/	/	/

Offset: 0x0184			Register Name: PLL_AUDIO1_PAT1_CTRL_REG
Bit	Read/Write	Default/Hex	Description
24	R/W	0x0	DITHER_EN Dither Enable 0: Disable 1: Enable
23:21	/	/	/
20	R/W	0x0	FRAC_EN Fraction Enable 0: Disable 1: Enable
19:17	/	/	/
16:0	R/W	0x0	FRAC_IN Fraction In

### 3.3.6.23 0x0300 PLL\_CPU Bias Register (Default Value: 0x8010\_0000)

Offset: 0x0300			Register Name: PLL_CPU_BIAS_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x1	PLL_VCO_RST_IN VCO reset in
30:21	/	/	/
20:16	R/W	0x10	PLL_CP PLL bias control
15:0	/	/	/

### 3.3.6.24 0x0310 PLL\_DDR Bias Register (Default Value: 0x0003\_0000)

Offset: 0x0310			Register Name: PLL_DDR_BIAS_REG
Bit	Read/Write	Default/Hex	Description
31:21	/	/	/
20:16	R/W	0x3	PLL_CP PLL bias control
15:0	/	/	/

**3.3.6.25 0x0320 PLL\_PERI Bias Register (Default Value: 0x0003\_0000)**

Offset: 0x0320			Register Name: PLL_PERI_BIAS_REG
Bit	Read/Write	Default/Hex	Description
31:21	/	/	/
20:16	R/W	0x3	PLL_CP PLL bias control
15:0	/	/	/

**3.3.6.26 0x0340 PLL\_VIDEO0 Bias Register (Default Value: 0x0003\_0000)**

Offset: 0x0340			Register Name: PLL_VIDEO0_BIAS_REG
Bit	Read/Write	Default/Hex	Description
31:21	/	/	/
20:16	R/W	0x3	PLL_CP PLL bias control
15:0	/	/	/

**3.3.6.27 0x0348 PLL\_VIDEO1 Bias Register (Default Value: 0x0003\_0000)**

Offset: 0x0348			Register Name: PLL_VIDEO1_BIAS_REG
Bit	Read/Write	Default/Hex	Description
31:21	/	/	/
20:16	R/W	0x3	PLL_CP PLL bias control
15:0	/	/	/

**3.3.6.28 0x0358 PLL\_VE Bias Register (Default Value: 0x0003\_0000)**

Offset: 0x0358			Register Name: PLL_VE_BIAS_REG
Bit	Read/Write	Default/Hex	Description
31:21	/	/	/
20:16	R/W	0x3	PLL_CP PLL bias control

<b>Offset: 0x0358</b>			<b>Register Name: PLL_VE_BIAS_REG</b>
<b>Bit</b>	<b>Read/Write</b>	<b>Default/Hex</b>	<b>Description</b>
15:0	/	/	/

### 3.3.6.29 0x0378 PLL\_AUDIO0 Bias Register (Default Value: 0x0003\_0000)

<b>Offset: 0x0378</b>			<b>Register Name: PLL_AUDIO0_BIAS_REG</b>
<b>Bit</b>	<b>Read/Write</b>	<b>Default/Hex</b>	<b>Description</b>
31:21	/	/	/
20:16	R/W	0x3	PLL_CP PLL bias control
15:0	/	/	/

### 3.3.6.30 0x0380 PLL\_AUDIO1 Bias Register (Default Value: 0x0003\_0000)

<b>Offset: 0x0380</b>			<b>Register Name: PLL_AUDIO1_BIAS_REG</b>
<b>Bit</b>	<b>Read/Write</b>	<b>Default/Hex</b>	<b>Description</b>
31:21	/	/	/
20:16	R/W	0x3	PLL_CP PLL bias control
15:0	/	/	/

### 3.3.6.31 0x0400 PLL\_CPU Tuning Register (Default Value: 0x4440\_4000)

<b>Offset: 0x0400</b>			<b>Register Name: PLL_CPU_TUN_REG</b>
<b>Bit</b>	<b>Read/Write</b>	<b>Default/Hex</b>	<b>Description</b>
31	/	/	/
30:28	R/W	0x4	PLL_VCO VCO range control
27	/	/	/
26:24	R/W	0x4	PLL_VCO_GAIN KVCO gain control
23	/	/	/

Offset: 0x0400			Register Name: PLL_CPU_TUN_REG
Bit	Read/Write	Default/Hex	Description
22:16	R/W	0x40	PLL_CNT_INT Counter initial control
15	R/W	0x0	PLL_REG_OD PLL-REG-OD0 for verify
14:8	R/W	0x40	PLL_B_IN PLL-B-IN [6:0] for verify
7	R/W	0x0	PLL_REG_OD1 PLL-REG-OD1 for verify
6:0	R	0x0	PLL_B_OUT PLL-B-OUT [6:0] for verify

### 3.3.6.32 0x0500 CPU\_AXI Configuration Register (Default Value: 0x0000\_0301)

Offset: 0x0500			Register Name: CPU_AXI_CFG_REG
Bit	Read/Write	Default/Hex	Description
31:27	/	/	/
26:24	R/W	0x0	CPU_CLK_SEL Clock Source Select 000: HOSC 001: CLK32K 010: CLK16M_RC 011: PLL_CPU/P 100: PLL_PERI(1X) 101: PLL_PERI(2X) 110: PLL_PERI(800M) CPU Clock = Clock Source CPU_AXI Clock = Clock Source/M <b>Note: The clock select is lack of glitch switching and supports dynamic configuration.</b>
23:18	/	/	/

Offset: 0x0500			Register Name: CPU_AXI_CFG_REG
Bit	Read/Write	Default/Hex	Description
17:16	R/W	0x0	<p>PLL_CPU_OUT_EXT_DIVP PLL Output External Divider P</p> <p>00: 1 01: 2 10: 4 11: /</p> <p>When the output clock is less than 288 MHz, the clock frequency can be got by dividing P.</p>
15:10	/	/	/
9:8	R/W	0x3	<p>CPU_DIV2. Factor N (N = FACTOR_N +1) FACTOR_N is from 1 to 3.</p> <p><b>Note: The clock division is lack of glitch switching and supports dynamic configuration.</b></p>
7:2	/	/	/
1:0	R/W	0x1	<p>CPU_DIV1. Factor M (M= FACTOR_M +1) FACTOR_M is from 1 to 3.</p> <p><b>Note: The clock division is lack of glitch switching and supports dynamic configuration.</b></p>

### 3.3.6.33 0x0504 CPU\_GATING Configuration Register (Default Value: 0x8000\_0000)

Offset: 0x0504			Register Name: CPU_GATING_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x1	<p>CPU_GATING Gating Special Clock 0: Clock is OFF 1: Clock is ON</p>
30:16	/	/	/
15:0	W	0x0	<p>CPU_GATING_FIELD CPU Gating Field If CPU_GATING_FIELD==16'h16AA, the bit31 can be configured.</p>

### 3.3.6.34 0x0510 PSI Clock Register (Default Value: 0x0000\_0000)

Offset: 0x0510			Register Name: PSI_CLK_REG
Bit	Read/Write	Default/Hex	Description
31:26	/	/	/
25:24	R/W	0x0	<p>CLK_SRC_SEL Clock Source Select 000: HOSC 001: CLK32K 010: CLK16M_RC 011: PLL_PERI(1X) PSI_CLK = Clock Source/M/N.</p> <p><b>Note: The clock select is lack of glitch switching and supports dynamic configuration.</b></p>
23:10	/	/	/
9:8	R/W	0x0	<p>FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8</p> <p><b>Note: The clock division is lack of glitch switching and supports dynamic configuration.</b></p>
7:2	/	/	/
1:0	R/W	0x0	<p>FACTOR_M Factor M. M= FACTOR_M +1 FACTOR_M is from 0 to 3.</p> <p><b>Note: The clock division is lack of glitch switching and supports dynamic configuration.</b></p>

### 3.3.6.35 0x0520 APB0 Clock Register (Default Value: 0x0000\_0000)

Offset: 0x0520			Register Name: APB0_CLK_REG
Bit	Read/Write	Default/Hex	Description
31:26	/	/	/

Offset: 0x0520			Register Name: APB0_CLK_REG
Bit	Read/Write	Default/Hex	Description
25:24	R/W	0x0	<p>CLK_SRC_SEL Clock Source Select 00: HOSC 01: CLK32K 10: PSI_CLK 11: PLL_PERI(1X) APB0_CLK = Clock Source/M/N.</p> <p><b>Note: The clock select is lack of glitch switching and supports dynamic configuration.</b></p>
23:10	/	/	/
9:8	R/W	0x0	<p>FACTOR_N. Factor N 00: 1 01: 2 10: 4 11: 8</p> <p><b>Note: The clock division is lack of glitch switching and supports dynamic configuration.</b></p>
7:5	/	/	/
4:0	R/W	0x0	<p>FACTOR_M Factor M M = FACTOR_M+1 FACTOR_M is from 0 to 31.</p> <p><b>Note: The clock division is lack of glitch switching and supports dynamic configuration.</b></p>

### 3.3.6.36 0x0524 APB1 Clock Register (Default Value: 0x0000\_0000)

Offset: 0x0524			Register Name: APB1_CLK_REG
Bit	Read/Write	Default/Hex	Description
31:26	/	/	/

Offset: 0x0524			Register Name: APB1_CLK_REG
Bit	Read/Write	Default/Hex	Description
25:24	R/W	0x0	<p>CLK_SRC_SEL Clock Source Select 00: HOSC 01: CLK32K 10: PSI_CLK 11: PLL_PERI(1X) APB1_CLK = Clock Source/M/N</p> <p><b>Note: The clock select is lack of glitch switching and supports dynamic configuration.</b></p>
23:10	/	/	/
9:8	R/W	0x0	<p>FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8</p> <p><b>Note: The clock division is lack of glitch switching and supports dynamic configuration.</b></p>
7:5	/	/	/
4:0	R/W	0x0	<p>FACTOR_M Factor M M = FACTOR_M+1 FACTOR_M is from 0 to 31.</p> <p><b>Note: The clock division is lack of glitch switching and supports dynamic configuration.</b></p>

### 3.3.6.37 0x0540 MBUS Clock Register (Default Value: 0xC000\_0000)

Offset: 0x0540			Register Name: MBUS_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	/	/	/
30	R/W	0x1	<p>MBUS_RST MBUS Reset 0: Assert 1: De-assert</p>

Offset: 0x0540			Register Name: MBUS_CLK_REG
Bit	Read/Write	Default/Hex	Description
29:0	/	/	/



The MBUS clock is from the 4 frequency-division of PLL\_DDR, and it has the same source with the DRAM clock.

### 3.3.6.38 0x0600 DE Clock Register (Default Value: 0xC000\_0000)

Offset: 0x0600			Register Name: DE_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	DE_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON DE_CLK = Clock Source/M.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: PLL_PERI(2X) 001: PLL_VIDEO0(4X) 010: PLL_VIDEO1(4X) 011: PLL_AUDIO1(DIV2)
23:5	/	/	/
4:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 31.

**3.3.6.39 0x060C DE Bus Gating Reset Register (Default Value: 0x0000\_0000)**

Offset: 0x060C			Register Name: DE_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	DE_RST DE Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	DE_GATING DE Gating Clock 0: Mask 1: Pass

**3.3.6.40 0x0620 DI Clock Register (Default Value: 0x0000\_0000)**

Offset: 0x0620			Register Name: DI_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	DI_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON DI_CLK = Clock Source/M.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: PLL_PERI(2X) 001: PLL_VIDEO0(4X) 010: PLL_VIDEO1(4X) 011: PLL_AUDIO1(DIV2)
23:5	/	/	/
4:0	R/W	0x0	FACTOR_M Factor M M = FACTOR_M +1 FACTOR_M is from 0 to 31.

**3.3.6.41 0x062C DI Bus Gating Reset Register (Default Value: 0x0000\_0000)**

Offset: 0x062C			Register Name: DI_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	DI_RST DI Reset 0: Assert 1: DE-assert
15:1	/	/	/
0	R/W	0x0	DI_GATING DI Gating Clock 0: Mask 1: Pass

**3.3.6.42 0x0630 G2D Clock Register (Default Value: 0x0000\_0000)**

Offset: 0x0630			Register Name: G2D_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	G2D_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON G2D_CLK = Clock Source/M.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: PLL_PERI(2X) 001: PLL_VIDEO0(4X) 010: PLL_VIDEO1(4X) 011: PLL_AUDIO1(DIV2)
23:5	/	/	/
4:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 31.

**3.3.6.43 0x063C G2D Bus Gating Reset Register (Default Value: 0x0000\_0000)**

Offset: 0x063C			Register Name: G2D_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	G2D_RST G2D Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	G2D_GATING G2D Gating Clock 0: Mask 1: Pass

**3.3.6.44 0x0680 CE Clock Register (Default Value: 0x0000\_0000)**

Offset: 0x0680			Register Name: CE_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	CE_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON CE_CLK = Clock Source/M/N.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: HOSC 01: PLL_PERI(2X) 10: PLL_PERI(1X)
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8

Offset: 0x0680			Register Name: CE_CLK_REG
Bit	Read/Write	Default/Hex	Description
7:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 15.

### 3.3.6.45 0x068C CE Bus Gating Reset Register (Default Value: 0x0000\_0000)

Offset: 0x068C			Register Name: CE_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	CE_RST CE Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	CE_GATING CE Gating Clock 0: Mask 1: Pass

### 3.3.6.46 0x0690 VE Clock Register (Default Value: 0x0000\_0000)

Offset: 0x0690			Register Name: VE_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	VE_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON VE_CLK = Clock Source/M.
30:25	/	/	/

Offset: 0x0690			Register Name: VE_CLK_REG
Bit	Read/Write	Default/Hex	Description
24	R/W	0x0	CLK_SRC_SEL Clock Source Select 0: PLL_VE 1: PLL_PERI(2X)
23:5	/	/	/
4:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 31.

### 3.3.6.47 0x069C VE Bus Gating Reset Register (Default Value: 0x0000\_0000)

Offset: 0x069C			Register Name: VE_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	VE_RST VE Reset For VE_PROT 0: Assert 1: DE-assert
15:1	/	/	/
0	R/W	0x0	VE_GATING Gating Clock For VE_PROT 0: Mask 1: Pass

### 3.3.6.48 0x070C DMA Bus Gating Reset Register (Default Value: 0x0000\_0000)

Offset: 0x070C			Register Name: DMA_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/

Offset: 0x070C			Register Name: DMA_BGR_REG
Bit	Read/Write	Default/Hex	Description
16	R/W	0x0	DMA_RST DMA Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	DMA_GATING DMA Gating Clock 0: Mask 1: Pass <b>Note: The working clock is from PSI_CLK.</b>

### 3.3.6.49 0x073C HSTIMER Bus Gating Reset Register (Default Value: 0x0000\_0000)

Offset: 0x073C			Register Name: HSTIMER_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	HSTIMER_RST HSTIMER Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	HSTIMER_GATING Gating Clock for HSTIMER 0: Mask 1: Pass

**3.3.6.50 0x0740 AVS Clock Register (Default Value: 0x0000\_0000)**

Offset: 0x0740			Register Name: AVS_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	AVS_CLK_GATING Gating Clock The AVS_CLK is from HOSC. 0: Clock is OFF 1: Clock is ON
30:0	/	/	/

**3.3.6.51 0x078C DBGSYS Bus Gating Reset Register (Default Value: 0x0000\_0000)**

Offset: 0x078C			Register Name: DBGSYS_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	DBGSYS_RST DBGSYS Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	DBGSYS_GATING Gating Clock for DBGSYS The clock of DBGSYS is from HOSC. 0: Mask 1: Pass

**3.3.6.52 0x07AC PWM Bus Gating Reset Register (Default Value: 0x0000\_0000)**

Offset: 0x07AC			Register Name: PWM_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	PWM_RST PWM Reset 0: Assert 1: De-assert

Offset: 0x07AC			Register Name: PWM_BGR_REG
Bit	Read/Write	Default/Hex	Description
15:1	/	/	/
0	R/W	0x0	PWM_GATING Gating Clock for PWM 0: Mask 1: Pass

### 3.3.6.53 0x07BC IOMMU Bus Gating Reset Register (Default Value: 0x0000\_0000)

Offset: 0x07BC			Register Name: IOMMU_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	IOMMU_GATING Gating Clock for IOMMU 0: Mask 1: Pass

### 3.3.6.54 0x0800 DRAM Clock Register (Default Value: 0x8000\_0000)

Offset: 0x0800			Register Name: DRAM_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x1	DRAM_CLK_GATING Gating Clock DRAM_CLK = Clock Source/M/N. 0: Clock is OFF 1: Clock is ON
30:28	/	/	/

Offset: 0x0800			Register Name: DRAM_CLK_REG
Bit	Read/Write	Default/Hex	Description
27	R/WAC	0x0	<p>SDRCLK_UPD SDRCLK Configuration 0 Update 0: Invalid 1: Valid When SDRCLK_UPD is set to 1, dram_clk_sel/dram_div2/dram_clk1 will be updated. The SDRCLK_UPD will be automatically cleared after it is valid. Here supports dram req/ack signal to update dram_clk_sel/dram_div2/dram_clk1.</p>
26:24	R/W	0x0	<p>DRAM_CLK_SEL Clock Source Select 00: PLL_DDR 01: PLL_AUDIO1(DIV2) 10: PLL_PERI(2X) 11: PLL_PERI(800M)</p> <p><b>Note: The clock select is lack of glitch switching and supports dynamic configuration.</b></p>
23:10	/	/	/
9:8	R/W	0x0	<p>DRAM_DIV2 Factor N 00: 1 01: 2 10: 4 11: 8</p>
7:2	/	/	/
1:0	R/W	0x0	<p>DRAM_DIV1 Factor M M = FACTOR_M + 1 FACTOR_M is from 0 to 3.</p> <p><b>Note: The clock division is lack of glitch switching and supports dynamic configuration.</b></p>

**3.3.6.55 0x0804 MBUS Master Clock Gating Register (Default Value: 0x0000\_0000)**

Offset: 0x0804			Register Name: MBUS_MAT_CLK_GATING_REG
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10	R/W	0x0	G2D_MCLK_EN Gating MBUS Clock For G2D 0: Mask 1: Pass
9	/	/	/
8	R/W	0x0	Reserved
7	R/W	0x0	Reserved
6:3	/	/	/
2	R/W	0x0	CE_MCLK_EN Gating MBUS Clock For CE 0: Mask 1: Pass
1	R/W	0x0	VE_MCLK_EN Gating MBUS Clock For VE 0: Mask 1: Pass
0	R/W	0x0	DMA_MCLK_EN Gating MBUS Clock For DMA 0: Mask 1: Pass

**3.3.6.56 0x080C DRAM Bus Gating Reset Register (Default Value: 0x0000\_0001)**

Offset: 0x080C			Register Name: DRAM_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	DRAM_RST DRAM Reset 0: Assert 1: De-assert
15:1	/	/	/

Offset: 0x080C			Register Name: DRAM_BGR_REG
Bit	Read/Write	Default/Hex	Description
0	R/W	0x1	DRAM_GATING Gating Clock for DRAM 0: Mask 1: Pass

### 3.3.6.57 0x0830 SMHC0 Clock Register (Default Value: 0x0000\_0000)

Offset: 0x0830			Register Name: SMHC0_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	SMHC0_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON SMHC0_CLK = Clock Source/M/N.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: HOSC 001: PLL_PERI(1X) 010: PLL_PERI(2X) 011: PLL_AUDIO1(DIV2) Others: Reserved
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M M = FACTOR_M +1 FACTOR_M is from 0 to 15.

**3.3.6.58 0x0834 SMHC1 Clock Register (Default Value: 0x0000\_0000)**

Offset: 0x0834			Register Name: SMHC1_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	SMHC1_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/M/N.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: HOSC 001: PLL_PERI(1X) 010: PLL_PERI(2X) 011: PLL_AUDIO1(DIV2) Others: Reserved
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 15.

**3.3.6.59 0x0838 SMHC2 Clock Register (Default Value: 0x0000\_0000)**

Offset: 0x0838			Register Name: SMHC2_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	SMHC2_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/M/N.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: HOSC 001: PLL_PERI(1X) 010: PLL_PERI(2X) 011: PLL_PERI(800M) 100: PLL_AUDIO1(DIV2) Others: Reserved
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 15.

**3.3.6.60 0x084C SMHC Bus Gating Reset Register (Default Value: 0x0000\_0000)**

Offset: 0x084C			Register Name: SMHC_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:19	/	/	/

Offset: 0x084C			Register Name: SMHC_BGR_REG
Bit	Read/Write	Default/Hex	Description
18	R/W	0x0	SMHC2_RST SMHC2 Reset 0: Assert 1: De-assert
17	R/W	0x0	SMHC1_RST SMHC1 Reset 0: Assert 1: De-assert
16	R/W	0x0	SMHCO_RST SMHCO Reset 0: Assert 1: De-assert
15:3	/	/	/
2	R/W	0x0	SMHC2_GATING Gating Clock for SMHC2 0: Mask 1: Pass
1	R/W	0x0	SMHC1_GATING Gating Clock for SMHC1 0: Mask 1: Pass
0	R/W	0x0	SMHCO_GATING Gating Clock for SMHCO 0: Mask 1: Pass

### 3.3.6.61 0x090C UART Bus Gating Reset Register (Default Value: 0x0000\_0000)



The clock of the UART is from APB1.

Offset: 0x090C			Register Name: UART_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:22	/	/	/
21	R/W	0x0	UART5_RST UART5 Reset 0: Assert 1: De-assert
20	R/W	0x0	UART4_RST UART4 Reset 0: Assert 1: De-assert
19	R/W	0x0	UART3_RST UART3 Reset 0: Assert 1: De-assert
18	R/W	0x0	UART2_RST UART2 Reset 0: Assert 1: De-assert
17	R/W	0x0	UART1_RST UART1 Reset 0: Assert 1: De-assert
16	R/W	0x0	UART0_RST UART0 Reset 0: Assert 1: De-assert
15:6	/	/	/
5	R/W	0x0	UART5_GATING Gating Clock for UART5 0: Mask 1: Pass
4	R/W	0x0	UART4_GATING Gating Clock for UART4 0: Mask 1: Pass

Offset: 0x090C			Register Name: UART_BGR_REG
Bit	Read/Write	Default/Hex	Description
3	R/W	0x0	UART3_GATING Gating Clock for UART3 0: Mask 1: Pass
2	R/W	0x0	UART2_GATING Gating Clock for UART2 0: Mask 1: Pass
1	R/W	0x0	UART1_GATING Gating Clock for UART1 0: Mask 1: Pass
0	R/W	0x0	UART0_GATING Gating Clock for UART0 0: Mask 1: Pass

### 3.3.6.62 0x091C TWI Bus Gating Reset Register (Default Value: 0x0000\_0000)



#### NOTE

The clock of the TWI is from APB1.

Offset: 0x091C			Register Name: TWI_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:20	/	/	/
19	R/W	0x0	TWI3_RST TWI3 Reset 0: Assert 1: De-assert
18	R/W	0x0	TWI2_RST TWI2 Reset 0: Assert 1: De-assert

Offset: 0x091C			Register Name: TWI_BGR_REG
Bit	Read/Write	Default/Hex	Description
17	R/W	0x0	TWI1_RST TWI1 Reset 0: Assert 1: De-assert
16	R/W	0x0	TWI0_RST TWI0 Reset 0: Assert 1: De-assert
15:4	/	/	/
3	R/W	0x0	TWI3_GATING Gating Clock for TWI3 0: Mask 1: Pass
2	R/W	0x0	TWI2_GATING Gating Clock for TWI2 0: Mask 1: Pass
1	R/W	0x0	TWI1_GATING Gating Clock for TWI1 0: Mask 1: Pass
0	R/W	0x0	TWI0_GATING Gating Clock for TWI0 0: Mask 1: Pass

### 3.3.6.63 0x0940 SPI0 Clock Register (Default Value: 0x0000\_0000)

Offset: 0x0940			Register Name: SPI0_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	SPI0_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/M/N.

Offset: 0x0940			Register Name: SPI0_CLK_REG
Bit	Read/Write	Default/Hex	Description
30:27	/	/	/
26:24	R/W	0x0	<p>CLK_SRC_SEL Clock Source Select 000: HOSC 001: PLL_PERI(1X) 010: PLL_PERI(2X) 011: PLL_AUDIO1(DIV2) 100: PLL_AUDIO1(DIV5) Others: Reserved</p>
23:10	/	/	/
9:8	R/W	0x0	<p>FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8</p>
7:4	/	/	/
3:0	R/W	0x0	<p>FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 15.</p>

### 3.3.6.64 0x0944 SPI1 Clock Register (Default Value: 0x0000\_0000)

Offset: 0x0944			Register Name: SPI1_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<p>SCLK_GATING Gating Special Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/M/N.</p>
30:27	/	/	/

Offset: 0x0944			Register Name: SPI1_CLK_REG
Bit	Read/Write	Default/Hex	Description
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: HOSC 001: PLL_PERI(1X) 010: PLL_PERI(2X) 011: PLL_AUDIO1(DIV2) 100: PLL_AUDIO1(DIV5) Others: /
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 15.

### 3.3.6.65 0x096C SPI Bus Gating Reset Register (Default Value: 0x0000\_0000)

Offset: 0x096C			Register Name: SPI_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
17	R/W	0x0	SPI1_RST SPI1 Reset 0: Assert 1: De-assert
16	R/W	0x0	SPI0_RST SPI0 Reset 0: Assert 1: De-assert
15:2	/	/	/

Offset: 0x096C			Register Name: SPI_BGR_REG
Bit	Read/Write	Default/Hex	Description
1	R/W	0x0	SPI1_GATING Gating Clock for SPI1 0: Mask 1: Pass
0	R/W	0x0	SPI0_GATING Gating Clock for SPI0 0: Mask 1: Pass

### 3.3.6.66 0x0970 EMAC\_25M Clock Register (Default: 0x0000\_0000)

Offset: 0x0970			Register Name: EMAC_25M_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	EMAC_25M_CLK_GATING Gating Special Clock 0: Clock is OFF 1: Clock is ON CLK = PLL_PERI(1X)/24 = 25 MHz
30	R/W	0x0	EMAC_25M_CLK_SRC_GATING Gating the Source Clock of Special Clock It is for low power design. 0: Clock is OFF 1: Clock is ON
29:0	/	/	/

### 3.3.6.67 0x097C EMAC Bus Gating Reset Register (Default Value: 0x0000\_0000)

Offset: 0x097C			Register Name: EMAC_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	EMAC_RST EMAC Reset 0: Assert 1: De-assert

Offset: 0x097C			Register Name: EMAC_BGR_REG
Bit	Read/Write	Default/Hex	Description
15:1	/	/	/
0	R/W	0x0	EMAC_GATING Gating Clock for EMAC 0: Mask 1: Pass

### 3.3.6.68 0x09C0 IRTX Clock Register (Default: 0x0000\_0000)

Offset: 0x09C0			Register Name: IRTX_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	IRTX_CLK_GATING Gating Clock for IRTX 0: Clock is OFF 1: Clock is ON IRTX_CLK = Clock Source/M/N.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 0: HOSC 1: PLL_PERI(1X)
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 15.

### 3.3.6.69 0x09CC IRTX Bus Gating Reset Register (Default Value: 0x0000\_0000)

Offset: 0x09CC			Register Name: IRTX_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	IRTX_RST IRTX Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	IRTX_GATING Gating Clock for IRTX 0: Mask 1: Pass

### 3.3.6.70 0x09EC GPADC Bus Gating Reset Register (Default Value: 0x0000\_0000)

Offset: 0x09EC			Register Name: GPADC_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	GPADC_RST GPADC Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	GPADC_GATING Gating Clock For GPADC 0: Mask 1: Pass

### 3.3.6.71 0x09FC THS Bus Gating Reset Register (Default Value: 0x0000\_0000)

Offset: 0x09FC			Register Name: THS_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/

Offset: 0x09FC			Register Name: THS_BGR_REG
Bit	Read/Write	Default/Hex	Description
16	R/W	0x0	THS_RST THS Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	THS_GATING Gating Clock For THS 0: Mask 1: Pass

### 3.3.6.72 0x0A14 I2S/PCM1 Clock Register (Default Value: 0x0000\_0000)

Offset: 0x0A14			Register Name: I2S/PCM1_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	I2S/PCM1_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON I2S/PCM1_CLK = Clock Source/M/N.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: PLL_AUDIO0(1X) 01: PLL_AUDIO0(4X) 10: PLL_AUDIO1(DIV2) 11: PLL_AUDIO1(DIV5)
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8

Offset: 0x0A14			Register Name: I2S/PCM1_CLK_REG
Bit	Read/Write	Default/Hex	Description
7:5	/	/	/
4:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 31.

### 3.3.6.73 0x0A18 I2S/PCM2 Clock Register (Default Value: 0x0000\_0000)

Offset: 0x0A18			Register Name: I2S/PCM2_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	I2S/PCM2_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON I2S/PCM2_CLK = Clock Source/M/N.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: PLL_AUDIO0(1X) 001: PLL_AUDIO0(4X) 010: PLL_AUDIO1(DIV2) 011: PLL_AUDIO1(DIV5)
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:5	/	/	/
4:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 31.

**3.3.6.74 0x0A1C I2S/PCM2\_ASRC Clock Register (Default Value: 0x0000\_0000)**

Offset: 0x0A1C			Register Name: I2S/PCM2_ASRC_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	I2S/PCM2_ASRC_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON I2S/PCM2_ASRC_CLK = Clock Source/M/N.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: PLL_AUDIO0(4X) 01: PLL_PERI(1X) 10: PLL_AUDIO1(DIV2) 11: PLL_AUDIO1(DIV5)
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:5	/	/	/
4:0	R/W	0x0	FACTOR_M Factor M M = FACTOR_M +1 FACTOR_M is from 0 to 31.

**3.3.6.75 0x0A20 I2S/PCM Bus Gating Reset Register (Default Value: 0x0000\_0000)**

Offset: 0x0A20			Register Name: I2S/PCM_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:19	/	/	/

Offset: 0x0A20			Register Name: I2S/PCM_BGR_REG
Bit	Read/Write	Default/Hex	Description
18	R/W	0x0	I2S/PCM2_RST I2S/PCM2 Reset 0: Assert 1: De-assert
17	R/W	0x0	I2S/PCM1_RST I2S/PCM1 Reset 0: Assert 1: De-assert
16	R/W	0x0	Reserved
15:3	/	/	/
2	R/W	0x0	I2S/PCM2_GATING Gating Clock for I2S/PCM2 0: Mask 1: Pass
1	R/W	0x0	I2S/PCM1_GATING Gating Clock for I2S/PCM1 0: Mask 1: Pass
0	R/W	0x0	Reserved

### 3.3.6.76 0x0A24 OWA\_TX Clock Register (Default Value: 0x0000\_0000)

Offset: 0x0A24			Register Name: OWA_TX_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	OWA_TX_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON OWA_TX_CLK = Clock Source/M/N.
30:27	/	/	/

Offset: 0x0A24			Register Name: OWA_TX_CLK_REG
Bit	Read/Write	Default/Hex	Description
26:24	R/W	0x0	<p>CLK_SRC_SEL Clock Source Select</p> <p>00: PLL_AUDIO0(1X) 01: PLL_AUDIO0(4X) 10: PLL_AUDIO1(DIV2) 11: PLL_AUDIO1(DIV5)</p>
23:10	/	/	/
9:8	R/W	0x0	<p>FACTOR_N Factor N</p> <p>00: 1 01: 2 10: 4 11: 8</p>
7:5	/	/	/
4:0	R/W	0x0	<p>FACTOR_M Factor M</p> <p>M= FACTOR_M +1</p> <p>FACTOR_M is from 0 to 31.</p>

### 3.3.6.77 0x0A2C OWA Bus Gating Reset Register (Default Value: 0x0000\_0000)

Offset: 0x0A2C			Register Name: OWA_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	<p>OWA_RST OWA Reset</p> <p>0: Assert 1: De-assert</p>
15:1	/	/	/
0	R/W	0x0	<p>OWA_GATING Gating Clock for OWA</p> <p>0: Mask 1: Pass</p>

**3.3.6.78 0x0A40 DMIC Clock Register (Default Value: 0x0000\_0000)**

Offset: 0x0A40			Register Name: DMIC_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	DMIC_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON DMIC_CLK = Clock Source/M/N.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: PLL_AUDIO0(1X) 01: PLL_AUDIO1(DIV2) 10: PLL_AUDIO1(DIV5)
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:5	/	/	/
4:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 31.

**3.3.6.79 0x0A4C DMIC Bus Gating Reset Register (Default Value: 0x0000\_0000)**

Offset: 0x0A4C			Register Name: DMIC_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	DMIC_RST DMIC Reset 0: Assert 1: De-assert

Offset: 0x0A4C			Register Name: DMIC_BGR_REG
Bit	Read/Write	Default/Hex	Description
15:1	/	/	/
0	R/W	0x0	DMIC_GATING Gating Clock for DMIC 0: Mask 1: Pass

### 3.3.6.80 0x0A50 AUDIO\_CODEC\_DAC Clock Register (Default Value: 0x0000\_0000)

Offset: 0x0A50			Register Name: AUDIO_CODEC_DAC_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	AUDIO_CODEC_DAC_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON AUDIO_CODEC_DAC_CLK = Clock Source/M/N.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SE. Clock Source Select 00: PLL_AUDIO0(1X) 01: PLL_AUDIO1(DIV2) 10: PLL_AUDIO1(DIV5)
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:5	/	/	/
4:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 31.

### 3.3.6.81 0x0A54 AUDIO\_CODEC\_ADC Clock Register (Default Value: 0x0000\_0000)

Offset: 0x0A54			Register Name: AUDIO_CODEC_ADC_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	AUDIO_CODEC_ADC_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON AUDIO_CODEC_ADC_CLK = Clock Source/M/N.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SE Clock Source Select 00: PLL_AUDIO0(1X) 01: PLL_AUDIO1(DIV2) 10: PLL_AUDIO1(DIV5)
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:5	/	/	/
4:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 31.

### 3.3.6.82 0x0A5C AUDIO\_CODEC Bus Gating Reset Register (Default Value: 0x0000\_0000)

Offset: 0x0A5C			Register Name: AUDIO_CODEC_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	AUDIO_CODEC_RST AUDIO_CODEC Reset 0: Assert 1: De-assert

Offset: 0x0A5C			Register Name: AUDIO_CODEC_BGR_REG
Bit	Read/Write	Default/Hex	Description
15:1	/	/	/
0	R/W	0x0	AUDIO_CODEC_GATING Gating Clock For AUDIO_CODEC 0: Mask 1: Pass

### 3.3.6.83 0x0A70 USBO Clock Register (Default Value: 0x0000\_0000)

Offset: 0x0A70			Register Name: USBO_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	USBO_CLKEN Gating Special Clock For OHCI0 0: Clock is OFF 1: Clock is ON
30	R/W	0x0	USBPHY0_RSTN USB PHY0 Reset 0: Assert 1: De-assert
29:26	/	/	/
25:24	R/W	0x0	USBO_CLK12M_SEL OHCI0 12M Source Select 00: 12M divided from 48 MHz 01: 12M divided from 24 MHz 10: RTC_32K 11: /
23:0	/	/	/

**3.3.6.84 0x0A74 USB1 Clock Register (Default Value: 0x0000\_0000)**

Offset: 0x0A74			Register Name: USB1_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	USB1_CLKEN Gating Special Clock For OHCI1 0: Clock is OFF 1: Clock is ON
30	R/W	0x0	USBPHY1_RSTN USB PHY1 Reset 0: Assert 1: De-assert
29:26	/	/	/
25:24	R/W	0x0	USB1_CLK12M_SEL OHCI0 12M Source Select 00: 12M divided from 48 MHz 01: 12M divided from 24 MHz 10: RTC_32K 11: /
23:0	/	/	/

**3.3.6.85 0x0A8C USB Bus Gating Reset Register (Default Value: 0x0000\_0000)**

Offset: 0x0A8C			Register Name: USB_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:25	/	/	/
24	R/W	0x0	USBOTGO_RST USBOTGO Reset 0: Assert 1: De-assert
23:22	/	/	/
21	R/W	0x0	USBEHCI1_RST USBEHCI1 Reset 0: Assert 1: De-assert

Offset: 0x0A8C			Register Name: USB_BGR_REG
Bit	Read/Write	Default/Hex	Description
20	R/W	0x0	USBEHCI0_RST USBEHCI0 Reset 0: Assert 1: De-assert
19:18	/	/	/
17	R/W	0x0	USBOHCI1_RST USBOHCI1 Reset 0: Assert 1: De-assert
16	R/W	0x0	USBOHCI0_RST USBOHCI0 Reset 0: Assert 1: De-assert
15:9	/	/	/
8	R/W	0x0	USBOTG0_GATING Gating Clock For USBOTG0 0: Mask 1: Pass
7:6	/	/	/
5	R/W	0x0	USBEHCI1_GATING Gating Clock For USBEHCI1 0: Mask 1: Pass
4	R/W	0x0	USBEHCI0_GATING Gating Clock For USBEHCI0 0: Mask 1: Pass
3:2	/	/	/
1	R/W	0x0	USBOHCI1_GATING Gating Clock For USBOHCI1 0: Mask 1: Pass
0	R/W	0x0	USBOHCI0_GATING Gating Clock For USBOHCI0 0: Mask 1: Pass

**3.3.6.86 0x0ABC DPSS\_TOP Bus Gating Reset Register (Default Value: 0x0000\_0000)**

Offset: 0x0ABC			Register Name: DPSS_TOP_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	DPSS_TOP_RST DPSS_TOP Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	DPSS_TOP_GATING Gating Clock For DPSS_TOP 0: Mask 1: Pass

**3.3.6.87 0x0B04 HDMI\_24M Clock Register (Default Value: 0x0000\_0000)**

Offset: 0x0B04			Register Name: HDMI_24M_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	HDMI_24M_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON HDMI_24M_CLK = HOSC
30:0	/	/	/

**3.3.6.88 0x0B10 HDMI CEC Clock Register (Default Value: 0x0000\_0000)**

Offset: 0x0B10			Register Name: HDMI_CEC_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	HDMI_CEC_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON HDMI_CEC_CLK = Clock Source.

Offset: 0x0B10			Register Name: HDMI_CEC_CLK_REG
Bit	Read/Write	Default/Hex	Description
30	R/W	0x0	PERI_GATING Gating PERIPLL Clock 0: Clock is OFF 1: Clock is ON
29:25	/	/	/
24	R/W	0x0	CLK_SRC_SEL Clock Source Select 0: CLK32K 1: HDMI_CEC_CLK32K, PLL_PERI(2X)/36621 = 32.768 kHz
23:0	/	/	/

### 3.3.6.89 0x0B1C HDMI Bus Gating Reset Register (Default Value: 0x0000\_0000)

Offset: 0x0B1C			Register Name: HDMI_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
17	R/W	0x0	HDMI_SUB_RST HDMI_SUB Reset 0: Assert 1: De-assert
16	R/W	0x0	HDMI_MAIN_RST HDMI_MAIN Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	HDMI_GATING Gating Clock For HDMI 0: Mask 1: Pass

### 3.3.6.90 0x0B24 DSI Clock Register (Default Value: 0x0000\_0000)

Offset: 0x0B24			Register Name: DSI_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	DSI_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON DSI_CLK = Clock Source/M.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: HOSC 001: PLL_PERI(1X) 010: PLL_VIDEO0(2X) 011: PLL_VIDEO1(2X) 100: PLL_AUDIO1(DIV2)
23:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 15.

### 3.3.6.91 0x0B4C DSI Bus Gating Reset Register (Default Value: 0x0000\_0000)

Offset: 0x0B4C			Register Name: DSI_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	DSI_RST DSI Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	DSI_GATING Gating Clock For DSI 0: Mask 1: Pass

**3.3.6.92 0x0B60 TCONLCD Clock Register (Default Value: 0x0000\_0000)**

Offset: 0x0B60			Register Name: TCONLCD_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	TCONLCD_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON TCONLCD_CLK = Clock Source/M/N.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: PLL_VIDEO0(1X) 001: PLL_VIDEO0(4X) 010: PLL_VIDEO1(1X) 011: PLL_VIDEO1(4X) 100: PLL_PERI(2X) 101: PLL_AUDIO1(DIV2)
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1. FACTOR_M is from 0 to 15.

**3.3.6.93 0x0B7C TCONLCD Bus Gating Reset Register (Default Value: 0x0000\_0000)**

Offset: 0x0B7C			Register Name: TCONLCD_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	TCONLCD_RST TCON LCD Reset 0: Assert 1: DE-assert
15:1	/	/	/
0	R/W	0x0	TCONLCD_GATING Gating Clock For TCON LCD 0: Mask 1: Pass

**3.3.6.94 0x0B80 TCONTV Clock Register (Default Value: 0x0000\_0000)**

Offset: 0x0B80			Register Name: TCONTV_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	TCONTV_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON TCONTV_CLK = Clock Source/M/N.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: PLL_VIDEO0(1X) 001: PLL_VIDEO0(4X) 010: PLL_VIDEO1(1X) 011: PLL_VIDEO1(4X) 100: PLL_PERI(2X) 101: PLL_AUDIO1(DIV2)
23:10	/	/	/

Offset: 0x0B80			Register Name: TCONTV_CLK_REG
Bit	Read/Write	Default/Hex	Description
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 15.

### 3.3.6.95 0x0B9C TCONTV Bus Gating Reset Register (Default Value: 0x0000\_0000)

Offset: 0x0B9C			Register Name: TCONTV_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	TCONTV_RST TCONTV Reset 0: Assert 1: DE-assert
15:1	/	/	/
0	R/W	0x0	TCONTV_GATING. Gating Clock For TCONTV 0: Mask 1: Pass

### 3.3.6.96 0x0BAC LVDS Bus Gating Reset Register (Default Value: 0x0000\_0000)

Offset: 0x0BAC			Register Name: LVDS_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/

Offset: 0x0BAC			Register Name: LVDS_BGR_REG
Bit	Read/Write	Default/Hex	Description
16	R/W	0x0	LVDS0_RST LVDS0 Reset 0: Assert 1: De-assert
15:0	/	/	/

### 3.3.6.97 0x0BF0 LEDC Clock Register (Default Value: 0x0000\_0000)

Offset: 0x0BF0			Register Name: LEDC_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	LDEC_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON LEDC_CLK = Clock Source/M/N.
30:25	/	/	/
24	R/W	0x0	CLK_SRC_SEL Clock Source Select 0: HOSC 1: PLL_PERI(1X)
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 15.

### 3.3.6.98 0x0BFC LEDC Bus Gating Reset Register (Default: 0x0000\_0000)

Offset: 0x0BFC			Register Name: LEDC_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	LEDC_RST LEDC Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	LEDC_GATING Gating Clock For LEDC 0: Mask 1: Pass

### 3.3.6.99 0x0C6C TZMA Bus Gating Reset Register (Default Value: 0x0000\_0000)

Offset: 0x0C6C			Register Name: TZMA_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	TZMA_GATING Gating Clock For TZMA 0: Mask 1: Pass

### 3.3.6.100 0x0F00 CCU Security Switch Register (Default Value: 0x0000\_0000)

Offset: 0x0F00			Register Name: CCU_SEC_SWITCH_REG
Bit	Read/Write	Default/Hex	Description
31:3	/	/	/

Offset: 0x0F00			Register Name: CCU_SEC_SWITCH_REG
Bit	Read/Write	Default/Hex	Description
2	R/W	0x0	MBUS_SEC MBUS Clock Register Security 0: Secure 1: Non-secure
1	R/W	0x0	BUS_SEC Bus Relevant Registers Security 0: Secure 1: Non-secure
0	R/W	0x0	PLL_SEC PLL Relevant Registers Security 0: Secure 1: Non-secure

### 3.3.6.101 0x0F04 PLL Lock Debug Control Register (Default Value: 0x0000\_0000)

Offset: 0x0F04			Register Name: PLL_LOCK_DBG_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	PLL_LOCK_FLAG_EN Debug Enable 0: Disable 1: Enable
30:23	/	/	/
22:20	R/W	0x0	PLL_LOCK_FLAG_SEL Debug Select 000: PLL_CPU 001: PLL_DDR 010: PLL_PERI(2X) 011: PLL_VIDEO0(4X) 100: PLL_VIDEO1(4X) 101: PLL_VE 110: PLL_AUDIO0 111: PLL_AUDIO1 Others: /
19:0	/	/	/

### 3.3.6.102 0x0F08 Frequence Detect Control Register (Default Value: 0x0000\_0020)

Offset: 0x0F08			Register Name: FRE_DET_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/WOC	0x0	<p>ERROR_FLAG Error Flag 0: Write 0 to clear 1: Error</p>
30:9	/	/	/
8:4	R/W	0x2	<p>DET_TIME Detect Time Time=1/32k*(2^RegValue) RegValue is from 0 to 16.</p>
3:2	/	/	/
1	R/W	0x0	<p>FRE_DET_IRQ_EN Frequence Detect IRQ Enable 0: Disable 1: Enable</p>
0	R/W	0x0	<p>FRE_DET_FUN_EN Frequence Detect Function Enable 0: Disable 1: Enable</p>

### 3.3.6.103 0x0F0C Frequency Up Limit Register (Default Value: 0x0000\_0000)

Offset: 0x0F0C			Register Name: FRE_UP_LIM_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<p>FRE_UP_LIM Frequency Up Limit <b>The value of the register must be an integral multiple of 32. The unit is kHz.</b></p>

**3.3.6.104 0x0F10 Frequence Down Limit Register (Default Value: 0x0000\_0000)**

Offset: 0x0F10			Register Name: FRE_DOWN_LIM_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	FRE_DOWN_LIM Frequence Down Limit <b>The value of the register must be an integral multiple of 32. The unit is kHz.</b>

**3.3.6.105 0x0F30 CCU FANOUT Clock Gate Register (Default Value: 0x0000\_0000)**

Offset: 0x0F30			Register Name: CCU_FAN_GATE_REG
Bit	Read/Write	Default/Hex	Description
31:5	/	/	/
4	R/W	0x0	CLK32K_EN Gating for CLK32K 0: Clock is OFF 1: Clock is ON
3	R/W	0x0	CLK25M_EN Gating for CLK25M 0: Clock is OFF 1: Clock is ON
2	R/W	0x0	CLK16M_EN Gating for CLK16M 0: Clock is OFF 1: Clock is ON
1	R/W	0x0	CLK12M_EN Gating for CLK12M 0: Clock is OFF 1: Clock is ON
0	R/W	0x0	CLK24M_EN Gating for CLK24M 0: Clock is OFF 1: Clock is ON

**3.3.6.106 0x0F34 CLK27M FANOUT Register (Default Value: 0x0000\_0000)**

Offset: 0x0F34			Register Name: CLK27M_FAN_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	CLK27M_EN Gating for CLK27M 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/M/N.
30:26	/	/	/
25:24	R/W	0x0	CLK27M_SCR_SEL Clock Source Select 00: PLL_VIDEO0(1X) 01: PLL_VIDEO1(1X) 10: / 11: /
23:10	/	/	/
9:8	R/W	0x0	CLK27M_DIV1 Factor N N= FACTOR_N +1. FACTOR_N is from 0 to 3.
7:5	/	/	/
4:0	R/W	0x0	CLK27M_DIV0 Factor M M= FACTOR_M +1. FACTOR_M is from 0 to 31.

**3.3.6.107 0x0F38 PCLK FANOUT Register (Default Value: 0x0000\_0000)**

Offset: 0x0F38			Register Name: PCLK_FAN_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	PCLK_DIV_EN Gating for PCLK 0: Clock is OFF 1: Clock is ON PCLK = APBO_CLK/M.
30:5	/	/	/

Offset: 0x0F38			Register Name: PCLK_FAN_REG
Bit	Read/Write	Default/Hex	Description
4:0	R/W	0x0	PCLK_DIV Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 31.

### 3.3.6.108 0x0F3C CCU FANOUT Register (Default Value: 0x0000\_0000)

Offset: 0x0F3C			Register Name: CCU_FAN_REG
Bit	Read/Write	Default/Hex	Description
31:27	/	/	/
26:24	R/W	0x0	Reserved
23	R/W	0x0	CLK_FANOUT2_EN Gating for CLK_FANOUT2 0: Clock is OFF 1: Clock is ON
22	R/W	0x0	CLK_FANOUT1_EN Gating for CLK_FANOUT1 0: Clock is OFF 1: Clock is ON
21	R/W	0x0	CLK_FANOUT0_EN Gating for CLK_FANOUT0 0: Clock is OFF 1: Clock is ON
20:18	/	/	/
17:9	R/W	0x0	Reserved
8:6	R/W	0x0	CLK_FANOUT2_SEL 000:CLK32K (From PLL_PERI(2X)) 001:CLK12M (From DCXO/2) 010:CLK16M (From PLL_PERI(2X)) 011:CLK24M (From DCXO) 100:CLK25M (From PLL_PERI(1X)) 101:CLK27M 110:PCLK CLK_FANOUT2 can be selected to output from the above seven sources.

Offset: 0x0F3C			Register Name: CCU_FAN_REG
Bit	Read/Write	Default/Hex	Description
5:3	R/W	0x0	<p>CLK_FANOUT1_SEL</p> <p>000:CLK32K (From PLL_PERI(2X))</p> <p>001:CLK12M (From DCXO/2)</p> <p>010:CLK16M (From PLL_PERI(2X))</p> <p>011:CLK24M (From DCXO)</p> <p>100:CLK25M (From PLL_PERI(1X))</p> <p>101:CLK27M</p> <p>110:PCLK</p> <p>CLK_FANOUT1 can be selected to output from the above seven sources.</p>
2:0	R/W	0x0	<p>CLK_FANOUT0_SEL</p> <p>000:CLK32K (From PLL_PERI(2X))</p> <p>001:CLK12M (From DCXO)</p> <p>010:CLK16M (From PLL_PERI(2X))</p> <p>011:CLK24M (From DCXO)</p> <p>100:CLK25M (From PLL_PERI(1X))</p> <p>101:CLK27M</p> <p>110:PCLK</p> <p>CLK_FANOUT0 can be selected to output from the above seven sources.</p>

## 3.4 BROM System

### 3.4.1 Overview

The system has several ways to boot. It has an integrated on-chip Boot ROM (BROM) that is considered the primary program-loader. On the startup process, the H133 starts to fetch the first instruction from address 0x0, where is the BROM located at.

The BROM system is divided into two parts: the firmware exchange launch (FEL) module and the Medium Boot module. FEL is responsible for writing the external data to the local NVM, and Medium Boot is responsible for loading an effective and legitimate BOOT0 from NVM and running.

The BROM system includes the following features:

- Supports CPU0 boot process
- Supports mandatory upgrade process through USB and SD card
- Supports GPIO pin and eFuse to select the boot media type
- Supports normal booting and secure booting
- Secure BROM loads only certified firmware
- Secure BROM ensures that the Secure Boot is in a trusted environment

### 3.4.2 Functional Description

#### 3.4.2.1 Selecting the Boot Medium

The BROM system supports the following boot media:

- SD card
- eMMC
- SPI NOR FLASH
- SPI NAND FLASH

There are two ways to select the boot medium: GPIO Pin Select and eFuse Select. On startup, the BROM will read the state of BOOT\_MODE, and decide whether the GPIO or eFuse to select the type of boot medium based on the state of BOOT\_MODE.

The following table shows BOOT\_MODE Setting.

**Table 3-7 BOOT\_MODE Setting**

BOOT_MODE	Boot_Select Type
0	GPIO boot select, indicates that the boot medium is decided by the value of the GPIO pin.
1	eFuse boot select, indicates that the boot medium is decided by the value of the eFuse type.

#### GPIO Boot Select

If the state of the BOOT\_MODE is 0, the boot medium is decided by the value of the GPIO pin. The following table shows the boot medium priority. The boot medium priority describes the possibility that each medium to be selected as the boot medium. The BROM reads the boot0 of the medium with the highest priority first. If the medium does not exist or has any problems, the BROM will try the next medium. Otherwise, the medium will be selected as the boot medium.

**Table 3-8 GPIO Boot Select Configuration**

Pin_Boot_Select[1:0]	Boot Medium Priority
00	SPI NOR->SPI NAND
01	SMHCO->SPI NOR->other media
10	SMHCO->SPI NAND->other media
11	SMHCO->EMMC2_BOOT->EMMC2_USR->other media



The status of the GPIO boot select pin can be read by the bit[12:11] of the system configuration module (register: 0x03000024).

---

#### eFuse Boot Select

If the state of the BOOT\_MODE is 1, the boot medium is decided by the value of eFuse\_Boot\_Select\_Cfg. The eFuse\_Boot\_Select\_Cfg is divided into 4 groups and each group is 3-bit. The following table shows the groups of eFUSE\_Boot\_Select.

---



The status of the efuse boot select pin can be read by the bit[27:16] of the SID module (register: 0x03006210).

---

**Table 3-9 Groups of eFuse\_Boot\_Select**

eFuse_Boot_Select_Cfg[11:0]	Group
eFuse_Boot_Select_Cfg[2:0]	eFuse_Boot_Select_1
eFuse_Boot_Select_Cfg[5:3]	eFuse_Boot_Select_2
eFuse_Boot_Select_Cfg[8:6]	eFuse_Boot_Select_3
eFuse_Boot_Select_Cfg[11:9]	eFuse_Boot_Select_4

The four groups take effect with the following priority:

**eFuse\_Boot\_Select\_1 -> eFuse\_Boot\_Select\_2 -> eFuse\_Boot\_Select\_3 -> eFuse\_Boot\_Select\_4**

For example, eFuse\_Boot\_Select\_2 will not take effect unless eFuse\_Boot\_Select\_1 is set as 0x111, eFuse\_Boot\_Select\_3 will not take effect unless eFuse\_Boot\_Select\_2 is set as 0x111, and so on.

The following table shows the boot medium priority for the different values of eFuse\_Boot\_Select\_n, where n = [4:1]. The eFuse\_Boot\_Select\_1 to eFuse\_Boot\_Select\_3 are the same setting. But for eFuse\_Boot\_Select\_4, if its value is 0x111, the BROM will select the boot medium in the Try mode. The BROM in the Try mode follows the order below to select the boot medium:

**SMHC0 -> SPI NOR -> SPI NAND -> SMHC2**

**Table 3-10 eFuse Boot Select Setting**

eFuse_Boot_Select_n	Boot Medium Priority
000	Select the boot medium in Try mode.
001	Reserved
010	SHMC2
011	SPI NOR
100	SPI NAND
101	Reserved
110	Reserved
111	When n is 1 to 3: The boot medium is decided by the value of <b>eFuse_Boot_Select_(n + 1)</b> . When n is 4: Select the boot medium in Try mode.

### 3.4.2.2 Selecting the Boot Mode

For SoCs that have implemented and enabled the ARM TrustZone technology, there are two boot modes: Normal BROM Mode and Secure BROM Mode. Secure BROM Mode is designed to protect against the potential threat that attackers modify the code or data areas in programmable memory.

On startup, the BROM will select the boot mode according to the value of the Secure Enable bit. If the value of Secure Enable bit is 0, the system will boot in Normal BROM Mode. Otherwise, it will boot in Secure BROM Mode.



The System on Chip (SoC) supports the ARM TrustZone technology. If the Secure Enable Bit is enabled, the BROM will be safely booted based on this ARM TrustZone technology.

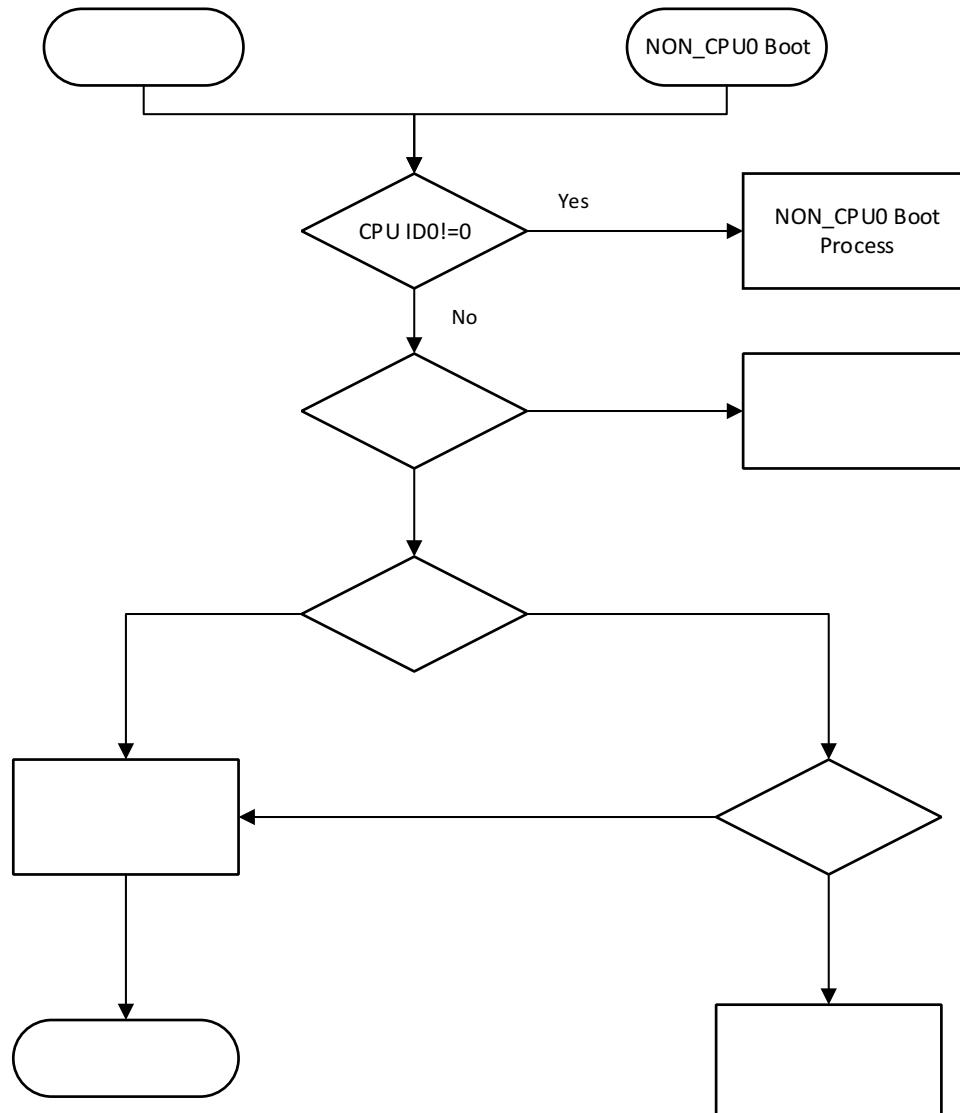
---

#### Normal BROM Mode

In Normal BROM Mode, the system boot starts from CPU0, and then the BROM will read the state of the FEL pin. If the FEL pin is high, the system will jump to the fast boot process. If it is low, the system will jump to the mandatory upgrade process.

The following figure shows the boot process in Normal BROM Mode.

**Figure 3-6 Boot Process in Normal BROM Mode**



### Secure BROM Mode

The Secure BROM Mode has the following features:

- Supports X509 certificate

The certificate is used to check whether the Security Boot software is modified or replaced. Before running the Security Boot software, the system checks the integrity of the certificate to make sure the software has not been modified or replaced.

- Supports cryptographic algorithms
  - SHA-256
  - RSA-2048

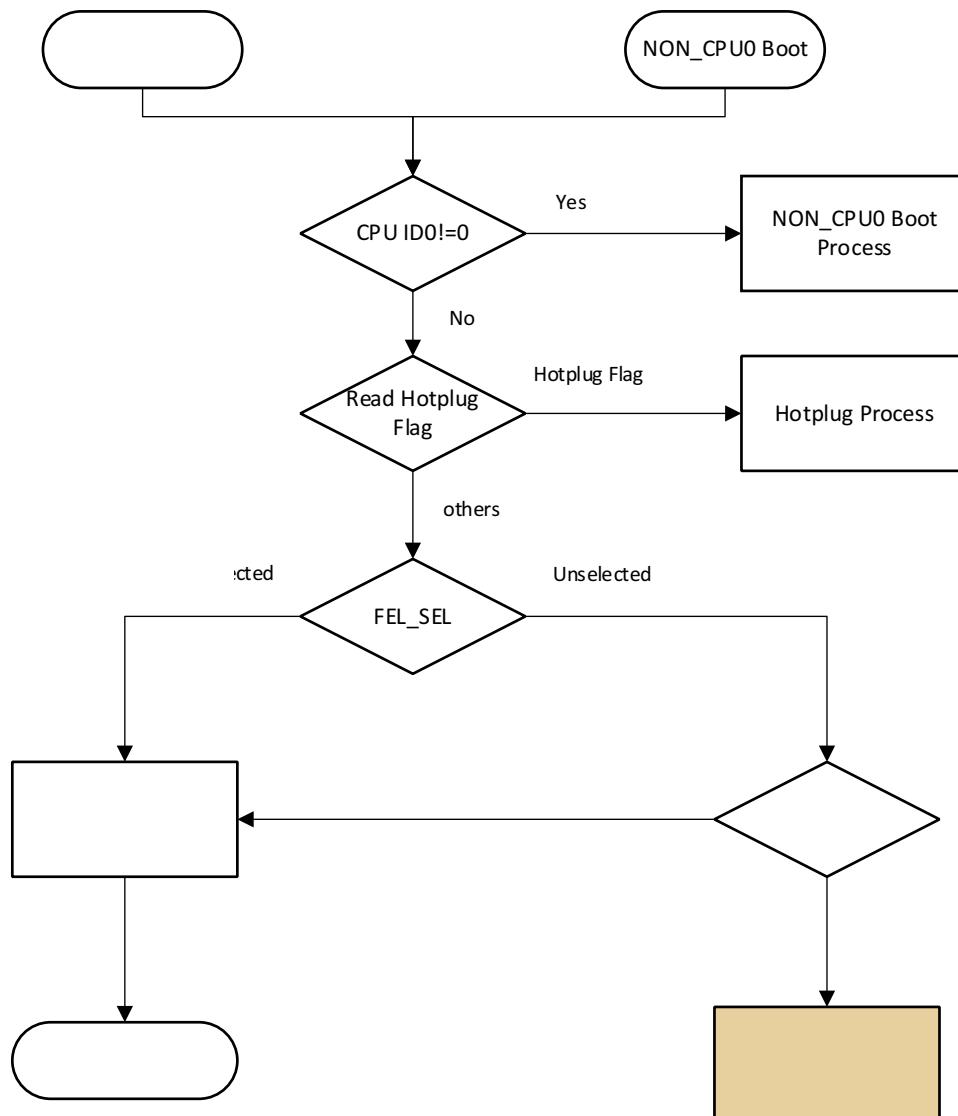
The system uses the Crypto Engine (CE) hardware module to accelerate the speed of encryption and decryption. The use of standard cryptography ensures the reliability of the firmware images. The reliable firmware image ensures that the system security state can be as expected.

- Supports OTP/eFuse

The process of selecting the boot medium in Secure BROM Mode is the same as that in Normal BROM Mode.

In Secure BROM Mode, after the boot medium is selected, the system additionally runs the Security Boot software to authenticate the Sboot bin file.

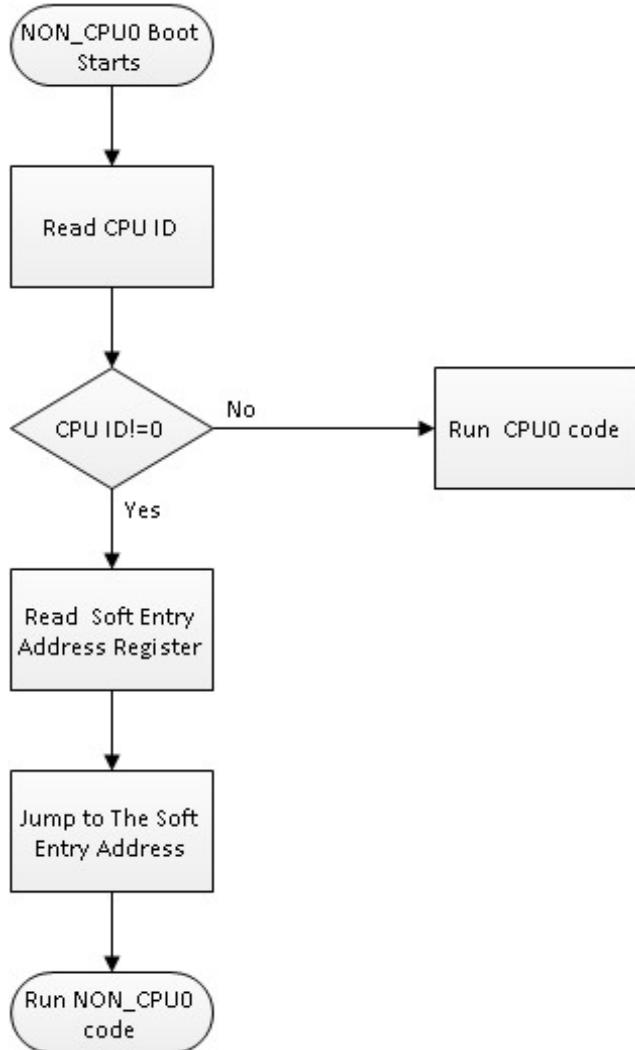
**Figure 3-7 Boot Process in Secure BROM Mode**



### 3.4.2.3 NON\_CPU0 Boot Process

If CPU ID is greater than 0, the system boot from NON\_CPU0, the BROM will read the Soft Entry Address Register, then jump to the Soft Entry Address, and run the NON\_CPU0 boot code. The following figure shows the NON\_CPU0 Boot Process.

**Figure 3-8 NON\_CPU0 Boot Process Diagram**



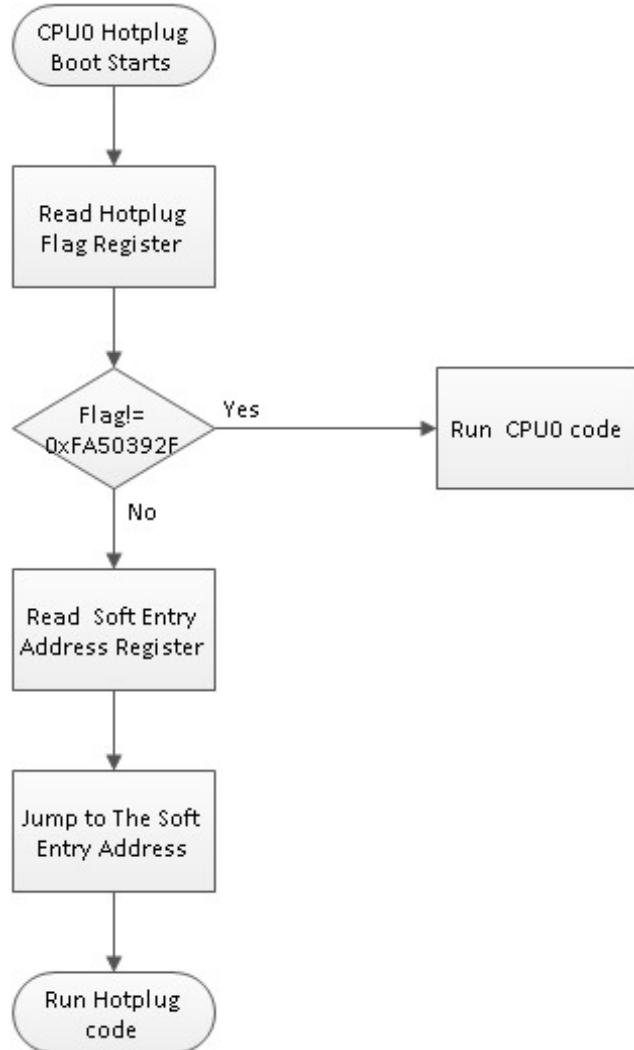
The Soft Entry Address Register of CPU0 is 0x0700005C4.

The Soft Entry Address Register of CPU1 is 0x0700005C8.

### 3.4.2.4 CPU0 Hotplug Process

The Hotplug Flag determines whether the system will do Hotplug boot, if the CPU Hotplug Flag value is equal to 0xFA50392F, then read the Soft Entry Register and the system will jump to the Soft Entry Address. The following figure shows the CPU0 Hotplug Process.

Figure 3-9 CPU0 Hotplug Process Diagram



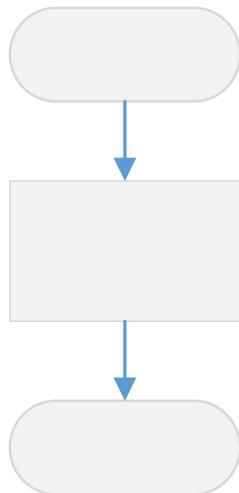
**NOTE**

- The Hotplug Flag Register is 0x070005C0.
- The Soft Entry Address Register is 0x070005C4.

### 3.4.2.5 Mandatory Upgrade Process

If the FEL pin is detected to pull low, the system will jump to the mandatory upgrade process. The following figure shows the mandatory upgrade process.

**Figure 3-10 Mandatory Upgrade Process**



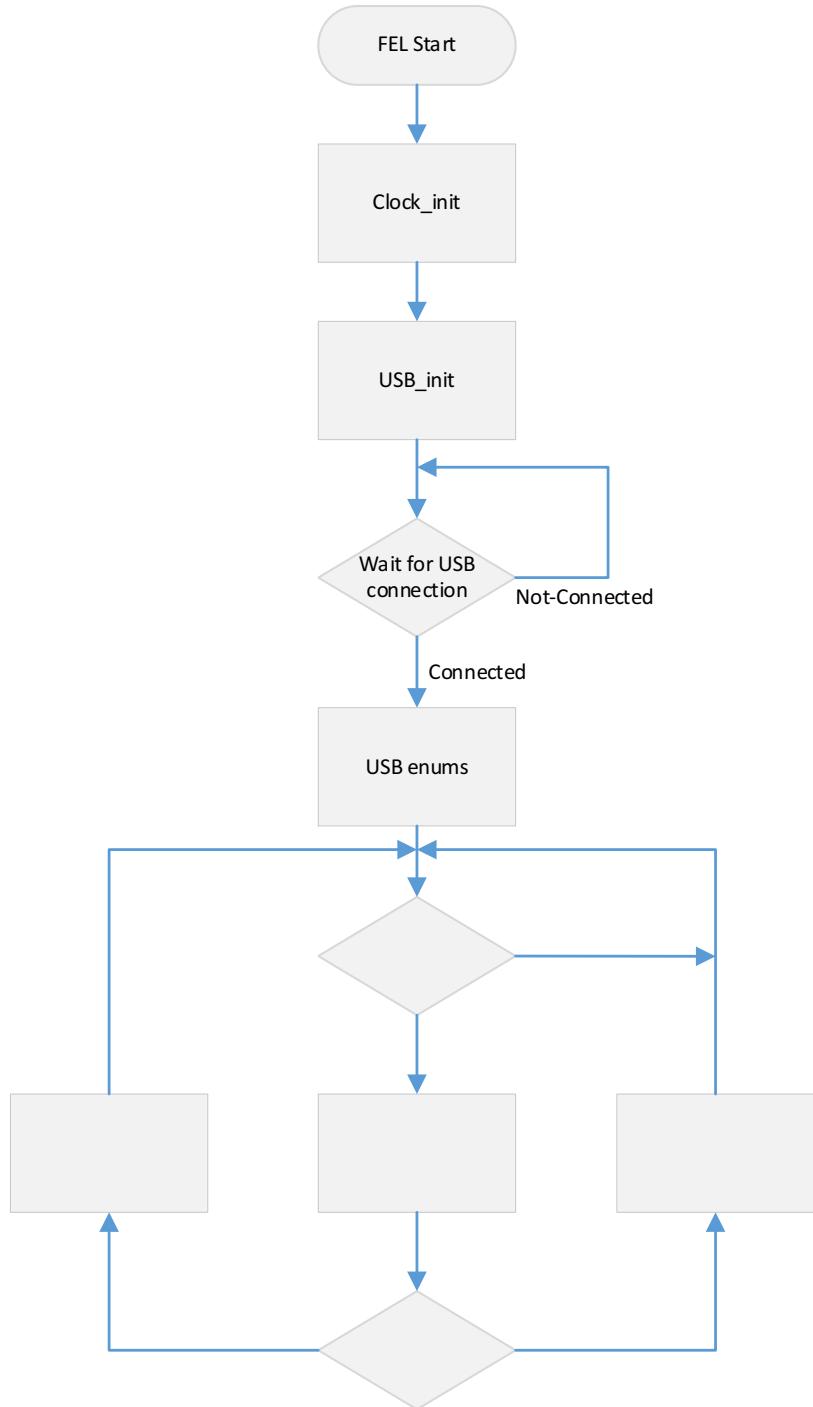
#### NOTE

- The FEL address of the Normal BROM is 0x20.
- The FEL address of the Secure BROM is 0x64.
- The status of the FEL pin is the bit[8] of the SYSCTRL\_REG (0x03000024).

### 3.4.2.6 FEL Process

When the system chooses to enter the Mandatory Upgrade Process, the system will jump to the FEL process. The following figure shows the FEL upgrade process.

**Figure 3-11 USB FEL Process**



### **3.4.2.7 Fast Boot Process**

If the value of the Fast Boot register (0x07090120) in RTC module is not zero, the system will enter the Fast Boot Process. The following table shows the boot medium priority for different values of the Fast Boot register.

**Table 3-11 Fast Boot Select Setting**

Reg_bit[31:28]	Boot Medium Priority
1	Try process
2	SMHCO->EMMC2_USER->EMMC2_BOOT->Other media
3	SMHCO->SPI NOR->Other media
4	SMHCO->SPI NAND->Other media
5	EMMC2_BOOT->EMMC2_USER->Other media
6	SMHCO->Try process
7	SMHCO->Try process

 **NOTE**

- The bit[28:0] of Fast Boot register is used to record the media information.
- The try process is SMHCO->SPI NOR->SPI NAND->EMMC2\_USER->EMMC2\_BOOT.

## 3.5 System Configuration

### 3.5.1 Overview

The system configuration module is used to configure parameters for system domain, such as SRAM, CPU, PLL, BROM, and so on.

### 3.5.2 Register List

Module Name	Base Address
SYS_CFG	0x03000000

Register Name	Offset	Description
VER_REG	0x0024	Version Register
EMAC_EPHY_CLK_REG0	0x0030	EMAC-EPHY Clock Register 0
SYS_LDO_CTRL_REG	0x0150	System LDO Control Register
RESCAL_CTRL_REG	0x0160	Resistor Calibration Control Register
RES240_CTRL_REG	0x0168	240ohms Resistor Manual Control Register
RESCAL_STATUS_REG	0x016C	Resistor Calibration Status Register

### 3.5.3 Register Description

#### 3.5.3.1 0x0024 Version Register (Default Value: UDF)

Offset: 0x0024			Register Name: VER_REG
Bit	Read/Write	Default/Hex	Description
31:13	/	/	/
12:11	R	UDF	BOOT_SEL_PAD_STA The value of this bit decides the priority order for each medium type to be selected as the boot media.
10:9	/	/	/

Offset: 0x0024			Register Name: VER_REG
Bit	Read/Write	Default/Hex	Description
8	R	UDF	FEL_SEL_PAD_STA Fel_Select_Pin_Status 0: Run_FEL 1: Try Media Boot
7:0	/	/	/

### 3.5.3.2 0x0030 EMAC-EPHY Clock Register 0 (Default Value: 0x0005\_8000)

Offset: 0x0030			Register Name: EMAC_EPHY_CLK_REG0
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	BPS_EFUSE
27	R/W	0x0	XMII_SEL 0: Internal SMI and MII 1: External SMI and MII
26:25	R/W	0x0	EPHY_MODE Operation Mode Selection 00: Normal Mode 01: Simulation Mode 10: AFE Test Mode 11: Reserved
24:20	R/W	0x0	PHY_ADDR PHY Address
19	R/W	0x0	Reserved
18	R/W	0x1	CLK_SEL 0: 25 MHz 1: 24 MHz
17	R/W	0x0	LED_POL 0: High active 1: Low active
16	R/W	0x1	SHUTDOWN 0: Power up 1: Shut down

Offset: 0x0030			Register Name: EMAC_EPHY_CLK_REG0
Bit	Read/Write	Default/Hex	Description
15	R/W	0x1	PHY_SELECT 0: External PHY 1: Internal PHY
14	/	/	/
13	R/W	0x0	RMII_EN 0: Disable RMII Module 1: Enable RMII Module This bit is prior to bit[2]. When this bit is asserted, the MII and RGMII interfaces will be both disabled.
12:10	R/W	0x0	ETXDC Configure EMAC Transmit Clock Delay Chain
9:5	R/W	0x0	ERXDC Configure EMAC Receive Clock Delay Chain
4	R/W	0x0	ERXIE Enable EMAC Receive Clock Invertor 0: Disabled 1: Enabled
3	R/W	0x0	ETXIE Enable EMAC Transmit Clock Invertor 0: Disabled 1: Enabled
2	R/W	0x0	EPIT EMAC PHY Interface Type 0: MII 1: RGMII
1:0	R/W	0x0	ETCS EMAC Transmit Clock Source 00: Transmit clock source for MII 01: External transmit clock source for GMII and RGMII 10: Internal transmit clock source for GMII and RGMII 11: Reserved

**3.5.3.3 0x0150 System LDO Control Register (Default Value: 0x0000\_0EOF)**

Offset: 0x0150			Register Name: SYS_LDO_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:24	R/W	0x0	SPARE Reserved spare register
23:22	/	/	/
21	R/W	0x0	Reserved
20	R/W	0x0	Reserved
19:18	/	/	/
17	R/W	0x0	Reserved
16	R/W	0x0	Reserved
15:8	R/W	0xE	LDOB_TRIM LDOB Trimming. Adjust LDOB output, only the low 6-bit is used. 000000:1.167 000001:1.18 000010:1.193 000011:1.207 000100:1.22 000101:1.233 000110:1.247 000111:1.260 001000:1.273 001001:1.287 001010:1.3 001011:1.313 001100:1.327 001101:1.340 001110:1.353 (default) 001111:1.367 010000:1.38 010001:1.393 010010:1.407 010011:1.42 010100:1.433 010101:1.447 010110:1.46

Offset: 0x0150			Register Name: SYS_LDO_CTRL_REG
Bit	Read/Write	Default/Hex	Description
			010111:1.473
			011000:1.487
			011001:1.5
			011010:1.513
			011011:1.527
			011100:1.54
			011101:1.553
			011110:1.567
			011111:1.58
			100000:1.593
			100001:1.607
			100010:1.627
			100011:1.64
			100100:1.653
			100101:1.667
			100110:1.680
			100111:1.693
			101000:1.707
			101001:1.720
			101010:1.733
			101011:1.747
			101100:1.76
			101101:1.773
			101110:1.787
			101111:1.8
			110000:1.813
			110001:1.827
			110010:1.84
			110011:1.853
			110100:1.867
			110101:1.88
			110110:1.893
			110111:1.907
			111000:1.92
			111001:1.933
			111010:1.947
			111011:1.96

Offset: 0x0150			Register Name: SYS_LDO_CTRL_REG
Bit	Read/Write	Default/Hex	Description
			111100:1.973 111101:1.987 111110:2 111111:2.013
7:0	R/W	0xF	LDOA_TIM. LDOA Trimming Adjust LDOA output, only the low 5-bit is used. 00000:1.593 00001:1.607 00010:1.627 00011:1.64 00100:1.653 00101:1.667 00110:1.680 00111:1.693 01000:1.707 01001:1.720 01010:1.733 01011:1.747 01100:1.76 01101:1.773 01110:1.787 01111:1.8 (default) 10000:1.813 10001:1.827 10010:1.84 10011:1.853 10100:1.867 10101:1.88 10110:1.893 10111:1.907 11000:1.92 11001:1.933 11010:1.947 11011:1.96 11100:1.973 11101:1.987

Offset: 0x0150			Register Name: SYS_LDO_CTRL_REG
Bit	Read/Write	Default/Hex	Description
			11110:2
			11111:2.013

### 3.5.3.4 0x0160 Resistor Calibration Control Register (Default Value: 0x0033\_0003)

Offset: 0x0160			Register Name: RESCAL_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:25	/	/	/
24	R/W	0x0	Reserved
23:22	/	/	/
21:16	R/W	0x33	Reserved
15:9	/	/	/
8	R/W	0x0	DDR_RES240_Trimming_SEL 240ohms Resistor Trimming Source Select 0: Trimming value from RESCAL 1: Trimming value from RES240_TRIM
7:3	/	/	/
2	R/W	0x0	RESCAL_MODE RESCAL Calibration Mode Select 0: Auto Calibration 1: Reserved
1	R/W	0x1	CAL_ANA_EN. Calibration Circuits Analog Enable 0: Disable 1: Enable
0	R/W	0x1	CAL_EN Auto Calibration Enable 0: Disable 1: Enable

**3.5.3.5 0x0168 240ohms Resistor Manual Control Register (Default Value: 0x0000\_0033)**

Offset: 0x0168			Register Name: RES240_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:6	/	/	/
5:0	R/W	0x33	DDR_RES240_TRIM 240ohms Resistor trimming bit

**3.5.3.6 0x016C Resistor Calibration Status Register (Default Value: 0x0000\_0000)**

Offset: 0x016C			Register Name: RESCAL_STATUS_REG
Bit	Read/Write	Default/Hex	Description
31:9	/	/	/
8	RO	0x0	COUT Calibration Circuits Analog Compare Output
7:6	/	/	/
5:0	RO	0x0	RES_CAL_DO RESCAL Calibration Results Output

## 3.6 Timer

### 3.6.1 Overview

The timer module implements the timing and counting functions. The timer module includes timer0, timer1, watchdog and audio video synchronization (AVS).

The timer0 and timer1 are completely consistent. The main features for timer0 and timer1 are as follows:

- Alternative count clock: LOSC or OSC24M. The LOSC can be either the internal or external low-frequency clock, and the external one has more accuracy.
- Supports 8 prescale factors
- Programmable 32-bit down timer
- Supports two timing modes: periodic mode and single counting mode
- Generates an interrupt when the count is decreased to 0

The watchdog is used to transmit a reset signal to reset the entire system when an exception occurs in the system. The main features for the watchdog are as follows:

- Single clock source: OSC24M/750.
- Supports 12 initial values
- Supports the generation of timeout interrupts
- Supports the generation of reset signals
- Supports Watchdog restart

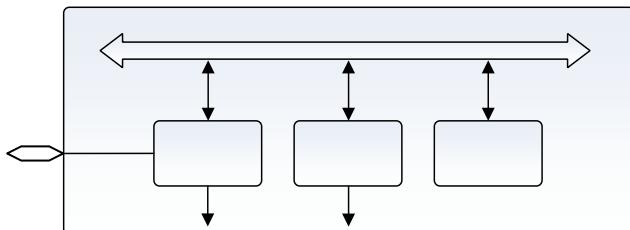
The AVS is used to synchronize the audio and video. The AVS module includes AVS0 and AVS1, which are completely consistent. The main features for the AVS are as follows:

- Single clock source: OSC24M
- Programmable 33-bit up timer
- Supports updating the initial value anytime
- 12-bit frequency divider factor
- Supports Pause/Start function

### 3.6.2 Block Diagram

The following figure shows the functional block diagram of the timer module.

**Figure 3-12 Timer Block Diagram**



The watchdog, timer (including timer0 and timer1), and AVS are all mounted at the APB0 bus. The system configures the parameters of their configure registers via APB0 bus.

The timer and watchdog are both down counters and support generating interrupts after the counting value reaches 0.

For watchdog, the system is responsible for configuring the interval value. If the system fails to restart the watchdog regularly because of some exceptional situations, such as the bus hang, the watchdog will send out a Watchdog Reset External signal to reset the system. And the signal will be transmitted to the Reset pad to reset the PMIC.

### 3.6.3 Functional Description

#### 3.6.3.1 Timer

The timer (including timer0 and timer1) is a 32-bit down counter. The counter value is decremented by 1 on each rising edge of the timer clock.

The following figure shows the block diagram for the timer.

**Figure 3-13 Block Diagram for the Timer**



The clock source for the timer can be either OSC24M or LOSC. For LOSC, it can be either the internal or external low-frequency clock. The external one has more accuracy.

Each timer has a prescale that divides the working clock frequency by 1, 2, 4, 8, 16, 32, 64, or 128. And each timer can generate independent interrupts.

### Timing Modes

The timer has two timing modes: the single counting mode and periodic mode. You can configure the timing mode via the bit[7] of [TMRn\\_CTRL\\_REG](#) (n = 0 or 1). The value 0 is for the period mode and value 1 is for the single counting mode.

- Single Counting Mode

In the single counting mode, the timer starts counting from the interval value and generates an interrupt after the counter decreases to 0, and then stops counting. It starts to count again only when a new interval value is loaded.

- Periodic Mode

In the periodic mode, the timer restarts another round of counting after generating the interrupt. It reloads data from the [TMRn\\_INTV\\_VALUE\\_REG](#) and then continues to count.

### Formula for Calculating the Timer Time

The following formula describes the relationship among timer parameters.

$$T_{\text{timer}} = \frac{\text{TMRn\_INTV\_VALUE\_REG} - \text{TMRn\_CUR\_VALUE\_REG}}{\text{TMRn\_CLK\_SRC}} \times \text{TMRn\_CLK\_PRES}$$

Where,

The parameter n is either 0 or 1;

$T_{\text{timer}}$  is the remaining time of the timer;

$\text{TMRn\_INTV\_VALUE\_REG}$  is the interval value of the timer;

$\text{TMRn\_CUR\_VALUE\_REG}$  is the current value of the timer;

$\text{TMRn\_CLK\_SRC}$  is the frequency of the timer clock source;

$\text{TMRn\_CLK\_PRES}$  is the prescale ratio of the timer clock.

### Initializing the Timer

Follow the steps below to initialize the timer:

1. Configure the timer parameters clock source, prescale factor, and timing mode by writing [TMRn\\_CTRL\\_REG](#). There is no sequence requirement of configuring the parameters.
2. Write the interval value.
  - a) Write [TMRn\\_INTV\\_VALUE\\_REG](#) to configure the interval value for the timer.

- b) Write bit[1] of [TMRn\\_CTRL\\_REG](#) to load the interval value to the timer. The value of the bit will be cleared automatically after loading the interval value.
3. Write bit[0] of [TMRn\\_CTRL\\_REG](#) to start the timer. To get the current value of the timer, read [TMRn\\_CUR\\_VALUE\\_REG](#).

### Processing the Interrupt

Follow the steps below to process the interrupt:

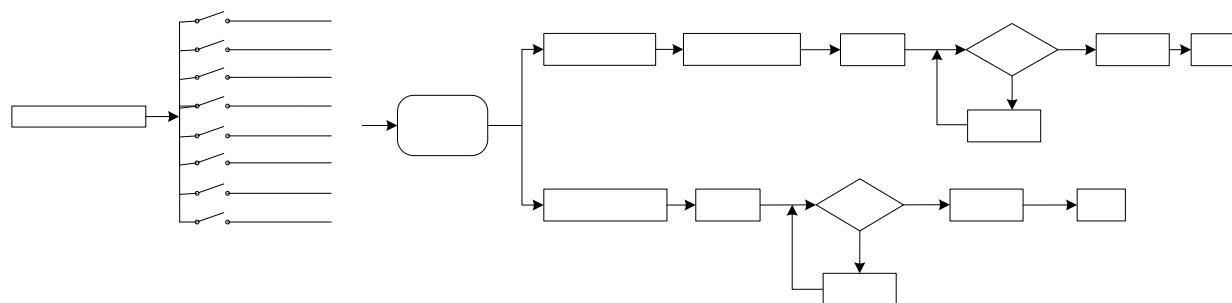
1. Enable interrupts for the timer: write the enable bit of the corresponding interrupt in [TMR\\_IRQ\\_EN\\_REG](#) for the timer. The timer will generate an interrupt everytime the count value reaches 0.
2. After entering the interrupt process, write the pending bit of the corresponding interrupt in [TMR\\_IRQ\\_STA\\_REG](#) to clear the interrupt pending, and execute the process of waiting for the interrupt.
3. Resume the interrupt and continue to execute the interrupted process.

#### 3.6.3.2 Watchdog

The watchdog is a 32-bit down counter. The counter value is decremented by 1 on each rising edge of the count clock.

The following figure shows the block diagram for the watchdog.

**Figure 3-14 Block Diagram for the Watchdog**



The clock source of the watchdog is OSC24M/750. There are 12 configurable initial count values.

### Operating Modes

The watchdog has two operating modes: the interrupt mode and reset mode.

- In the interrupt mode, when the counter value reaches 0 and WDOG\_IRQ\_EN\_REG is enabled, the watchdog generates an interrupt.
- In the reset mode, when the counter value reaches 0, the watchdog generates a reset signal to reset the entire system.

You can configure the operating mode for the watchdog via the bit[1:0] of the WDOG\_CFG\_REG. The value 0x2 is for the interrupt mode and the value 0x1 is for the reset mode.

Both the interrupt mode and reset mode support Watchdog Restart. You can make the watchdog to count from the initial value at any time by configuring the WDOG\_CTRL\_REG: write 0xA57 to bit[12:1], then write 1 to bit[0].

### Initializing the Watchdog

Follow the steps below to initialize the watchdog:

1. Write the bit[1:0] of [WDOG\\_CFG\\_REG](#) to configure the watchdog operating mode so that the watchdog can generate interrupts or output reset signals.
2. Write the bit[7:4] of [WDOG\\_MODE\\_REG](#) to configure the initial count value.
3. Write the bit[0] of [WDOG\\_MODE\\_REG](#) to enable the watchdog.

### Processing the Interrupt

In the interrupt mode, the watchdog is used as a counter. It generates an interrupt everytime the count value reaches 0.

Follow the steps below to process the interrupt:

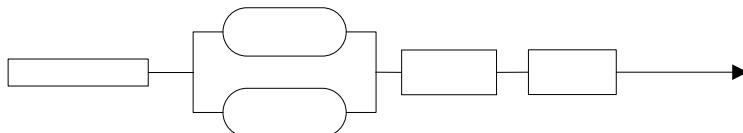
1. Write the enable bit of [WDOG\\_IRQ\\_EN\\_REG](#) to enable the interrupt.
2. After entering the interrupt process, write the pending bit of [WDOG\\_IRQ\\_STA\\_REG](#) to clear the interrupt pending and execute the process of waiting for the interrupt.
3. Resume the interrupt and continue to execute the interrupted process.

#### 3.6.3.3 AVS

The AVS is a 33-bit up counter. The counter value is increased by 1 on each rising edge of the count clock. There is a clock gate in [CCU](#) module to control the output of the AVS counter. To operate the AVS, open the clock gate first.

The following figure shows the block diagram for the AVS.

**Figure 3-15 Block Diagram for the AVS**



The clock source of the AVS is OSC24M. There is a 12-bit division factor for each AVS, N0 for AVS0 and N1 for AVS1. When the timer increases from 0 to N0 or N1, the AVS counter adds 1. When the counter reaches 33-bit upper limit, the AVS will start to count from the initial value again.

The AVS supports changing the initial value and division factor at anytime. And the AVS supports restarting from the initial value or pausing at anytime.

### Starting or Pausing the AVS

Follow the steps below to

1. Write [AVS\\_CNT\\_DIV\\_REG](#) to configure the division factor.
2. Write [AVS\\_CNTn\\_REG](#) ( $n = 0$  or  $1$ ) to configure the initial value.
3. Write [AVS\\_CNT\\_CTL\\_REG](#) to enable the AVS. You can pause the AVS at any time.

## 3.6.4 Programming Guidelines

### 3.6.4.1 Configuring the Timer

The following example shows how to make a one-millisecond delay with the clock source selected as OSC24M, the operating mode sets as single counting mode, and the pre-scale sets as 2.

```
writel(0x2EE0, TMR_0_INTV);           //Set the interval value  
writel(0x94, TMR_0_CTRL);            //Select Single mode, 24 MHz clock source, 2 pre-scale  
writel(readl(TMR_0_CTRL)|(1<<1), TMR_0_CTRL); //Set the Reload bit  
while((readl(TMR_0_CTRL)>>1)&1);        //Wait until the Reload bit turns to 0  
writel(readl(TMR_0_CTRL)|(1<<0), TMR_0_CTRL); //Enable Timer0
```

### 3.6.4.2 Resetting the Watchdog

The following example shows how to make the watchdog to generate a reset signal to the whole system after 1 second. The clock source for the watchdog is OSC24M/750.

```
writel(0x1, WDOG_CONFIG);           //Set the operating mode as the reset mode.  
writel(0x10, WDOG_MODE);          //Set the interval value as 1 s.  
writel(readl(WDOG_MODE)|(1<<0), WDOG_MODE); //Enable the Watchdog.
```

### 3.6.4.3 Restarting the Watchdog

The following example shows how to restart the watchdog. In this example, the clock source is OSC24M/750, the interval value is 1 second, and the watchdog operating mode is the reset mode.

If the execution time of “other codes” is shorter than 1 second, the watchdog will restart from the interval value before it count to zero and generates the reset signal. Otherwise, the watchdog will reset the whole system before the code “writel(readl(WDOG\_CTRL)|(0xA57<<1)|(1<<0), WDOG\_CTRL)” is executed.

```
writel(0x1, WDOG_CONFIG);           //To whole system  
writel(0x10, WDOG_MODE);          //Interval Value set 1s  
writel(readl(WDOG_MODE)|(1<<0), WDOG_MODE); //Enable Watchdog  
----other codes---  
writel(readl(WDOG_CTRL)|(0xA57<<1)|(1<<0), WDOG_CTRL); //Write 0xA57 at Key Field and Restart Watchdog
```

## 3.6.5 Register List

Module Name	Base Address
Timer	0x02050000

Register Name	Offset	Description
TMR_IRQ_EN_REG	0x0000	Timer IRQ Enable Register
TMR_IRQ_STA_REG	0x0004	Timer Status Register
TMRO_CTRL_REG	0x0010	Timer0 Control Register
TMRO_INTV_VALUE_REG	0x0014	Timer0 Interval Value Register
TMRO_CUR_VALUE_REG	0x0018	Timer0 Current Value Register

Register Name	Offset	Description
TMR1_CTRL_REG	0x0020	Timer1 Control Register
TMR1_INTV_VALUE_REG	0x0024	Timer1 Interval Value Register
TMR1_CUR_VALUE_REG	0x0028	Timer1 Current Value Register
WDOG_IRQ_EN_REG	0x00A0	Watchdog IRQ Enable Register
WDOG_IRQ_STA_REG	0x00A4	Watchdog Status Register
WDOG_SOFT_RST_REG	0x00A8	Watchdog Software Reset Register
WDOG_CTRL_REG	0x00B0	Watchdog Control Register
WDOG_CFG_REG	0x00B4	Watchdog Configuration Register
WDOG_MODE_REG	0x00B8	Watchdog Mode Register
WDOG_OUTPUT_CFG_REG	0x00BC	Watchdog Output Configuration Register
AVS_CNT_CTL_REG	0x00C0	AVS Control Register
AVS_CNT0_REG	0x00C4	AVS Counter 0 Register
AVS_CNT1_REG	0x00C8	AVS Counter 1 Register
AVS_CNT_DIV_REG	0x00CC	AVS Divisor Register

### 3.6.6 Register Description

#### 3.6.6.1 0x0000 Timer IRQ Enable Register (Default Value: 0x0000\_0000)

Offset: 0x0000			Register Name: TMR_IRQ_EN_REG
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1	R/W	0x0	TMR1_IRQ_EN Timer1 Interrupt Enable 0: Disabled 1: Enabled
0	R/W	0x0	TMRO_IRQ_EN Timer0 Interrupt Enable 0: Disabled 1: Enabled

### 3.6.6.2 0x0004 Timer IRQ Status Register (Default Value: 0x0000\_0000)

Offset: 0x0004			Register Name: TMR_IRQ_STA_REG
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1	R/W1C	0x0	<p>TMR1_IRQ_PEND The IRQ pending bit for Timer1 0: No effect 1: Pending, indicates that the interval value of the timer 1 is reached. Write 1 to clear the pending status.</p>
0	R/W1C	0x0	<p>TMR0_IRQ_PEND The IRQ pending bit for Timer0 0: No effect 1: Pending, indicates that the interval value of the timer 0 is reached. Write 1 to clear the pending status.</p>

### 3.6.6.3 0x0010 Timer0 Control Register (Default Value: 0x0000\_0004)

Offset: 0x0010			Register Name: TMRO_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R/W	0x0	<p>TMRO_MODE Select the timing mode for timer0 0: Periodic mode. When the interval value of the timer 0 is reached, the timer will restart another round of counting automatically. 1: Single counting mode. When the interval value of the timer 0 is reached, the timer will stop counting.</p>

Offset: 0x0010			Register Name: TMRO_CTRL_REG
Bit	Read/Write	Default/Hex	Description
6:4	R/W	0x0	<p>TMRO_CLK_PRES Select the pre-scale of timer0 clock source 000: /1 001: /2 010: /4 011: /8 100: /16 101: /32 110: /64 111: /128</p>
3:2	R/W	0x1	<p>TMRO_CLK_SRC Select the clock source for timer0 00: LOSC 01: OSC24M 10: / 11: /</p>
1	R/W	0x0	<p>TMRO_RELOAD Timer0 Reload 0: No effect 1: Reload the Interval value for timer0 After the bit is set, it can not be written again before it is cleared automatically.</p>
0	R/W	0x0	<p>TMRO_EN Timer0 Enable 0: Stop/Pause 1: Start By setting the bit to 1, the timer will be started. It reloads the interval value register and then counts from the interval value to 0. By setting the bit to 0 before the timer counts to 0, the timer will be paused. The bit will be locked to 0 for at least 2 cycles. Within the 2 cycles, you cannot set the bit to 1 to restart the timer. The timer supports updating the interval value in the pause state. To start to down-count from the updated interval value, set both the reload bit and enable bit to 1. Additionally, in the single counting mode, after the count value reaches 0, the system will automatically change the bit to 0 to stop the timer.</p>

### 3.6.6.4 0x0014 Timer0 Interval Value Register (Default Value: 0x0000\_0000)

Offset: 0x0014			Register Name: TMR0_INTV_VALUE_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	TMR0_INTV_VALUE Timer0 Interval Value



Take the system clock and timer clock source into consideration when setting the interval value.

### 3.6.6.5 0x0018 Timer0 Current Value Register (Default Value: 0x0000\_0000)

Offset: 0x0018			Register Name: TMR0_CUR_VALUE_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	TMR0_CUR_VALUE Timer0 Current Value Timer0 current value is a 32-bit down-counter (from interval value to 0).

### 3.6.6.6 0x0020 Timer1 Control Register (Default Value: 0x0000\_0004)

Offset: 0x0020			Register Name: TMR1_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R/W	0x0	TMR1_MODE Select the timing mode for timer1 0: Periodic mode. When the interval value of the timer 1 is reached, the timer will restart another round of counting automatically. 1: Single counting mode. When the interval value of the timer 1 is reached, the timer will stop counting.

Offset: 0x0020			Register Name: TMR1_CTRL_REG
Bit	Read/Write	Default/Hex	Description
6:4	R/W	0x0	<p>TMR1_CLK_PRES Select the pre-scale of timer1 clock source</p> <p>000: /1 001: /2 010: /4 011: /8 100: /16 101: /32 110: /64 111: /128</p>
3:2	R/W	0x1	<p>TMR1_CLK_SRC Select the pre-scale of timer1 clock source</p> <p>00: LOSC 01: OSC24M 10: / 11: /</p>
1	R/W	0x0	<p>TMR1_RELOAD Timer1 Reload</p> <p>0: No effect 1: Reload the interval value for timer1</p> <p>After the bit is set, it can not be written again before it is cleared automatically.</p>
0	R/W	0x0	<p>TMR1_EN Timer1 Enable</p> <p>0: Stop/Pause 1: Start</p> <p>By setting the bit to 1, the timer will be started. It reloads the interval value register and then counts from the interval value to 0.</p> <p>By setting the bit to 0 before the timer counts to 0, the timer will be paused. The bit will be locked to 0 for at least 2 cycles. Within the 2 cycles, you cannot set the bit to 1 to restart the timer.</p> <p>The timer supports updating the interval value in the pause state. To start to down-count from the updated interval value, set both the reload bit and enable bit to 1.</p> <p>Additionally, in the single counting mode, after the count value reaches 0, the system will automatically change the bit to 0 to stop the timer.</p>

### 3.6.6.7 0x0024 Timer1 Interval Value Register (Default Value: 0x0000\_0000)

Offset: 0x0024			Register Name: TMR1_INTV_VALUE_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	TMR1_INTV_VALUE Timer1 Interval Value



Take the system clock and timer clock source into consideration when setting the interval value.

### 3.6.6.8 0x0028 Timer1 Current Value Register (Default Value: 0x0000\_0000)

Offset: 0x0028			Register Name: TMR1_CUR_VALUE_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	TMR1_CUR_VALUE Timer1 Current Value Timer1 current value is a 32-bit down-counter (from interval value to 0).

### 3.6.6.9 0x00A0 Watchdog IRQ Enable Register (Default Value: 0x0000\_0000)

Offset: 0x00A0			Register Name: WDOG_IRQ_EN_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	WDOG_IRQ_EN Watchdog Interrupt Enable 0: Disabled 1: Enabled

### 3.6.6.10 0x00A4 Watchdog Status Register (Default Value: 0x0000\_0000)

Offset: 0x00A4			Register Name: WDOG_IRQ_STA_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W1C	0x0	<p>WDOG_IRQ_PEND The IRQ pending bit for the watchdog Write 1 to clear the pending status. 0: No effect 1: Pending, indicates that the interval value of the watchdog is reached.</p>

### 3.6.6.11 0x00A8 Watchdog Software Reset Register (Default Value: 0x0000\_0000)

Offset: 0x00A8			Register Name: WDOG_SOFT_RST_REG
Bit	Read/Write	Default/Hex	Description
31:16	W	0x0	<p>KEY_FIELD Key Field To change the value of bit[0], this field should be filled with 0x16AA.</p>
15:1	/	/	/
0	R/W1C	0x0	<p>Soft Reset Enable 0: De-assert 1: Reset the system <b>Note: To use the bit to reset the system, the watchdog first needs to be disabled.</b></p>

### 3.6.6.12 0x00B0 Watchdog Control Register (Default Value: 0x0000\_0000)

Offset: 0x00B0			Register Name: WDOG_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:13	/	/	/
12:1	W	0x0	<p>WDOG_KEY_FIELD Watchdog Key Field It should be written to 0xA57. Writing any other value in this field aborts the write operation.</p>

Offset: 0x00B0			Register Name: WDOG_CTRL_REG
Bit	Read/Write	Default/Hex	Description
0	R/W1S	0x0	<p>WDOG_RESTART Watchdog Restart 0: No effect 1: Restart the watchdog</p>

### 3.6.6.13 0x00B4 Watchdog Configuration Register (Default Value: 0x0000\_0001)

Offset: 0x00B4			Register Name: WDOG_CFG_REG
Bit	Read/Write	Default/Hex	Description
31:16	W	0x0	<p>KEY_FIELD Key Field To change the value of bit[15:0], this field should be filled with 0x16AA.</p>
15:9	/	/	/
8	R/W	0x0	<p>WDOG_CLK_SRC Select the clock source for the watchdog. 0: HOSC_32K, that is, OSC24M/750. It is a 32 KHz clock divided from the OSC24M. 1: LOSC_32K. A clock provided by the LOSC.</p>
7:2	/	/	/
1:0	R/W	0x1	<p>WDOG_MODE Configure the operating mode for the watchdog 00: / 01: Reset mode: Reset whole system 10: Only interrupt mode 11: /</p>

### 3.6.6.14 0x00B8 Watchdog Mode Register (Default Value: 0x0000\_0000)

Offset: 0x00B8			Register Name: WDOG_MODE_REG
Bit	Read/Write	Default/Hex	Description
31:16	W	0x0	<p>KEY_FIELD Key Field To change the value of bit[15:0], this field should be filled with 0x16AA.</p>
15:8	/	/	/
7:4	R/W	0x0	<p>WDOG_INTV_VALUE Watchdog Interval Value 0000: 16000 cycles (0.5 s) 0001: 32000 cycles (1 s) 0010: 64000 cycles (2 s) 0011: 96000 cycles (3 s) 0100: 128000 cycles (4 s) 0101: 160000 cycles (5 s) 0110: 192000 cycles (6 s) 0111: 256000 cycles (8 s) 1000: 320000 cycles (10 s) 1001: 384000 cycles (12 s) 1010: 448000 cycles (14 s) 1011: 512000 cycles (16 s) Others: Reserved <b>Note: The corresponding clock cycles for the interval value (IV) depends on the frequency of the clock: Cycles = F<sub>CLK</sub> * IV.</b> For example, to get a interval value of 0.5 second, if the clock source is HOSC_32K (whose frequency is 32 KHz), the cycle number is 16,000; if the clock source is LOSC_32K (whose frequency is 32.768 kHz), the cycle number is 16,384.</p>
3:1	/	/	/
0	R/W	0x0	<p>WDOG_EN Watchdog Enable 0: No effect 1: Enable the Watchdog</p>

**3.6.6.15 0x00BC Watchdog Output Configuration Register (Default Value: 0x0000\_001F)**

Offset: 0x00BC			Register Name: WDOG_OUTPUT_CFG_REG
Bit	Read/Write	Default/Hex	Description
31:12	/	/	/
11:0	R/W	0x1F	<p>WDOG OUTPUT CONFIG</p> <p>Configure the valid time for the watchdog reset signal.</p> <p><math>T = 1/32\text{ms} \times (N + 1)</math></p> <p>The default value is 1 ms.</p>

**3.6.6.16 0x00C0 AVS Counter Control Register (Default Value: 0x0000\_0000)**

Offset: 0x00C0			Register Name: AVS_CNT_CTL_REG
Bit	Read/Write	Default/Hex	Description
31:10	/	/	/
9	R/W	0x0	<p>AVS_CNT1_PS</p> <p>Audio/Video Sync Counter 1 Pause Control</p> <p>0: Do not pause.</p> <p>1: Pause the AVS counter1.</p>
8	R/W	0x0	<p>AVS_CNT0_PS</p> <p>Audio/Video Sync Counter 0 Pause Control</p> <p>0: Do not pause.</p> <p>1: Pause the AVS counter0.</p>
7:2	/	/	/
1	R/W	0x0	<p>AVS_CNT1_EN</p> <p>Audio/Video Sync Counter 1 Enable/Disable</p> <p>The clock source is OSC24M.</p> <p>0: Disabled</p> <p>1: Enabled</p>
0	R/W	0x0	<p>AVS_CNT0_EN</p> <p>Audio/Video Sync Counter 0 Enable/Disable</p> <p>The clock source is OSC24M.</p> <p>0: Disabled</p> <p>1: Enabled</p>

### 3.6.6.17 0x00C4 AVS Counter 0 Register (Default Value: 0x0000\_0000)

Offset: 0x00C4			Register Name: AVS_CNT0_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<p>AVS_CNT0</p> <p>The higher 32 bits of AVS counter0.</p> <p>AVS counter0 is a 33-bit up counter. The initial value consists of two parts: this register forms the bit[32:1] of AVS counter0 and the bit[0] is zero.</p> <p>You can set the initial value of the AVS counter0 by software. The initial value can be updated at anytime. You can also pause the counter by setting AVS_CNT0_PS to "1". The counter value will not increase when it is paused.</p>

### 3.6.6.18 0x00C8 AVS Counter 1 Register (Default Value: 0x0000\_0000)

Offset: 0x00C8			Register Name: AVS_CNT1_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<p>AVS_CNT1</p> <p>The higher 32 bits of AVS counter1.</p> <p>AVS counter1 is a 33-bit up counter. The initial value consists of two parts: this register forms the bit[32:1] of AVS counter0 and the bit[0] is zero.</p> <p>You can set the initial value of the AVS counter1 by software. The initial value can be updated at anytime. You can also pause the counter by setting AVS_CNT1_PS to "1". The counter value will not increase when it is paused.</p>

### 3.6.6.19 0x00CC AVS Counter Divisor Register (Default Value: 0x05DB\_05DB)

Offset: 0x00CC			Register Name: AVS_CNT_DIV_REG
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/

Offset: 0x00CC			Register Name: AVS_CNT_DIV_REG
Bit	Read/Write	Default/Hex	Description
27:16	R/W	0x5DB	<p>AVS_CNT1_D</p> <p>N1, the divisor factor for AVS1.</p> <p>The clock for AVS1 is 24 MHz/N1.</p> <p><math>N1 = \text{Bit}[27:16] + 1</math>.</p> <p>The valid value for N1 is from 1 to 0x7ff.</p> <p>There is an internal 12-bit counter maintained by the engine of the 33-bit AVS1. The 12-bit counter is used for counting the cycle number of the clock OSC24M. When the value of the 12-bit counter reaches N1, the internal 33-bit counter register will increase 1 and the 12-bit counter will reset to zero and restart again.</p> <p>You can change the value of N1 via the software at any time.</p>
15:12	/	/	/
11:0	R/W	0x5DB	<p>AVS_CNT0_D</p> <p>N0, the divisor factor for AVS0.</p> <p>The clock for AVS0 is 24MHz/N0.</p> <p><math>N0 = \text{Bit}[11:0] + 1</math>.</p> <p>The valid value for N0 is from 1 to 0x7ff.</p> <p>There is an internal 12-bit counter maintained by the engine of the 33-bit AVS0. The 12-bit counter is used for counting the cycle number of the clock OSC24M. When the value of the 12-bit counter reaches N0, the internal 33-bit counter register will increase 1 and the 12-bit counter will reset to zero and restart again.</p> <p>You can change the value of N0 via the software at any time.</p>

## 3.7 High Speed Timer

### 3.7.1 Overview

The high speed timer (HSTimer) module consists of HSTimer0 and HSTimer1. HSTimer0 and HSTimer1 are down counters that implement timing and counting functions. They are completely consistent. Compared with the timer module, the HSTimer module provides more precise timing and counting.

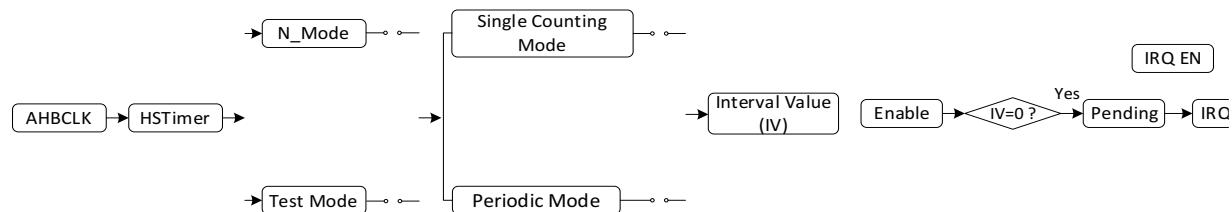
The HSTimer has the following features:

- Single clock source: AHB0
- Supports 5 prescale factors
- Configurable 56-bit down timer
- Supports 2 timing modes: periodic mode and one-shot mode
- Supports the test mode
- Generates an interrupt when the count is decreased to 0

### 3.7.2 Block Diagram

The following figure shows the block diagram of the HSTimer.

**Figure 3-16 HSTimer Block Diagram**



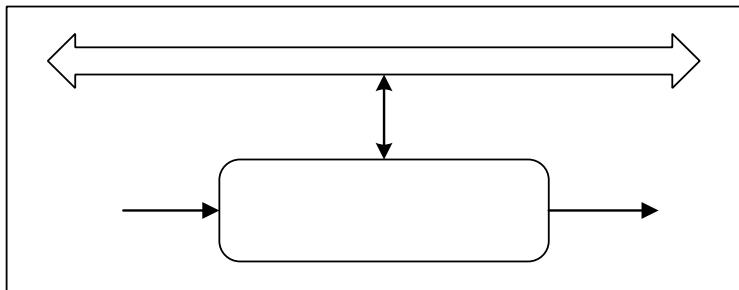
### 3.7.3 Functional Description

The HSTimers are 56-bit down counters. The counter value is decremented by 1 on each rising edge of the count clock. Each HSTimer has a prescaler that divides the working clock frequency of each working timer by 1, 2, 4, 8, or 16.

#### 3.7.3.1 Typical Application

The following figure shows a typical application of HSTimer module.

Figure 3-17 Typical Application for HSTimer



The HSTimer module is mounted at AHB0 bus, and can control the registers via AHB0 bus. AHB0 is the clock source of the HSTimer. When the count value reaches zero, the HSTimer generates an interrupt.

### 3.7.3.2 Count Modes

The HSTimer has two count modes: one-shot mode and periodic mode. You can configure the timing mode via the bit[7] of [HS\\_TMRn\\_CTRL\\_REG](#) (n = 0 or 1). The value 0 is for the period mode and value 1 is for the one-shot mode.

- One-shot Mode

When the count value of the HSTimer reaches 0, the HSTimer stops counting. The HSTimer starts to count again only when a new value is loaded.

- Periodic Mode

The HSTimer counts continuously. When the count value of the HSTimer reaches 0, the HSTimer reloads an initial value from [HS\\_TMRn\\_INTV\\_LO\\_REG](#) and [HS\\_TMRn\\_INTV\\_HI\\_REG](#) and then continues to count.

### 3.7.3.3 Operating Modes

The HSTimer has two operating modes: the normal mode and test mode. You can configure the operating mode via the bit[31] of [HS\\_TMRn\\_CTRL\\_REG](#). The value 0 is for the normal mode and value 1 is for the test mode.

- Normal Mode

In the normal mode, the HSTimer is used as a 56-bit down counter, which can finish one-shot counting and periodic counting. The interval value for the HSTimer consists of two parts: [HS\\_TMRn\\_INTV\\_LO\\_REG](#) forms the bit[31:0] and [HS\\_TMRn\\_INTV\\_HI\\_REG](#) forms the bit[55:32]. To read or write the interval value, [HS\\_TMRn\\_INTV\\_LO\\_REG](#) should be done before [HS\\_TMRn\\_INTV\\_HI\\_REG](#).

- Test Mode

In the test mode, the HSTimer is used as a 24-bit down counter. [HS\\_TMRn\\_INTV\\_LO\\_REG](#) must be set to 0x1, and [HS\\_TMRn\\_INTV\\_HI\\_REG](#) acts as the initial value for the HSTimer.

### 3.7.3.4 HSTimer Formula

The following formula describes the relationship among HSTimer parameters in the normal mode.

---

Where,

The parameter n is either 0 or 1;

$T_{HSTimer}$  is the remaining time of the timer;

$HS\_TMRn\_INTV\_HI\_REG$  is bit[55:32] of the HSTimer interval value;

$HS\_TMRn\_INTV\_LO\_REG$  is bit[31:0] of the HSTimer interval value;

$HS\_TMRn\_CURNT\_HI\_REG$  is bit[55:32] of the HSTimer current value;

$HS\_TMRn\_CURNT\_LO\_REG$  is bit[31:0] of the HSTimer current value;

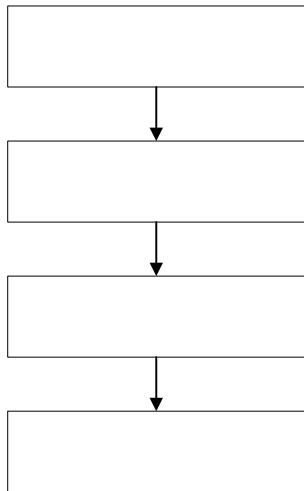
AHBOCLK is the frequency of AHBO Clock (the HSTimer clock source);

$HS\_TMRn\_CLK$  is the prescale ratio of the HSTimer clock.

### 3.7.3.5 Initializing the HSTimer

The following figure shows the process of HSTimer initialization.

Figure 3-18 HSTimer Initialization Process



1. AHB0 clock management: Open the clock gate of AHB0 and de-assert the soft reset of AHB0 in CCU.
2. Configure the corresponding parameters of the HSTimer: clock source, prescaler factor, operating mode, and timing mode. There is no sequence requirement when writing the above parameters to [HS\\_TMRn\\_CTRL\\_REG](#).
3. Write the initial value: Write the lower 32 bits of the initial value to [HS\\_TMRn\\_INTV\\_LO\\_REG](#) first, and then the higher 24 bits to [HS\\_TMRn\\_INTV\\_HI\\_REG](#). Normally, write the bit[1] of [HS\\_TMRn\\_CTRL\\_REG](#) to load the initial value. If the HSTimer is in the timing stop stage, write 1 to the bit[1] and bit[0] of [HS\\_TMRn\\_CTRL\\_REG](#) at the same time to reload the initial value.
4. Enable HSTimer: Write the bit[0] of [HS\\_TMRn\\_CTRL\\_REG](#) to enable HSTimer to count.
5. Reading [HS\\_TMRn\\_CURNT\\_LO\\_REG](#) and [HS\\_TMRn\\_CURNT\\_HI\\_REG](#) can get current counting value.

### 3.7.3.6 Processing the HSTimer Interrupt

Follow the steps below to process the HSTimer interrupt:

1. Enable interrupt: Write the corresponding interrupt enable bit of [HS\\_TMR\\_IRQ\\_EN\\_REG](#), when the counting time of HSTimer reaches, the corresponding interrupt generates.
2. After entering the interrupt process, write [HS\\_TMR\\_IRQ\\_STAS\\_REG](#) to clear the interrupt pending.
3. Resume the interrupt and continue to execute the interrupted process.

### 3.7.4 Programming Guidelines

The following example shows how to make a 1 us delay with HSTimer0. The frequency of the AHB0 clock is 100 MHz, the operating mode is the normal mode, the timing mode is single counting mode, and the pre-scale is 2.

```
writel(0x32, HS_TMR0_INTV_LO);           //Set bit[31:0] of the interval value as 0x32.

writel(0x0, HS_TMR0_INTV_HI);           //Set bit[55:32] of the interval value as 0x0.

writel(0x90, HS_TMR0_CTRL);           

//Set the operating mode as Normal Mode, the pre-scale as 2, and the timing mode as Single Counting Mode.

writel(readl(HS_TMR0_CTRL)|(1<<1), HS_TMR0_CTRL); //Set the reload bit.

writel(readl(HS_TMR0_CTRL)|(1<<0), HS_TMR0_CTRL); //Enable HSTimer0.

while(!(readl(HS_TMR_IRQ_STAS)&1));          //Wait for HSTimer0 to generate pending.

writel(1, HS_TMR_IRQ_STAS);                  //Clear HSTimer0 pending.
```

### 3.7.5 Register List

Module Name	Base Address
HSTimer	0x03008000

Register Name	Offset	Description
HS_TMR_IRQ_EN_REG	0x0000	HS Timer IRQ Enable Register
HS_TMR_IRQ_STAS_REG	0x0004	HS Timer Status Register
HS_TMR0_CTRL_REG	0x0020	HS Timer0 Control Register
HS_TMR0_INTV_LO_REG	0x0024	HS Timer0 Interval Value Low Register
HS_TMR0_INTV_HI_REG	0x0028	HS Timer0 Interval Value High Register
HS_TMR0_CURNT_LO_REG	0x002C	HS Timer0 Current Value Low Register
HS_TMR0_CURNT_HI_REG	0x0030	HS Timer0 Current Value High Register
HS_TMR1_CTRL_REG	0x0040	HS Timer1 Control Register
HS_TMR1_INTV_LO_REG	0x0044	HS Timer1 Interval Value Low Register
HS_TMR1_INTV_HI_REG	0x0048	HS Timer1 Interval Value High Register
HS_TMR1_CURNT_LO_REG	0x004C	HS Timer1 Current Value Low Register
HS_TMR1_CURNT_HI_REG	0x0050	HS Timer1 Current Value High Register

### 3.7.6 Register Description

#### 3.7.6.1 0x0000 HS Timer IRQ Enable Register (Default Value: 0x0000\_0000)

Offset: 0x0000			Register Name: HS_TMR_IRQ_EN_REG
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1	R/W	0x0	HS_TMR1_INT_EN HSTimer1 Interrupt Enable 0: Disabled 1: Enabled
0	R/W	0x0	HS_TMR0_INT_EN HSTimer0 Interrupt Enable 0: Disabled 1: Enabled

#### 3.7.6.2 0x0004 HS Timer IRQ Status Register (Default Value: 0x0000\_0000)

Offset: 0x0004			Register Name: HS_TMR_IRQ_STAS_REG
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1	R/W1C	0x0	HS_TMR1_IRQ_PEND HSTimer1 IRQ Pending The IRQ pending bit for HSTimer1. Write 1 to clear the pending status. 0: No effect 1: Pending, indicates that the initial value of the HSTimer is reached.
0	R/W1C	0x0	HS_TMR0_IRQ_PEND HSTimer0 IRQ Pending The IRQ pending bit for HSTimer0. Write 1 to clear the pending status. 0: No effect 1: Pending, indicates that the initial value of the HSTimer is reached.

### 3.7.6.3 0x0020 HS Timer0 Control Register (Default Value: 0x0000\_0000)

Offset: 0x0020			Register Name: HS_TMR0_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<p>HS_TMR0_TEST Select the operating mode for HSTimer0 0: Normal mode 1: Test mode In the test mode, the <b>HS_TMR0_INTV_LO_REG</b> must be set to 0x1, and <b>HS_TMR0_INTV_HI_REG</b> acts as the initial value for HSTimer0.</p>
30:8	/	/	/
7	R/W	0x0	<p>HS_TMR0_MODE Select the timing mode for HSTimer0 0: Periodic mode. When the count value is decreased to 0, the timer will restart another round of counting automatically. 1: One-shot mode. When the count value is decreased to 0, the timer will stop counting.</p>
6:4	R/W	0x0	<p>HS_TMR0_CLK Select the pre-scale for the HSTimer0 clock sources 000: /1 001: /2 010: /4 011: /8 100: /16 101: / 110: / 111: /</p>
3:2	/	/	/
1	R/W1S	0x0	<p>HS_TMR0_RELOAD HSTimer0 Reload 0: No effect 1: Reload the interval value of the HSTimer0</p>

Offset: 0x0020			Register Name: HS_TMR0_CTRL_REG
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	<p>HS_TMR0_EN HSTimer0 Enable 0: Stop/Pause 1: Start</p> <p>By setting the bit to 1, the timer will be started. It reloads the interval value register and then counts from the interval value to 0.</p> <p>By setting the bit to 0 before the timer counts to 0, the timer will be paused. The bit will be locked to 0 for at least 2 cycles. Within the 2 cycles, you cannot set the bit to 1 to restart the timer.</p> <p>The timer supports updating the interval value in the pause state. To start from the updated interval value, set both the reload bit and enable bit to 1.</p> <p>Additionally, in the one-shot mode, after the count value reaches 0, the system will automatically change the bit to 0 to stop the timer.</p>

### 3.7.6.4 0x0024 HS Timer0 Interval Value Lo Register (Default Value: 0x0000\_0000)

Offset: 0x0024			Register Name: HS_TMR0_INTV_LO_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<p>HS_TMR0_INTV_VALUE_LO Bit[31:0] of the HSTimer0 interval value.</p>

### 3.7.6.5 0x0028 HS Timer0 Interval Value Hi Register (Default Value: 0x0000\_0000)

Offset: 0x0028			Register Name: HS_TMR0_INTV_HI_REG
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23:0	R/W	0x0	<p>HS_TMR0_INTV_VALUE_HI Bit[55:32] of the HSTimer0 interval value.</p>

**NOTE**

HSTimer0 is a 56-bit counter. The interval value consists of two parts: HS\_TMR0\_INTV\_VALUE\_LO acts as the bit[31:0] and HS\_TMR0\_INTV\_VALUE\_HI acts as the bit[55:32]. To read or write the interval value, HS\_TMR0\_INTV\_LO\_REG should be done before HS\_TMR0\_INTV\_HI\_REG.

### 3.7.6.6 0x002C HS Timer0 Current Value Lo Register (Default Value: 0x0000\_0000)

Offset: 0x002C			Register Name: HS_TMR0_CURNT_LO_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	HS_TMR0_CUR_VALUE_LO Bit[31:0] of the HSTimer0 current value.

### 3.7.6.7 0x0030 HS Timer0 Current Value Hi Register (Default Value: 0x0000\_0000)

Offset: 0x0030			Register Name: HS_TMR0_CURNT_HI_REG
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23:0	R/W	0x0	HS_TMR0_CUR_VALUE_HI Bit[55:32] of the HSTimer0 current value.

**NOTE**

HSTimer0 is a 56-bit counter. The current value consists of two parts: HS\_TMR0\_CUR\_VALUE\_LO acts as the bit[31:0] and HS\_TMR0\_CUR\_VALUE\_HI acts as the bit[55:32]. To read or write the current value, HS\_TMR0\_CUR\_VALUE\_LO should be done before HS\_TMR0\_CUR\_VALUE\_HI.

### 3.7.6.8 0x0040 HS Timer1 Control Register (Default Value: 0x0000\_0000)

Offset: 0x0040			Register Name: HS_TMR1_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<p>HS_TMR1_TEST Select the operating mode for HSTimer1. 0: Normal mode 1: Test mode In the test mode, the <b>HS_TMR1_INTV_LO_REG</b> must be set to 0x1, and <b>HS_TMR1_INTV_HI_REG</b> acts as the interval value for HSTimer1.</p>
30:8	/	/	/
7	R/W	0x0	<p>HS_TMR1_MODE Select the timing mode for HSTimer1. 0: Periodic mode. When the count value is decreased to 0, the timer will restart another round of counting automatically. 1: One-shot mode. When the count value is decreased to 0, the timer will stop counting.</p>
6:4	R/W	0x0	<p>HS_TMR1_CLK Select the pre-scale of the HSTimer1 clock sources. 000: /1 001: /2 010: /4 011: /8 100: /16 101: / 110: / 111: /</p>
3:2	/	/	/
1	R/W1S	0x0	<p>HS_TMR1_RELOAD HSTimer1 Reload 0: No effect 1: Reload the HSTimer1 interval value.</p>

Offset: 0x0040			Register Name: HS_TMR1_CTRL_REG
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	<p>HS_TMR1_EN HSTimer1 Enable 0: Stop/Pause 1: Start</p> <p>By setting the bit to 1, the timer will be started. It reloads the interval value register and then counts from the interval value to 0.</p> <p>By setting the bit to 0 before the timer counts to 0, the timer will be paused. The bit will be locked to 0 for at least 2 cycles. Within the 2 cycles, you cannot set the bit to 1 to restart the timer.</p> <p>The timer supports updating the interval value in the pause state. To start from the updated interval value, set both the reload bit and enable bit to 1.</p> <p>Additionally, in the one-shot mode, after the count value reaches 0, the system will automatically change the bit to 0 to stop the timer.</p>

### 3.7.6.9 0x0044 HS Timer1 Interval Value Lo Register (Default Value: 0x0000\_0000)

Offset: 0x0044			Register Name: HS_TMR1_INTV_LO_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<p>HS_TMR1_INTV_VALUE_LO Bit[31:0] of the HSTimer1 interval value</p>

### 3.7.6.10 0x0048 HS Timer1 Interval Value Hi Register (Default Value: 0x0000\_0000)

Offset: 0x0048			Register Name: HS_TMR1_INTV_HI_REG
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23:0	R/W	0x0	<p>HS_TMR1_INTV_VALUE_HI Bit[55:32] of the HSTimer1 interval value</p>



HSTimer1 is a 56-bit counter. The interval value consists of two parts: HS\_TMR1\_INTV\_VALUE\_LO acts as the bit[31:0] and HS\_TMR1\_INTV\_VALUE\_HI acts as the bit[55:32]. To read or write the interval value, HS\_TMR1\_INTV\_LO\_REG should be done before HS\_TMR1\_INTV\_HI\_REG.

### 3.7.6.11 0x004C HS Timer1 Current Value Lo Register (Default Value: 0x0000\_0000)

Offset: 0x004C			Register Name: HS_TMR1_CURNT_LO_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	HS_TMR1_CUR_VALUE_LO Bit[31:0] of the HSTimer1 current value

### 3.7.6.12 0x0050 HS Timer1 Current Value Hi Register (Default Value: 0x0000\_0000)

Offset: 0x0050			Register Name: HS_TMR1_CURNT_HI_REG
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23:0	R/W	0x0	HS_TMR1_CUR_VALUE_HI Bit[55:32] of the HSTimer1 current value



HSTimer1 is a 56-bit counter. The current value consists of two parts: HS\_TMR1\_CUR\_VALUE\_LO acts as the bit[31:0] and HS\_TMR1\_CUR\_VALUE\_HI acts as the bit[55:32]. To read or write the current value, HS\_TMR1\_CUR\_VALUE\_LO should be done before HS\_TMR1\_CUR\_VALUE\_HI.

## 3.8 Generic Interrupt Controller (GIC)

### 3.8.1 Overview

The GIC-400 is a high-performance, area-optimized interrupt controller with an Advanced Microcontroller Bus Architecture (AMBA) Advanced eXtensible Interface (AXI) interface. It detects, manages, and distributes interrupts in System on Chip (SoC) configurations. You can configure the GIC-400 to provide the optimum features, performance, and gate count required for your intended application.

For details about GIC, please refer to the [GIC PL400 technical reference](#) manual and [ARM GIC Architecture Specification V2.0](#).

### 3.8.2 Functional Description

The following table describes the details of interrupt sources.

**Table 3-12 Interrupt Sources**

Interrupt Number	Interrupt Source	Interrupt Vector	Description
0	SGI 0	0x0000	SGI 0 interrupt
1	SGI 1	0x0004	SGI 1 interrupt
2	SGI 2	0x0008	SGI 2 interrupt
3	SGI 3	0x000C	SGI 3 interrupt
4	SGI 4	0x0010	SGI 4 interrupt
5	SGI 5	0x0014	SGI 5 interrupt
6	SGI 6	0x0018	SGI 6 interrupt
7	SGI 7	0x001C	SGI 7 interrupt
8	SGI 8	0x0020	SGI 8 interrupt
9	SGI 9	0x0024	SGI 9 interrupt
10	SGI 10	0x0028	SGI 10 interrupt
11	SGI 11	0x002C	SGI 11 interrupt
12	SGI 12	0x0030	SGI 12 interrupt
13	SGI 13	0x0034	SGI 13 interrupt
14	SGI 14	0x0038	SGI 14 interrupt
15	SGI 15	0x003C	SGI 15 interrupt
16	PPI 0	0x0040	PPI 0 interrupt
17	PPI 1	0x0044	PPI 1 interrupt
18	PPI 2	0x0048	PPI 2 interrupt

<b>Interrupt Number</b>	<b>Interrupt Source</b>	<b>Interrupt Vector</b>	<b>Description</b>
19	PPI 3	0x004C	PPI 3 interrupt
20	PPI 4	0x0050	PPI 4 interrupt
21	PPI 5	0x0054	PPI 5 interrupt
22	PPI 6	0x0058	PPI 6 interrupt
23	PPI 7	0x005C	PPI 7 interrupt
24	PPI 8	0x0060	PPI 8 interrupt
25	PPI 9	0x0064	PPI 9 interrupt
26	PPI 10	0x0068	PPI 10 interrupt
27	PPI 11	0x006C	PPI 11 interrupt
28	PPI 12	0x0070	PPI 12 interrupt
29	PPI 13	0x0074	PPI 13 interrupt
30	PPI 14	0x0078	PPI 14 interrupt
31	PPI 15	0x007C	PPI 15 interrupt
32		0x0080	
33		0x0084	
34	UART0	0x0088	
35	UART1	0x008C	
36	UART2	0x0090	
37	UART3	0x0094	
38	UART4	0x0098	
39	UART5	0x009C	
40		0x00A0	
41	TWI0	0x00A4	
42	TWI1	0x00A8	
43	TWI2	0x00AC	
44	TWI3	0x00B0	
45		0x00B4	
46		0x00B8	
47	SPI0	0x00BC	
48	SPI1	0x00C0	
49		0x00C4	
50	PWM	0x00C8	
51	IR_TX	0x00CC	
52	LEDC	0x00D0	

Interrupt Number	Interrupt Source	Interrupt Vector	Description
53		0x00D4	
54		0x00D8	
55	OWA	0x00DC	
56	DMIC	0x00E0	
57	AUDIO_CODEC	0x00E4	
58		0x00E8	
59	I2S/PCM1	0x00EC	
60	I2S/PCM2	0x00F0	
61	USBO_DEVICE	0x00F4	
62	USBO_EHCI	0x00F8	
63	USBO_OHCI	0x00FC	
64		0x0100	
65	USB1_EHCI	0x0104	
66	USB1_OHCI	0x0108	
67		0x010C	
68		0x0110	
69		0x0114	
70		0x0118	
71		0x011C	
72	SMHC0	0x0120	
73	SMHC1	0x0124	
74	SMHC2	0x0128	
75	MSI	0x012C	
76	SMC	0x0130	
77		0x0134	
78	EMAC	0x0138	
79	TZMA_ERR	0x013C	SRAM_A1 TZMA IRQ
80	CCU_FERR	0x0140	
81	AHB_HREADY_TIME_OUT	0x0144	SYS_CTRL ahb_hready time out
82	DMAC_NS	0x0148	
83	DMAC_S	0x014C	
84	CE_NS	0x0150	
85	CE_S	0x0154	

Interrupt Number	Interrupt Source	Interrupt Vector	Description
86		0x0158	
87	HSTIMERO	0x015C	
88	HSTIMER1	0x0160	
89	GPADC	0x0164	
90	THS	0x0168	Thermal Sensor IRQ
91	TIMERO	0x016C	
92	TIMER1	0x0170	
93		0x0174	
94		0x0178	
95	WATCHDOG	0x017C	
96	IOMMU	0x0180	
97		0x0184	
98	VE	0x0188	
99		0x018C	
100		0x0190	
101	GPIOB_NS	0x0194	
102	GPIOB_S	0x0198	
103	GPIOC_NS	0x019C	
104	GPIOC_S	0x01A0	
105	GPIOD_NS	0x01A4	
106	GPIOD_S	0x01A8	
107		0x01AC	
108		0x01B0	
109	GPIOF_NS	0x01B4	
110	GPIOF_S	0x01B8	
111	GPIOG_NS	0x01BC	
112	GPIOG_S	0x01C0	
113		0x01C4	
114		0x01C8	
115		0x01CC	
116		0x01D0	
117		0x01D4	
118		0x01D8	
119	DE	0x01DC	

Interrupt Number	Interrupt Source	Interrupt Vector	Description
120	DI	0x01E0	
121	G2D	0x01E4	
122	LCD	0x01E8	
123	TV	0x01EC	
124	DSI	0x01F0	
125	HDMI	0x01F4	
126		0x01F8	
127		0x01FC	
128		0x0200	
129		0x0204	
130		0x0208	
131		0x020C	
132		0x0210	
133		0x0214	
134		0x0218	
135		0x021C	
136		0x0220	
137		0x0224	
138		0x0228	
139		0x022C	
140		0x0230	
141		0x0234	
142		0x0238	
143		0x023C	
144		0x0240	
145		0x0244	
146		0x0248	
147		0x024C	
148		0x0250	
149		0x0254	
150		0x0258	
151		0x025C	
152		0x0260	
153		0x0264	

<b>Interrupt Number</b>	<b>Interrupt Source</b>	<b>Interrupt Vector</b>	<b>Description</b>
154		0x0268	
155		0x026C	
156		0x0270	
157		0x0274	
158		0x0278	
159		0x027C	
160		0x0280	
161		0x0284	
162		0x0288	
163		0x028C	
164		0x0290	
165		0x0294	
166		0x0298	
167		0x029C	
<b>CPUS Related</b>			
168	NMI	0x02A0	
169	PPU	0x02A4	
170	TWD	0x02A8	
171		0x02AC	
172	TIMER0	0x02B0	
173	TIMER1	0x02B4	
174	TIMER2	0x02B8	
175	TIMER3	0x02BC	
176	ALARM0	0x02C0	
177		0x02C4	
178		0x02C8	
179		0x02CC	
180		0x02D0	
181		0x02D4	
182		0x02D8	
183	IRRX	0x02DC	CPUS_IRRX_IRQ[9:0] OR to 1-bit in syscpus top
184		0x02E0	
185		0x02E4	

<b>Interrupt Number</b>	<b>Interrupt Source</b>	<b>Interrupt Vector</b>	<b>Description</b>
186	AHBS_HREADY_TIME_OUT	0x02EC	CPUS AHB READY TIME OUT IRQ
187		0x02F0	
188		0x02F4	
189		0x02F8	
190		0x02FC	
191		0x0300	
<b>CPUX Related</b>			
192	C0_CTL0	0x0304	C0_CTL0 interrupt
193	C0_CTL1	0x0308	C0_CTL1 interrupt
194		0x030C	
195		0x0310	
196	C0_COMMTX0	0x0314	C0_COMMTX0 interrupt
197	C0_COMMTX1	0x0318	C0_COMMTX1 interrupt
198		0x031C	
199		0x0320	
200	C0_COMMRX0	0x0324	C0_COMMRX0 interrupt
201	C0_COMMRX1	0x0328	C0_COMMRX1 interrupt
202		0x032C	
203		0x0330	
204	C0_PMU0	0x0334	C0_PMU0 interrupt
205	C0_PMU1	0x0338	C0_PMU1 interrupt
206		0x033C	
207		0x0340	
208	C0_AXI_ERROR	0x0344	C0_AXI_ERROR interrupt
209		0x0348	
210	AXI_WR_IRQ	0x034C	
211	AXI_RD_IRQ	0x0350	
212	DBGWRUPREQ_out[0]	0x0354	
213	DBGWRUPREQ_out[1]	0x0358	
214		0x035C	
215		0x0360	
216		0x0364	
217		0x0368	

Interrupt Number	Interrupt Source	Interrupt Vector	Description
218		0x036C	
219		0x0370	
220		0x0374	
221		0x0378	
222		0x037C	
223		0x0380	

## 3.9 Direct Memory Access Controller (DMAC)

### 3.9.1 Overview

The direct memory access (DMA) is a method of transferring data between peripherals and memories (including the SRAM and DRAM) without using the CPU. It is an efficient way to offload data transfer duties from the CPU. Without DMA, the CPU has to control all the data transfers. While with DMA, the DMAC directly transfers data between a peripheral and a memory, between peripherals, or between memories.

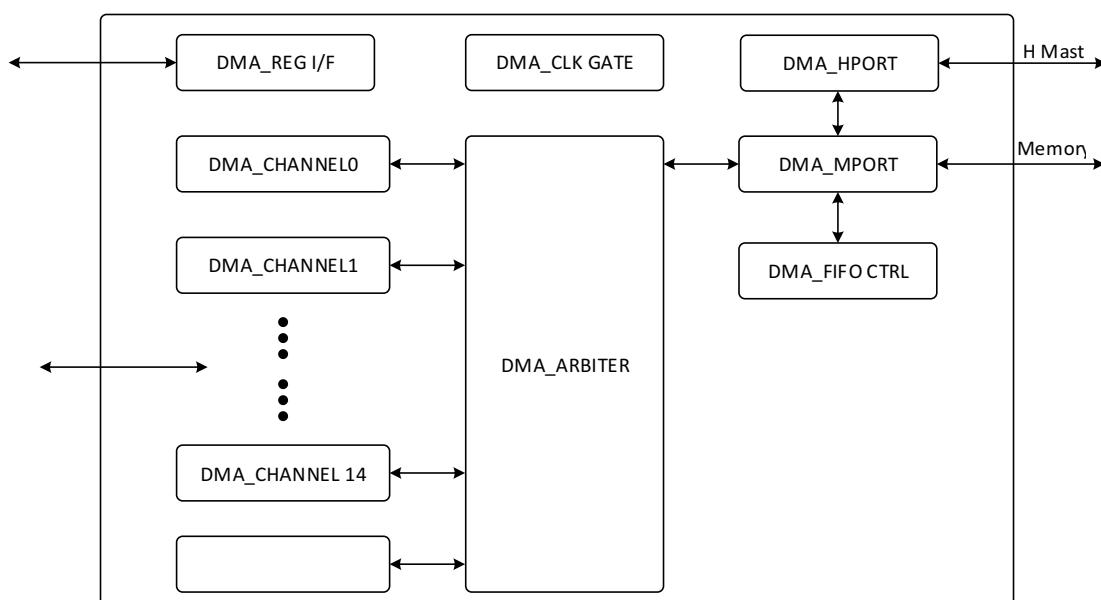
The DMAC has the following features:

- Up to 16 DMA channels
- Provides 32 peripheral DMA requests for data reading and 32 peripheral DMA requests for data writing
- Supports transferring data with a linked list
- Supports programmable 8-bit, 16-bit, 32-bit, and 64-bit data width
- Supports programmable DMA burst length
- DRQ response includes the waiting mode and handshake mode
- DMA channel supports pause function
- Memory devices support non-aligned transform

### 3.9.2 Block Diagram

The following figure shows a block diagram of DMAC.

**Figure 3-19 DMAC Block Diagram**



DMAC contains the following sub-blocks:

**Table 3-13 DMAC Sub-blocks**

Sub-block	Description
DMA_ARBITER	Arbitrates the DMA read/write requests from all channels, and converts the requests to the read/write requests of ports.
DMA_CHANNELS	DMA transfer engine. Each channel is independent. When the DMA requests from multiple peripherals are valid simultaneously, the channel with the highest priority starts data transfer first. The system uses the polling mechanism to decide the priorities of DMA channels. When DMA_ARBITER is idle, channel 0 has the highest priority, whereas channel 15 has the lowest priority. When DMA_ARBITER is busy processing the request from channel n, channel (n+1) has the highest priority. For n = 15, the channel (n + 1) should be channel 0.
DRQs	DMA requests. Peripherals use the DMA request signals to request a data transfer.
DMA_MPORT	Receives the read/write requests from DMA_ARBITER, and converts the requests to the corresponding MBUS access requests. It is mainly used for accessing the DRAM.
DMA_HPORT	The port for accessing the AHB Master. It is mainly used for accessing the SRAM and IO devices.
DMA_FIFO CTRL	Internal FIFO cell control module.
DMA_REG Interface	DMA_REG is the common register module that is mainly used to resolve AHB demands.
DMA_CLKGATE	The control module for hardware auto clock gating.

The DMAC integrates 16 independent DMA channels and each channel has an independent FIFO controller. When the DMA channel starts, the DMAC gets a DMA descriptor from the DMA\_DESC\_ADDR\_REG and uses it as the configuration information for the data transfer of the current DMA package. Then the DMAC can transfer data between the specified devices. After transferring a DMA package, the DMAC judges if the current channel transfer is finished via the linked address in the descriptor. If the linked address shows all the packages are transferred, the DMAC will end the chain transmission and close the channel.

### 3.9.3 Functional Description

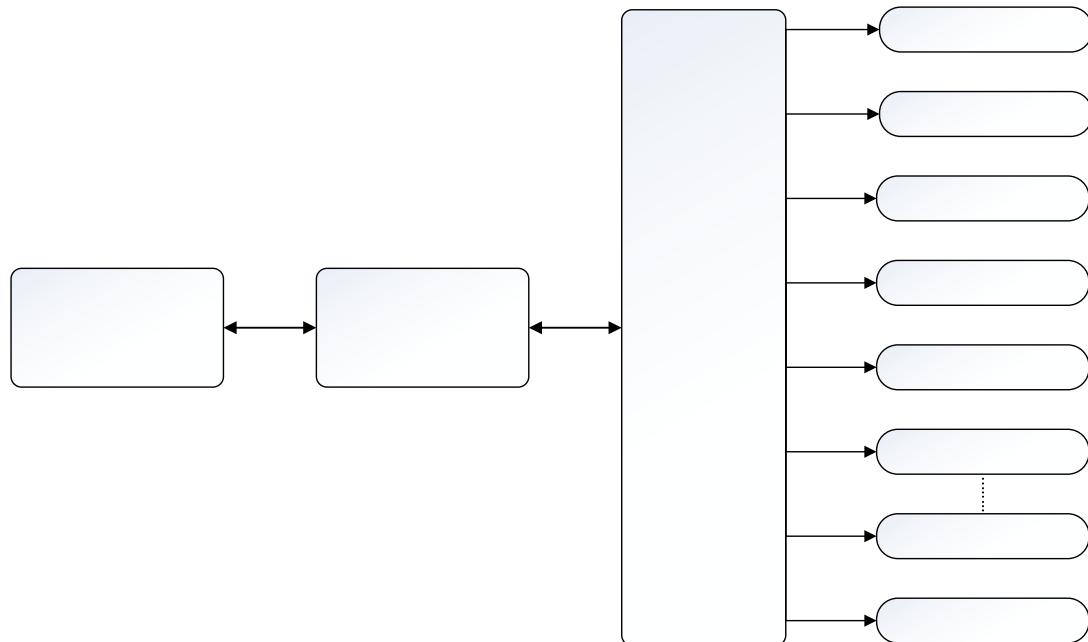
#### 3.9.3.1 Clock

The DMAC is on MBUS. The clock of MBUS influences the transfer efficiency of the DMAC.

### 3.9.3.2 Typical Application

The following figure shows a typical application of the DMAC.

Figure 3-20 DMAC Typical Application Diagram



### 3.9.3.3 DRQ Port of Peripherals

The following table shows the source DRQ types and destination DRQ types of different ports.

Table 3-14 DMA DRQ Type

Source DRQ Type		Destination DRQ Type	
port0	SRAM	port0	SRAM
port1	DRAM	port1	DRAM
port2		port2	OWA-TX
port3		port3	
port4	I2S/PCM1-RX	port4	I2S/PCM1-TX
port5	I2S/PCM2-RX	port5	I2S/PCM2-TX
port6		port6	
port7	Audio_Codec	port7	Audio_Codec
port8	DMIC	port8	

Source DRQ Type		Destination DRQ Type	
port9		port9	
port10		port10	
port11		port11	
port12	GPADC	port12	
port13		port13	IR-TX
port14	UART0-RX	port14	UART0-TX
port15	UART1-RX	port15	UART1-TX
port16	UART2-RX	port16	UART2-TX
port17	UART3-RX	port17	UART3-TX
port18	UART4-RX	port18	UART4-TX
port19	UART5-RX	port19	UART5-TX
port20		port20	
port21		port21	
port22	SPI0-RX	port22	SPI0-TX
port23	SPI1-RX	port23	SPI1-TX
port24		port24	
port25		port25	
port26		port26	
port27		port27	
port28		port28	
port29		port29	
Port30	USBO_EP1	Port30	USBO_EP1
Port31	USBO_EP2	Port31	USBO_EP2
Port32	USBO_EP3	Port32	USBO_EP3
Port33	USBO_EP4	Port33	USBO_EP4
Port34	USBO_EP5	Port34	USBO_EP5
Port35		Port35	
Port36		Port36	
Port37		Port37	
Port38		Port38	
Port39		Port39	
Port40		Port40	
Port41		Port41	
Port42		Port42	LEDC

Source DRQ Type		Destination DRQ Type	
Port43	TWI0	Port43	TWI0
Port44	TWI1	Port44	TWI1
Port45	TWI2	Port45	TWI2
Port46	TWI3	Port46	TWI3
Port47		Port47	
Port48		Port48	
Port49		Port49	
Port50		Port50	
Port51		Port51	
Port52		Port52	
Port53		Port53	

### 3.9.3.4 DMA Descriptor

The DMAC descriptor is the configuration information of DMA transfer that decides the DMA working mode. Each descriptor includes 6 words: Configuration, Source Address, Destination Address, Byte Counter, Parameter, and Link. The following figure shows the structure of the DMA descriptor.

**Figure 3-21 DMA Descriptor**



- **Configuration:** Configure the following information by DMA\_CFG\_REG.
  - DRQ type: DRQ type of the source and destination devices.
  - Address counting mode: For both the source and destination devices, there are two address counting modes: the IO mode and linear mode. The IO mode is for IO devices whose address is fixed during the data transfer and the linear mode is for the memory whose address is increasing during the data transfer.

- Transferred block length: The amount of data that non-memory peripherals can transfer in a valid DRQ. The block length supports 1 bit, 4 bits, 8 bits, and 16 bits.
- Transferred data width: The data width of operating the non-memory peripherals. The data width supports 8 bits, 16 bits, 32 bits, and 64 bits.



### NOTE

The configuration supports BMODE mode. The BMODE is used in the following scenario: the source is an IO device, and the destination is a memory device. Setting the BMODE mode can limit the amount of block data transferred in DMA block transmission to the amount of data transferred when the DRQ threshold of the source IO device is 1.

- **Source Address:** Configure the address of the source device.
- **Destination Address:** Configure the address of the destination device.

DMA reads data from the source address and then writes data to the destination address.

Both the DMA source and destination addresses have 34 bits. In the descriptor, because there are only 32 bits in the **Source/Destination Address** field, another 2 bits are stored in the **Parameter** field.

The following table shows the details of the related fields in the descriptor.

**Table 3-15 Source/Destination Address Distribution**

Descriptor Group	Bit	Description
Source Address	31:0	DMA transfers the lower 32 bits of the 34-bit source address
Destination Address	31:0	DMA transfers the lower 32 bits of the 34-bit destination address
Parameter	31:20	Reserved
	19:18	DMA transfers the higher 2 bits of the 34-bit destination address
	17:16	DMA transfers the high 2 bits of the 34-bit source address
	15:8	Reserved
	7:0	Wait Clock Cycles Set the waiting time in DRQ mode
Link	31:2	The address of the next group descriptor, the lower 30 bits of the word address
	1:0	The address of the next group descriptor, the higher 2 bits of the word address

From the above table, you can get:

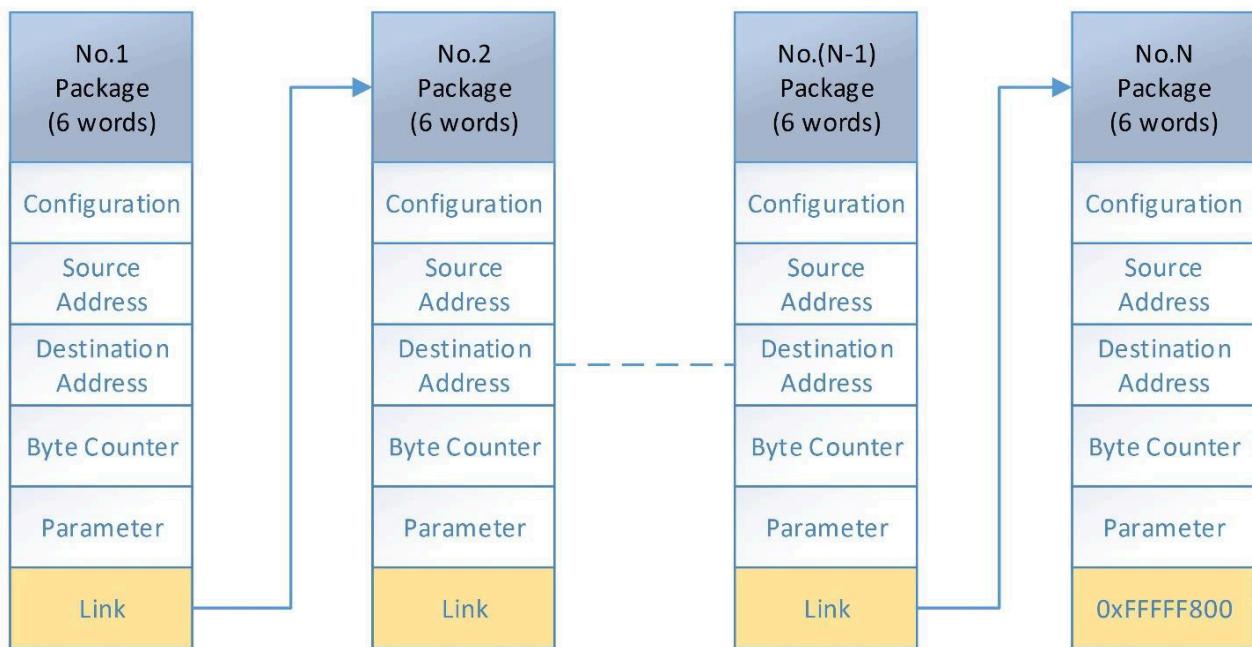
Real DMA source address (in byte mode) = {Parameter [17:16], Source Address [31:0]};

Real DMA destination address (in byte mode) = {Parameter [19:18], Destination Address [31:0]};

Real link address (in byte mode) = {Link[1:0], Link[31:2], 2'b00}.

- **Byte counter:** Configure the data amount of a package. The maximum value is (2<sup>25</sup>-1) bytes. If the data amount of the package reaches the maximum value, even if DRQ is valid, the DMA will stop the current transfer.
- **Parameter:** Configure the interval between the data block. The parameter is valid for non-memory peripherals. When DMA detects that the DRQ is high, the DMA transfers the data block and ignores the status changes of the DRQ until the data transfer finishes. After that, the DMA waits for certain clock cycles (WAIT\_CYC) and executes the next DRQ detection.
- **Link:** If the value of the link is 0xFFFF800, the current package is at the end of the linked list. The DMAC will stop the data transfer after transferring the package; otherwise, the value of the link is considered as the descriptor address of the next package.

Figure 3-22 DMA Chain Transfer



### 3.9.3.5 Interrupts

There are three kinds of DMA interrupts: the half package interrupt, package end interrupt, and queue end interrupt.

**Half package interrupt:** When enabled, the DMAC sends out a half package interrupt after transferring half of a package.

**Package end interrupt:** When enabled, the DMAC sends out a package end interrupt after transferring a complete package.

---

**Queue end interrupt:** When enabled, the DMAC sends out a queue end interrupt after transferring a complete queue.

Notice that when CPU does not respond to the interrupts timely, or two DMA interrupts are generated very closely, the later interrupt may override the former one. That is, from the perspective of the CPU, the DMAC has only a system interrupt source.

---

 **NOTE**

The DMAC has 16 channels and 2 groups of interrupts. The channel [7:0] corresponds to one group of interrupt, the channel [15:8] corresponds to another group of interrupt. Each group of interrupt includes the secure and non-secure interrupts.

---

### 3.9.3.6 Security

The DMAC supports the system Trustzone and the security mode configuration of the DMA channel.

Each DMA channel is secure by default. When system Trustzone is enabled, the DMAC is secure, only the secure devices can access the DMAC.

When a DMA channel is configured to the non-secure mode, this channel can only access the non-secure area of the memory, that is, the DMAC cannot write data to the secure area of the memory, and the read-back data from the secure area of the memory is 0.

### 3.9.3.7 Clock Gating

The DMA\_CLK\_GATE module is a hardware module for controlling the clock gating automatically. It provides clock sources for sub-modules in DMAC and the module local circuits.

The DMA\_CLK\_GATE module consists of two parts: the channel clock gate and the common clock gate.

**Channel clock gate:** Controls the DMA clock of the DMA channels. When the system accesses the register of the current DMA channel and the DMA channel is enabled, the channel clock gate automatically opens the DMA clock. With a 16-HCLK-cycle delay after the system finishes accessing the register or the DMA data transfer is completed, the channel clock gate automatically closes the DMA clock. Also, the clock for the related circuits, such as for the channel control and FIFO control modules, will be closed.

**Common clock gate:** Controls the clocks of the DMA common circuits. The common circuits include the common circuit of the FIFO control module, MPORT module, and MBUS. When all the DMA channels are enabled, the common clock gate automatically closes the clocks for the above circuits.

The DMA clock gating can support all the functions stated above or not by software.

### 3.9.3.8 Transfer Mode

The peripherals initiate data transfer by transmitting DMA request signals to the DMAC. After receiving the request signal, the DMAC converts it to the internal DRQ signal and controls the DMA data transfer.

The DMAC supports two data transfer modes: the waiting mode and handshake mode.

#### The principle of waiting mode

- When the DMAC detects a valid external request signal, the DMAC starts to operate the peripheral device. The internal DRQ always holds high before the transferred data amount reaches the transferred block length.
- When the transferred data amount reaches the transferred block length, the internal DRQ pulls low automatically.
- The internal DRQ holds low for certain clock cycles (WAIT\_CYC), and then the DMAC restarts to detect the external requests. If the external request signal is valid, then the next transfer starts.

#### The principle of handshake mode

- When the DMAC detects a valid external request signal, the DMAC starts to operate the peripheral device. The internal DRQ always holds high before the transferred data amount reaches the transferred block length.
- When the transferred data amount reaches the transferred block length, the internal DRQ will be pulled down automatically. For the last data transfer of the block, the DMAC sends a DMA Last signal with the DMA commands to the peripheral device. The DMA Last signal will be packed as part of the DMA commands and transmitted on the bus. It is used to inform the peripheral device that it is the end of the data transfer for the current DRQ.
- When the peripheral device receives the DMA Last signal, it can judge that the data transfer for the current DRQ is finished. To continue the data transfer, it sends a DMA Active signal to the DMAC.



#### NOTE

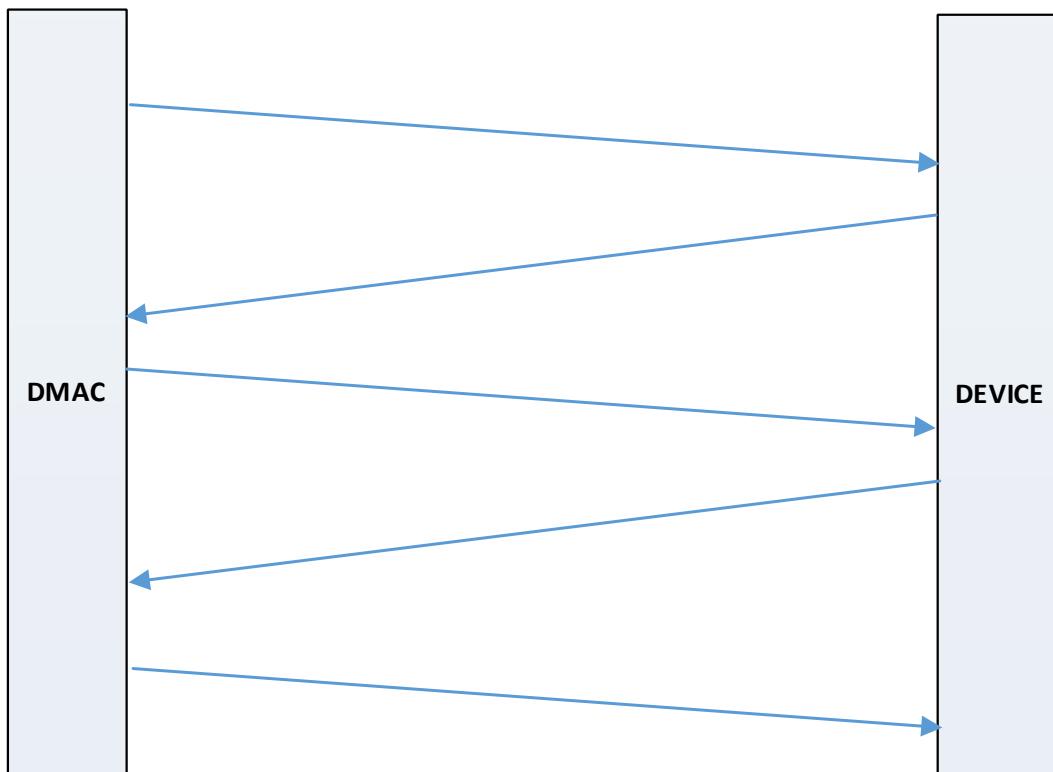
One DMA Active signal will be converted to one DRQ signal in the DMA module. To generate multiple DRQs, the peripheral device needs to send out multiple DMA Active signals via the bus protocol.

- When the DMAC received the DMA Active signal, it sends back a DMA ACK signal to the peripheral device.
- When the peripheral device receives the DMA ACK signal, it waits for all the operations on the local device completed, and both the FIFO and DRQ status refreshed. Then it invalidates the DMA Active signal.

- When the DMAC detects the falling edge of the DMA Active signal, it invalidates the corresponding DMA ACK signal, and restarts to detect the external request signals. If a valid request signal is detected, the next data transfer starts.

The following figure shows the workflow of the handshake mode.

**Figure 3-23 Workflow of the DMAC Handshake Mode**



### 3.9.3.9 Address Auto-Alignment

For the non-IO devices whose start address is not 32-byte-aligned, the DMAC will adjust the address to 32-byte-aligned through the burst transfer within 32 bytes. Adjusting address to 32-byte-aligned improves the DRAM access efficiency.

The following example shows how the DMAC adjusts the address: when the peripheral device of a DMA channel is a non-IO device whose start address is 0x86 (not 32-byte-aligned), the DMAC firstly uses a 26-byte burst transfer to align the address to 0xA0 (32-byte-aligned), and then transfers data by 64-byte burst (the maximum transfer amount that MBUS allows).

The IO devices do not support address alignment, so the bit width of IO devices must match the address offset; otherwise, the DMAC will ignore the inconsistency and directly transmit data of the corresponding bit width to the address.

The address of the DMA descriptor does not support the address auto-alignment. Make sure the address is word-aligned; otherwise the DMAC cannot identify the descriptor.

### 3.9.3.10 DMAC Clock Control

- The DMAC clock is synchronous with the AHBO clock. Make sure that the DMAC gating bit of AHBO clock is enabled before accessing the DMAC register.
- The reset input signal of the DMAC is asynchronous with AHBO and is low valid by default. Make sure that the reset signal of the DMAC is de-asserted before accessing the DMA register.
- To avoid the indefinite state within registers, de-assert the reset signal first, and then open the gating bit of AHBO.
- The DMAC supports Clock Auto Gating function to reduce power consumption, the system will automatically disable the DMAC clock in the DMAC idle state. Clock Auto Gating is enabled by default.

## 3.9.4 Programming Guidelines

### 3.9.4.1 Using DMAC Transfer Process

The DMAC transfer process is as follows.

**Step 1** Request DMA channel, and check if the DMA channel is idle by checking if it is enabled. A disabled channel indicates it is idle, while an enabled channel indicates it is busy.

**Step 2** Write the descriptor with 6 words into the memory. The descriptor must be word-aligned. For more details, refer to section 3.9.3.4 “[DMA Descriptor](#)”.

**Step 3** Write the start address of the descriptor to [DMA\\_DESC\\_ADDR\\_REGN](#).

**Step 4** Enable the DMA channel, and write the corresponding channel to [DMAC\\_EN\\_REGN](#).

**Step 5** The DMA obtains the descriptor information.

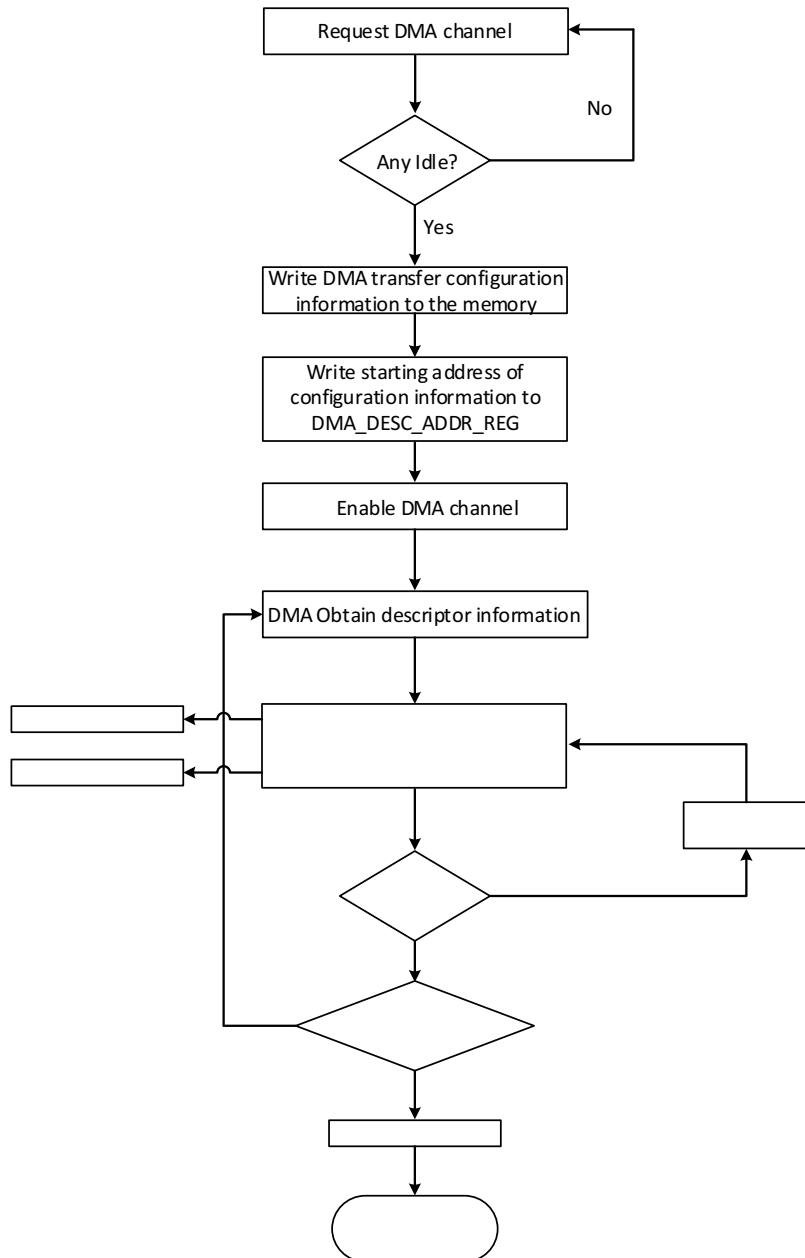
**Step 6** Start to transmit a package. When half of the package is completed, the DMA sends a Half Package Transfer Interrupt; when a total package is completed, the DMA sends a Package End Transfer Interrupt. These interrupt status can be read by [DMAC\\_IRQ\\_PEND\\_REG0](#).

**Step 7** Set [DMAC\\_PAU\\_REGN](#) to pause or resume the data transfer.

**Step 8** After completing a total package transfer, the DMA decides to start the next package transfer or end the transfer by the link of the descriptor. If the link is 0xFFFFF800, the transfer ends; otherwise, the next package starts to transmit. When the transfer ends, the DMA sends a Queue End Transfer Interrupt.

**Step 9** Disable the DMA channel.

**Figure 3-24 DMAC Transfer Process**



#### 3.9.4.2 Processing DMAC Interrupt

Follow the steps below to process the DMAC interrupt:

**Step 1** Enable interrupt: write the corresponding interrupt enable bit of [DMAC IRQ\\_EN\\_REG0](#). The system generates an interrupt when the corresponding condition is satisfied.

**Step 2** After entering the interrupt process, write <C:\Users\chenxiaofang\AppData\Roaming\Microsoft\Word\ Hlk52357013 - Hlk52357103> to clear the interrupt pending and execute the process of waiting for the interrupt.

**Step 3** Resume the interrupt and continue to execute the interrupted process.

### 3.9.4.3 Configuring DMAC

To configure the DMAC, follow the guidelines below:

- Make sure the transfer bit width of IO devices is consistent with the offset of the start address.
- The MBUS protocol does not support the read operation of non-integer words. For the devices whose bit width is not word-aligned, after receiving the read command, they should resolve the read command according to their FIFO bit width instead of the command bit width, and ignore the redundant data caused by the inconsistency of the bit width.
- When the DMA transfer is paused, this is equivalent to invalid DRQ. Because there is a certain time delay between DMA transfer commands, the DMAC will not stop data transfer until the DMAC finishes processing the current command and the commands in Arbiter (at most 32 bytes data).

DMAC application example:

```
writel(0x00000000, mem_address + 0x00); //Set configurations. The mem_address must be word-aligned.  
writel(0x00001000, mem_address + 0x04); // Set the start address for the source device.  
writel(0x20000000, mem_address + 0x08); //Set the start address for the destination device.  
writel(0x00000020, mem_address + 0x0C); // Set the data package size.  
writel(0x00000000, mem_address + 0x10); //Set the parameters.  
writel(0xFFFFF800, mem_address + 0x14); //Set the start address for the next descriptor.  
writel(mem_address, 0x01C02000+ 0x100 + 0x08); //Set the start address for the DMA channel0 descriptor.  
  
do{  
    If(mem_address == readl(0x01C02000 + 0x100 + 0x08));  
    break;  
}while(1); //Make sure that the writing operation is valid.  
writel(0x00000001, 0x01C02000 + 0x100 + 0x00); // Enable DMA channel0 transfer.
```

The DMAC supports increasing data package in transfer, pay attention to the following points:

- The 0xFFFFF800 value of [DMAC\\_FDESC\\_ADDR\\_REGN](#) indicates that the DMA channel has got back the descriptor of the last package. The DMA channel will automatically stop the data transfer after transferring the current package.
- To add a package during the data transfer, check if the DMA channel has got back the descriptor of the last package. If yes, you cannot add any package in the current queue. Request another DMA channel with a new DRQ to transfer the package. Otherwise, you can add the package by modifying the [DMAC\\_FDESC\\_ADDR\\_REGN](#) of the last package from 0xFFFFF800 to the start address of the to-be-added package.

To ensure that the modification is valid, read the value of [DMAC\\_FDESC\\_ADDR\\_REGN](#) after the modification. The value 0xFFFFF800 indicates the modification fails and the other values indicate you have successfully added packages to the queue.

Another problem is, the system needs some time to process the modification, during which the DMA channel may get back the descriptor of the last package. You can read [DMAC\\_CUR\\_SRC\\_REGN](#) and [DMAC\\_CUR\\_DEST\\_REGN](#) and check if the increasing memory address accords with the information of the added package. If yes, the package is added successfully; otherwise, the modification failed.

To ensure a higher rate of success, it is suggested that you add the package before the half package interrupt of the penultimate package.

### 3.9.5 Register List

Module Name	Base Address
DMAC	0x03002000

Register Name	Offset	Description
DMAC IRQ_EN_REG0	0x0000	DMAC IRQ Enable Register 0
DMAC IRQ_EN_REG1	0x0004	DMAC IRQ Enable Register 1
DMAC IRQ_PEND_REG0	0x0010	DMAC IRQ Pending Register 0
DMAC IRQ_PEND_REG1	0x0014	DMAC IRQ Pending Register 1
DMAC_SEC_REG	0x0020	DMAC Security Register
DMAC_AUTO_GATE_REG	0x0028	DMAC Auto Gating Register
DMAC_STA_REG	0x0030	DMAC Status Register
DMAC_EN_REGN	0x0100 + N*0x0040	DMAC Channel Enable Register N (N = 0 to 15)
DMAC_PAU_REGN	0x0104 + N*0x0040	DMAC Channel Pause Register N (N = 0 to 15)
DMAC_DESC_ADDR_REGN	0x0108 + N*0x0040	DMAC Channel Start Address Register N (N = 0 to 15)
DMAC_CFG_REGN	0x010C + N*0x0040	DMAC Channel Configuration Register N (N = 0 to 15)

Register Name	Offset	Description
DMAC_CUR_SRC_REGN	0x0110 + N*0x0040	DMAC Channel Current Source Register N (N = 0 to 15)
DMAC_CUR_DEST_REGN	0x0114 + N*0x0040	DMAC Channel Current Destination Register N (N = 0 to 15)
DMAC_BCNT_LEFT_REGN	0x0118 + N*0x0040	DMAC Channel Byte Counter Left Register N (N = 0 to 15)
DMAC_PARA_REGN	0x011C + N*0x0040	DMAC Channel Parameter Register N (N = 0 to 15)
DMAC_MODE_REGN	0x0128 + N*0x0040	DMAC Mode Register N (N = 0 to 15)
DMAC_FDESC_ADDR_REGN	0x012C + N*0x0040	DMAC Former Descriptor Address Register N (N = 0 to 15)
DMAC_PKG_NUM_REGN	0x0130 + N*0x0040	DMAC Package Number Register N (N = 0 to 15)

### 3.9.6 Register Description

#### 3.9.6.1 0x0000 DMAC IRQ Enable Register0 (Default Value: 0x0000\_0000)

Offset: 0x0000			Register Name: DMAC_IRQ_EN_REG0
Bit	Read/Write	Default/Hex	Description
31	/	/	/
30	R/W	0x0	DMA7_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 7 0: Disabled 1: Enabled
29	R/W	0x0	DMA7_PKG_IRQ_EN Enable the package end interrupt of DMA channel 7 0: Disabled 1: Enabled
28	R/W	0x0	DMA7_HLAF_IRQ_EN Enable the half package interrupt of DMA channel 7 0: Disabled 1: Enabled
27	/	/	/
26	R/W	0x0	DMA6_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 6 0: Disabled 1: Enabled

Offset: 0x0000			Register Name: DMAC_IRQ_EN_REG0
Bit	Read/Write	Default/Hex	Description
25	R/W	0x0	DMA6_PKG_IRQ_EN Enable the package end interrupt of DMA channel 6 0: Disabled 1: Enabled
24	R/W	0x0	DMA6_HLAF_IRQ_EN Enable the half package interrupt of DMA channel 6 0: Disabled 1: Enabled
23	/	/	/
22	R/W	0x0	DMA5_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 5 0: Disabled 1: Enabled
21	R/W	0x0	DMA5_PKG_IRQ_EN Enable the package end interrupt of DMA channel 5 0: Disabled 1: Enabled
20	R/W	0x0	DMA5_HLAF_IRQ_EN Enable the half package interrupt of DMA channel 5 0: Disabled 1: Enabled
19	/	/	/
18	R/W	0x0	DMA4_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 4 0: Disabled 1: Enabled
17	R/W	0x0	DMA4_PKG_IRQ_EN Enable the package end interrupt of DMA channel 4 0: Disabled 1: Enabled
16	R/W	0x0	DMA4_HLAF_IRQ_EN Enable the half package interrupt of DMA channel 4 0: Disabled 1: Enabled
15	/	/	/

Offset: 0x0000			Register Name: DMAC_IRQ_EN_REG0
Bit	Read/Write	Default/Hex	Description
14	R/W	0x0	DMA3_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 3 0: Disabled 1: Enabled
13	R/W	0x0	DMA3_PKG_IRQ_EN Enable the package end interrupt of DMA channel 3 0: Disabled 1: Enabled
12	R/W	0x0	DMA3_HLAF_IRQ_EN Enable the half package interrupt of DMA channel 3 0: Disabled 1: Enabled
11	/	/	/
10	R/W	0x0	DMA2_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 2 0: Disabled 1: Enabled
9	R/W	0x0	DMA2_PKG_IRQ_EN Enable the package end interrupt of DMA channel 2 0: Disabled 1: Enabled
8	R/W	0x0	DMA2_HLAF_IRQ_EN Enable the half package interrupt of DMA channel 2 0: Disabled 1: Enabled
7	/	/	/
6	R/W	0x0	DMA1_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 1 0: Disabled 1: Enabled
5	R/W	0x0	DMA1_PKG_IRQ_EN Enable the package end interrupt of DMA channel 1 0: Disabled 1: Enabled

Offset: 0x0000			Register Name: DMAC_IRQ_EN_REG0
Bit	Read/Write	Default/Hex	Description
4	R/W	0x0	DMA1_HLAF_IRQ_EN Enable the half package interrupt of DMA channel 1 0: Disabled 1: Enabled
3	/	/	/
2	R/W	0x0	DMA0_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 0 0: Disabled 1: Enabled
1	R/W	0x0	DMA0_PKG_IRQ_EN Enable the package end interrupt of DMA channel 0 0: Disabled 1: Enabled
0	R/W	0x0	DMA0_HLAF_IRQ_EN Enable the half package interrupt of DMA channel 0 0: Disabled 1: Enabled

### 3.9.6.2 0x0004 DMAC IRQ Enable Register1 (Default Value: 0x0000\_0000)

Offset: 0x0004			Register Name: DMAC_IRQ_EN_REG1
Bit	Read/Write	Default/Hex	Description
31	/	/	/
30	R/W	0x0	DMA15_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 15 0: Disabled 1: Enabled
29	R/W	0x0	DMA15_PKG_IRQ_EN Enable the package end interrupt of DMA channel 15 0: Disabled 1: Enabled
28	R/W	0x0	DMA15_HALF_IRQ_EN Enable the half package interrupt of DMA channel 15 0: Disabled 1: Enabled

Offset: 0x0004			Register Name: DMAC_IRQ_EN_REG1
Bit	Read/Write	Default/Hex	Description
27	/	/	/
26	R/W	0x0	DMA14_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 14 0: Disabled 1: Enabled
25	R/W	0x0	DMA14_PKG_IRQ_EN Enable the package end interrupt of DMA channel 14 0: Disabled 1: Enabled
24	R/W	0x0	DMA14_HALF_IRQ_EN Enable the half package interrupt of DMA channel 14 0: Disabled 1: Enabled
23	/	/	/
22	R/W	0x0	DMA13_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 13 0: Disabled 1: Enabled
21	R/W	0x0	DMA13_PKG_IRQ_EN Enable the package end interrupt of DMA channel 13 0: Disabled 1: Enabled
20	R/W	0x0	DMA13_HALF_IRQ_EN Enable the half package interrupt of DMA channel 13 0: Disabled 1: Enabled
19	/	/	/
18	R/W	0x0	DMA12_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 12 0: Disabled 1: Enabled
17	R/W	0x0	DMA12_PKG_IRQ_EN Enable the package end interrupt of DMA channel 12 0: Disabled 1: Enabled

Offset: 0x0004			Register Name: DMAC_IRQ_EN_REG1
Bit	Read/Write	Default/Hex	Description
16	R/W	0x0	DMA12_HALF_IRQ_EN Enable the half package interrupt of DMA channel 12 0: Disabled 1: Enabled
15	/	/	/
14	R/W	0x0	DMA11_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 11 0: Disabled 1: Enabled
13	R/W	0x0	DMA11_PKG_IRQ_EN Enable the package end interrupt of DMA channel 11 0: Disabled 1: Enabled
12	R/W	0x0	DMA11_HALF_IRQ_EN Enable the half package interrupt of DMA channel 11 0: Disabled 1: Enabled
11	/	/	/
10	R/W	0x0	DMA10_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 10 0: Disabled 1: Enabled
9	R/W	0x0	DMA10_PKG_IRQ_EN Enable the package end interrupt of DMA channel 10 0: Disabled 1: Enabled
8	R/W	0x0	DMA10_HALF_IRQ_EN Enable the half package interrupt of DMA channel 10 0: Disabled 1: Enabled
7	/	/	/
6	R/W	0x0	DMA9_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 9 0: Disabled 1: Enabled

Offset: 0x0004			Register Name: DMAC_IRQ_EN_REG1
Bit	Read/Write	Default/Hex	Description
5	R/W	0x0	DMA9_PKG_IRQ_EN Enable the package end interrupt of DMA channel 9 0: Disabled 1: Enabled
4	R/W	0x0	DMA9_HALF_IRQ_EN Enable the half package interrupt of DMA channel 9 0: Disabled 1: Enabled
3	/	/	/
2	R/W	0x0	DMA8_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 8 0: Disabled 1: Enabled
1	R/W	0x0	DMA8_PKG_IRQ_EN Enable the package end interrupt of DMA channel 8 0: Disabled 1: Enabled
0	R/W	0x0	DMA8_HALF_IRQ_EN Enable the half package interrupt of DMA channel 8 0: Disabled 1: Enabled

### 3.9.6.3 0x0010 DMAC IRQ Pending Status Register 0 (Default Value: 0x0000\_0000)

Offset: 0x0010			Register Name: DMAC_IRQ_PEND_REG0
Bit	Read/Write	Default/Hex	Description
31	/	/	/
30	R/W1C	0x0	DMA7_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 7. Write 1 to clear the pending status. 0: No effect 1: Pending

Offset: 0x0010			Register Name: DMAC_IRQ_PEND_REG0
Bit	Read/Write	Default/Hex	Description
29	R/W1C	0x0	DMA7_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 7. Write 1 to clear the pending status. 0: No effect 1: Pending
28	R/W1C	0x0	DMA7_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 7. Write 1 to clear the pending status. 0: No effect 1: Pending
27	/	/	/
26	R/W1C	0x0	DMA6_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 6. Write 1 to clear the pending status. 0: No effect 1: Pending
25	R/W1C	0x0	DMA6_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 6. Write 1 to clear the pending status. 0: No effect 1: Pending
24	R/W1C	0x0	DMA6_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 6. Write 1 to clear the pending status. 0: No effect 1: Pending
23	/	/	/
22	R/W1C	0x0	DMA5_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 5. Write 1 to clear the pending status. 0: No effect 1: Pending
21	R/W1C	0x0	DMA5_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 5. Write 1 to clear the pending status. 0: No effect 1: Pending

Offset: 0x0010			Register Name: DMAC_IRQ_PEND_REG0
Bit	Read/Write	Default/Hex	Description
20	R/W1C	0x0	DMA5_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 5. Write 1 to clear the pending status. 0: No effect 1: Pending
19	/	/	/
18	R/W1C	0x0	DMA4_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 4. Write 1 to clear the pending status. 0: No effect 1: Pending
17	R/W1C	0x0	DMA4_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 4. Write 1 to clear the pending status. 0: No effect 1: Pending
16	R/W1C	0x0	DMA4_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 4. Write 1 to clear the pending status. 0: No effect 1: Pending
15	/	/	/
14	R/W1C	0x0	DMA3_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 3. Write 1 to clear the pending status. 0: No effect 1: Pending
13	R/W1C	0x0	DMA3_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 3. Write 1 to clear the pending status. 0: No effect 1: Pending
12	R/W1C	0x0	DMA3_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 3. Write 1 to clear the pending status. 0: No effect 1: Pending

Offset: 0x0010			Register Name: DMAC_IRQ_PEND_REG0
Bit	Read/Write	Default/Hex	Description
11	/	/	/
10	R/W1C	0x0	DMA2_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 2. Write 1 to clear the pending status. 0: No effect 1: Pending
9	R/W1C	0x0	DMA2_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 2. Write 1 to clear the pending status. 0: No effect 1: Pending
8	R/W1C	0x0	DMA2_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 2. Write 1 to clear the pending status. 0: No effect 1: Pending
7	/	/	/
6	R/W1C	0x0	DMA1_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 1. Write 1 to clear the pending status. 0: No effect 1: Pending
5	R/W1C	0x0	DMA1_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 1. Write 1 to clear the pending status. 0: No effect 1: Pending
4	R/W1C	0x0	DMA1_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 1. Write 1 to clear the pending status. 0: No effect 1: Pending
3	/	/	/

Offset: 0x0010			Register Name: DMAC_IRQ_PEND_REG0
Bit	Read/Write	Default/Hex	Description
2	R/W1C	0x0	<p>DMA0_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 0. Write 1 to clear the pending status.</p> <p>0: No effect 1: Pending</p>
1	R/W1C	0x0	<p>DMA0_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 0. Write 1 to clear the pending status.</p> <p>0: No effect 1: Pending</p>
0	R/W1C	0x0	<p>DMA0_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 0. Write 1 to clear the pending status.</p> <p>0: No effect 1: Pending</p>

### 3.9.6.4 0x0014 DMAC IRQ Pending Status Register 1 (Default Value: 0x0000\_0000)

Offset: 0x0010			Register Name: DMAC_IRQ_PEND_REG0
Bit	Read/Write	Default/Hex	Description
31	/	/	/
30	R/W1C	0x0	<p>DMA15_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 15. Write 1 to clear the pending status.</p> <p>0: No effect 1: Pending</p>
29	R/W1C	0x0	<p>DMA15_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 15. Write 1 to clear the pending status.</p> <p>0: No effect 1: Pending</p>
28	R/W1C	0x0	<p>DMA15_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 15. Write 1 to clear the pending status.</p> <p>0: No effect 1: Pending</p>

Offset: 0x0010			Register Name: DMAC_IRQ_PEND_REG0
Bit	Read/Write	Default/Hex	Description
27	/	/	/
26	R/W1C	0x0	DMA14_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 14. Write 1 to clear the pending status. 0: No effect 1: Pending
25	R/W1C	0x0	DMA14_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 14. Write 1 to clear the pending status. 0: No effect 1: Pending
24	R/W1C	0x0	DMA14_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 14. Write 1 to clear the pending status. 0: No effect 1: Pending
23	/	/	/
22	R/W1C	0x0	DMA13_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 13. Write 1 to clear the pending status. 0: No effect 1: Pending
21	R/W1C	0x0	DMA13_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 13. Write 1 to clear the pending status. 0: No effect 1: Pending.
20	R/W1C	0x0	DMA13_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 13. Write 1 to clear the pending status. 0: No effect 1: Pending
19	/	/	/

Offset: 0x0010			Register Name: DMAC_IRQ_PEND_REG0
Bit	Read/Write	Default/Hex	Description
18	R/W1C	0x0	DMA12_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 12. Write 1 to clear the pending status. 0: No effect 1: Pending
17	R/W1C	0x0	DMA12_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 12. Write 1 to clear the pending status. 0: No effect 1: Pending
16	R/W1C	0x0	DMA12_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 12. Write 1 to clear the pending status. 0: No effect 1: Pending
15	/	/	/
14	R/W1C	0x0	DMA11_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 11. Write 1 to clear the pending status. 0: No effect 1: Pending
13	R/W1C	0x0	DMA11_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 11. Write 1 to clear the pending status. 0: No effect 1: Pending
12	R/W1C	0x0	DMA11_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 11. Write 1 to clear the pending status. 0: No effect 1: Pending
11	/	/	/
10	R/W1C	0x0	DMA10_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 10. Write 1 to clear the pending status. 0: No effect 1: Pending

Offset: 0x0010			Register Name: DMAC_IRQ_PEND_REG0
Bit	Read/Write	Default/Hex	Description
9	R/W1C	0x0	DMA10_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 10. Write 1 to clear the pending status. 0: No effect 1: Pending
8	R/W1C	0x0	DMA10_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 10. Write 1 to clear the pending status. 0: No effect 1: Pending
7	/	/	/
6	R/W1C	0x0	DMA9_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 9. Write 1 to clear the pending status. 0: No effect 1: Pending
5	R/W1C	0x0	DMA9_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 9. Write 1 to clear the pending status. 0: No effect 1: Pending
4	R/W1C	0x0	DMA9_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 9. Write 1 to clear the pending status. 0: No effect 1: Pending
3	/	/	/
2	R/W1C	0x0	DMA8_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 8. Write 1 to clear the pending status. 0: No effect 1: Pending
1	R/W1C	0x0	DMA8_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 8. Write 1 to clear the pending status. 0: No effect 1: Pending

Offset: 0x0010			Register Name: DMAC_IRQ_PEND_REG0
Bit	Read/Write	Default/Hex	Description
0	R/W1C	0x0	<p>DMA8_HALF_IRQ_PEND</p> <p>The IRQ pending bit for the half package interrupt of the DMA channel 8. Write 1 to clear the pending status.</p> <p>0: No effect 1: Pending</p>

### 3.9.6.5 0x0020 DMAC Security Register (Default Value: 0x0000\_FFFF)

Offset: 0x0020			Register Name: DMAC_SEC_REG
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xFF	<p>DMA_SEC</p> <p>Configure the security mode for DMA channel[15:0]</p> <p>The meaning of each bit:</p> <p>0: Secure mode 1: Non-secure mode</p>

### 3.9.6.6 0x0028 DMAC Auto Gating Register (Default Value: 0x0000\_0000)

Offset: 0x0028			Register Name: DMAC_AUTO_GATE_REG
Bit	Read/Write	Default/Hex	Description
31:3	/	/	/
2	R/W	0x0	<p>DMA_MCLK_CIRCUIT</p> <p>Auto gating bit of DMA MCLK interface circuit</p> <p>0: Auto gating enabled 1: Auto gating disabled</p>
1	R/W	0x0	<p>DMA_COMMON_CIRCUIT</p> <p>Auto gating bit of DMA common circuit</p> <p>0: Auto gating enabled 1: Auto gating disabled</p>
0	R/W	0x0	<p>DMA_CHAN_CIRCUIT</p> <p>Auto gating bit of DMA channel circuit</p> <p>0: Auto gating enabled 1: Auto gating disabled</p>



When initializing the DMA Controller, the bit[2] should be set up.

### 3.9.6.7 0x0030 DMAC Status Register (Default Value: 0x0000\_0000)

Offset: 0x0030			Register Name: DMAC_STA_REG
Bit	Read/Write	Default/Hex	Description
31	R	0x0	MBUS FIFO Status 0: Empty 1: Not Empty
30:16	/	/	/
15:0	R	0x0	DMA_STATUS DMA Channel[15:0] Status The meaning of each bit: 0: Idle 1: Busy

### 3.9.6.8 0x0100 + N\*0x0040 DMAC Channel Enable Register N (Default Value: 0x0000\_0000)

Offset: 0x0100 + N*0x0040 (N = 0 to 15)			Register Name: DMAC_EN_REGN
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	DMA_EN DMA Channel Enable 0: Disabled 1: Enabled

### 3.9.6.9 0x0104 + N\*0x0040 DMAC Channel Pause Register N (Default Value: 0x0000\_0000)

Offset: 0x0104 + N*0x0040 (N = 0 to 15)			Register Name: DMAC_PAU_REGN
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	DMA_PAUSE Pause the DMA Channel Transfer Data 0: Resume Transferring 1: Pause Transferring

### 3.9.6.10 0x0108 + N\*0x0040 DMAC Channel Descriptor Address Register N (Default Value: 0x0000\_0000)

Offset: 0x0108 + N*0x0040 (N = 0 to 15)			Register Name: DMAC_DESC_ADDR_REGN
Bit	Read/Write	Default/Hex	Description
31:2	R/W	0x0	DMA_DESC_ADDR Lower 30 bits of DMA channel descriptor address The descriptor address must be word-aligned.
1:0	R/W	0x0	DMA_DESC_HIGH_ADDR Higher 2 bits of DMA channel descriptor high address The real address is as follows. DMA Channel Descriptor Address = {bit[1:0], bit[31:2], 2'b00}

### 3.9.6.11 0x010C + N\*0x0040 DMAC Channel Configuration Register N (Default Value: 0x0000\_0000)

Offset: 0x010C + N*0x0040 (N = 0 to 15)			Register Name: DMAC_CFG_REGN
Bit	Read/Write	Default/Hex	Description
31	/	/	/
30	R	0x0	BMODE_SEL 0: Normal Mode 1: BMODE
29:27	/	/	/

Offset: 0x010C + N*0x0040 (N = 0 to 15)			Register Name: DMAC_CFG_REGN
Bit	Read/Write	Default/Hex	Description
26:25	R	0x0	DMA_DEST_DATA_WIDTH DMA Destination Data Width 00: 8-bit 01: 16-bit 10: 32-bit 11: 64-bit
24	R	0x0	DMA_ADDR_MODE DMA Destination Address Mode 0: Linear Mode 1: IO Mode
23:22	R	0x0	DMA_DEST_BLOCK_SIZE DMA Destination Block Size 00: 1 01: 4 10: 8 11: 16
21:16	R	0x0	DMA_DEST_DRQ_TYPE DMA Destination DRQ Type The details in DRQ Type and Port Corresponding Relation.
15:11	/	/	/
10:9	R	0x0	DMA_SRC_DATA_WIDTH DMA Source Data Width 00: 8-bit 01: 16-bit 10: 32-bit 11: 64-bit
8	R	0x0	DMA_SRC_ADDR_MODE DMA Source Address Mode 0: Linear Mode 1: IO Mode

<b>Offset: 0x010C + N*0x0040 (N = 0 to 15)</b>			<b>Register Name: DMAC_CFG_REGN</b>
<b>Bit</b>	<b>Read/Write</b>	<b>Default/Hex</b>	<b>Description</b>
7:6	R	0x0	DMA_SRC_BLOCK_SIZE DMA Source Block Size 00: 1 01: 4 10: 8 11: 16
5:0	R	0x0	DMA_SRC_DRQ_TYPE DMA Source DRQ Type The details in DRQ Type and Port Corresponding Relation.

### 3.9.6.12 0x0110 + N\*0x0040 DMAC Channel Current Source Address Register N (Default Value: 0x0000\_0000)

<b>Offset: 0x0110 + N*0x0040 (N = 0 to 15)</b>			<b>Register Name: DMAC_CUR_SRC_REGN</b>
<b>Bit</b>	<b>Read/Write</b>	<b>Default/Hex</b>	<b>Description</b>
31:0	R	0x0	DMA_CUR_SRC DMA Channel Current Source Address.

### 3.9.6.13 0x0114 + N\*0x0040 DMAC Channel Current Destination Address Register N (Default Value: 0x0000\_0000)

<b>Offset: 0x0114 + N*0x0040 (N = 0 to 15)</b>			<b>Register Name: DMAC_CUR_DEST_REGN</b>
<b>Bit</b>	<b>Read/Write</b>	<b>Default/Hex</b>	<b>Description</b>
31:0	R	0x0	DMA_CUR_DEST DMA Channel Current Destination Address.

### 3.9.6.14 0x0118 + N\*0x0040 DMAC Channel Byte Counter Left Register N (Default Value: 0x0000\_0000)

<b>Offset: 0x0118 + N*0x0040 (N = 0 to 15)</b>			<b>Register Name: DMAC_BCNT_LEFT_REGN</b>
<b>Bit</b>	<b>Read/Write</b>	<b>Default/Hex</b>	<b>Description</b>
31:25	/	/	/

<b>Offset: 0x0118 + N*0x0040 (N = 0 to 15)</b>			<b>Register Name: DMAC_BCNT_LEFT_REGN</b>
<b>Bit</b>	<b>Read/Write</b>	<b>Default/Hex</b>	<b>Description</b>
24:0	R	0x0	DMA_BCNT_LEFT DMA Channel Byte Counter Left.

### 3.9.6.15 0x011C + N\*0x0040 DMAC Channel Parameter Register N (Default Value: 0x0000\_0000)

<b>Offset: 0x011C + N*0x0040 (N = 0 to 15)</b>			<b>Register Name: DMAC_PARA_REGN</b>
<b>Bit</b>	<b>Read/Write</b>	<b>Default/Hex</b>	<b>Description</b>
31:8	/	/	/
7:0	R	0x0	WAIT_CYC Wait Clock Cycles

### 3.9.6.16 0x0128 + N\*0x0040 DMAC Mode Register N (Default Value: 0x0000\_0000)

<b>Offset: 0x0128 + N*0x0040 (N = 0 to 15)</b>			<b>Register Name: DMAC_MODE_REGN</b>
<b>Bit</b>	<b>Read/Write</b>	<b>Default/Hex</b>	<b>Description</b>
31:4	/	/	/
3	R/W	0x0	DMA_DST_MODE Destination Communication Mode Select 0: Waiting mode 1: Handshake mode
2	R/W	0x0	DMA_SRC_MODE Source Communication Mode Select 0: Waiting mode 1: Handshake mode
1:0	/	/	/

**3.9.6.17 0x012C + N\*0x0040 DMAC Former Descriptor Address Register N (Default Value: 0x0000\_0000)**

Offset: 0x012C + N*0x0040 (N = 0 to 15)			Register Name: DMAC_FDESC_ADDR_REGN
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	DMA_FDESC_ADDR This register is used to store the former value of DMA Channel Descriptor Address Register.

**3.9.6.18 0x0130 + N\*0x0040 DMAC Package Number Register N (Default Value: 0x0000\_0000)**

Offset: 0x0130 + N*0x0040 (N = 0 to 15)			Register Name: DMAC_PKG_NUM_REGN
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	DMA_PKG_NUM This register records the number of packages which has been completed in one transmission.

## 3.10 Thermal Sensor Controller (THS)

### 3.10.1 Overview

The thermal sensors are common elements in wide range of modern system on chips (SoCs) platform. The thermal sensors are used to constantly monitor the temperature on the chip.

The thermal sensor controller (THS) embeds one thermal sensor located in the CPU. When the temperature reaches a certain thermal threshold, the thermal sensor can generate interrupts to the software to lower the temperature via the dynamic voltage and frequency scaling (DVFS) technology.

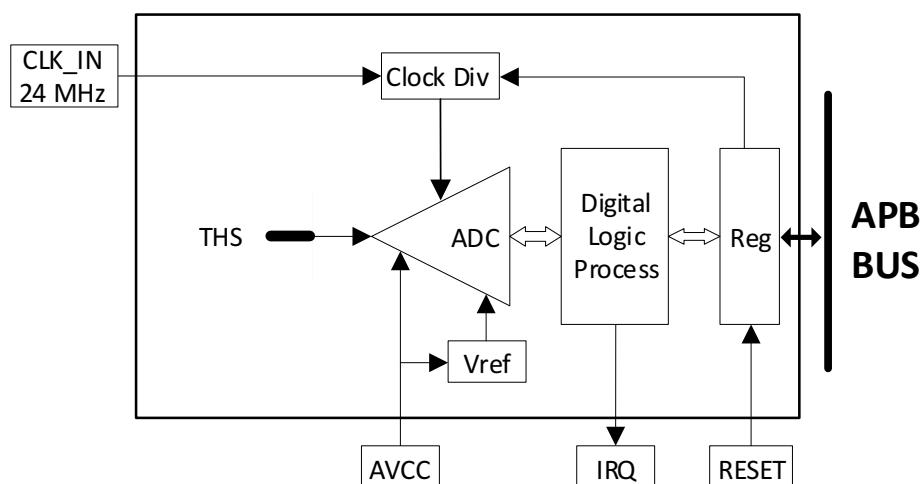
The THS has the following features:

- Temperature accuracy:  $\pm 3^{\circ}\text{C}$  from  $0^{\circ}\text{C}$  to  $+100^{\circ}\text{C}$ ,  $\pm 5^{\circ}\text{C}$  from  $-25^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$
- Averaging filter for thermal sensor reading
- Supports over-temperature protection interrupt and over-temperature alarm interrupt

### 3.10.2 Block Diagram

The following figure shows a block diagram of the THS.

**Figure 3-25 THS Block Diagram**



### 3.10.3 Functional Description

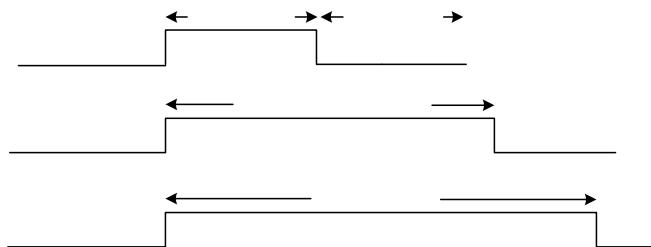
#### 3.10.3.1 Clock Source

The THS gets one clock source: OSC24M. For details about clock configurations, refer to section 3.3 “[CCU](#)”.

### 3.10.3.2 Timing Requirements

The following figure shows the timing requirements for the THS.

**Figure 3-26 Thermal Sensor Timing Requirement**



CLK\_IN = 24 MHz

CONV\_TIME (Conversion Time) =  $1/24 \text{ MHz} \times 14 \text{ Cycles} = 0.583 \text{ us}$

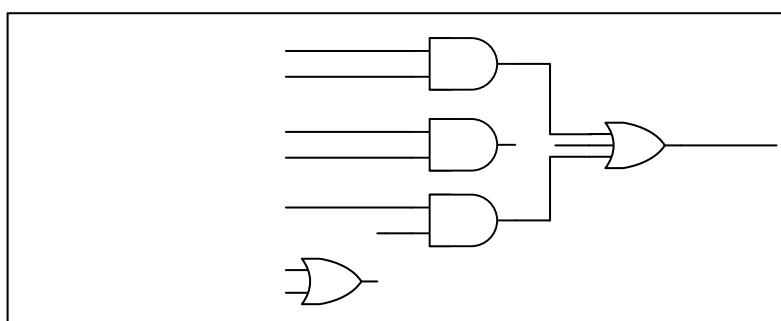
TACQ >  $1/24 \text{ MHz} \times 24 \text{ Cycles}$

THERMAL\_PER > ADC\_Sample\_Frequency > TACQ + CONV\_TIME

### 3.10.3.3 Interrupts

The THS has four interrupt sources: DATA\_IRQ, SHUTDOWN\_IRQ, ALARM\_IRQ, and ALARM\_OFF\_IRQ. The following figure shows thermal sensor interrupt sources.

**Figure 3-27 Thermal Sensor Controller Interrupt Source**



**DATA\_IRQ:** The interrupt is generated when the measured sensor\_data is updated.

**SHUTDOWN\_IRQ:** The interrupt is generated when the temperature is higher than the shutdown threshold.

**ALARM\_IRQ:** The interrupt is generated when the temperature is higher than the Alarm\_Threshold.

**ALARM\_OFF\_IRQ:** The interrupt is generated when the temperature drops to lower than the Alarm\_Off\_Threshold. It is triggered at the fall edge.

### 3.10.3.4 THS Temperature Conversion Formula

$$T = (\text{sensor\_data} - 2800) / (-14.85)$$

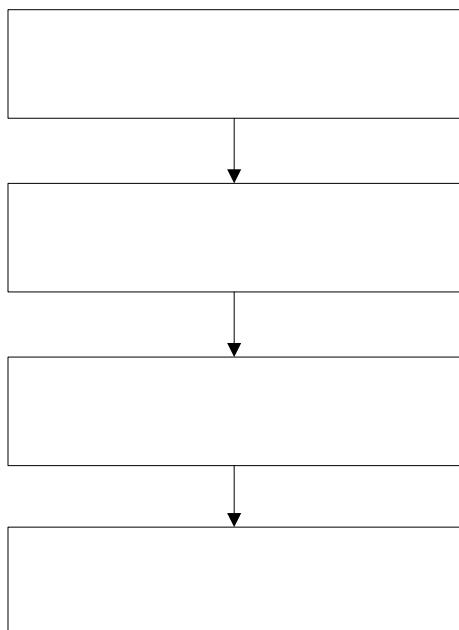
Unit of T: Celsius degree (°C).

The sensor\_data is read from the sensor data register.

### 3.10.4 Programming Guidelines

The initial process of the THS is as follows.

**Figure 3-28 THS Initial Process**



In the final test (FT) stage, the THS is calibrated through the ambient temperature, and the calibration value is written in the SID module. The following table shows the THS information in the SID.

**Table 3-16 THS Information in the SID**

Base Address: 0x14	Register Name: THS
Bit	Description
27:16	The calibration value of the T-sensor.

Before enabling THS, read eFuse value and write the value to [THS\\_CDATA](#).

### Query Mode

- Step 1** Write 0x1 to the bit[16] of [THS\\_BGR\\_REG](#) to dessert the reset.
- Step 2** Write 0x1 to the bit[0] of [THS\\_BGR\\_REG](#) to open the THS clock.
- Step 3** Write 0x2F to the bit[15:0] of [THS\\_CTRL](#) to set the ADC acquire time.
- Step 4** Write 0x1DF to the bit[31:16] of [THS\\_CTRL](#) to set the ADC sample frequency divider.
- Step 5** Write 0x3A to the bit[31:12] of [THS\\_PER](#) to set the THS work period.
- Step 6** Write 0x1 to the bit[2] of [THS\\_FILTER](#) to enable the temperature convert filter.
- Step 7** Write 0x1 to the bit[1:0] of [THS\\_FILTER](#) to select the filter type.
- Step 8** Read THS eFuse value from SID, then write the eFuse value to [THS\\_CDATA](#) to calibrate THS.
- Step 9** Write 0x1 to the bit[0] of [THSO\\_EN](#) to enable THS.
- Step 10** Read the bit[0] of [THS\\_DATA\\_INTS](#). If it is 1, the temperature conversion is complete.
- Step 11** Read the bit[11:0] of [THS\\_DATA](#), and calculate the THS temperature based on section 3.10.3.4 “[THS Temperature Conversion Formula](#)”.

### Interrupt Mode

- Step 1** Write 0x1 to the bit16 of [THS\\_BGR\\_REG](#) to dessert the reset.
- Step 2** Write 0x1 to the bit0 of [THS\\_BGR\\_REG](#) to open the THS clock.
- Step 3** Write 0x2F to the bit[15:0] of [THS\\_CTRL](#) to set the ADC acquire time.
- Step 4** Write 0x1DF to the bit[31:16] of [THS\\_CTRL](#) to set the ADC sample frequency divider.
- Step 5** Write 0x3A to the bit[31:12] of [THS\\_PER](#) to set the THS work period.
- Step 6** Write 0x1 to the bit2 of [THS\\_FILTER](#) to enable the temperature convert filter.
- Step 7** Write 0x1 to the bit[1:0] of [THS\\_FILTER](#) to select the filter type.
- Step 8** Read THS eFuse value from SID, and then write the eFuse value to [THS\\_CDATA](#) to calibrate THS.
- Step 9** Write 0x1 to the bit[0] of [THS\\_DATA\\_INTC](#) to enable the interrupt of THS.
- Step 10** Set GIC interface based on IRQ 90.
- Step 11** Put the interrupt handler address into the interrupt vector table.
- Step 12** Write 0x1 to the bit[0] of [THSO\\_EN](#) to enable THS.
- Step 13** Read the bit[0] of [THS\\_DATA\\_INTS](#). If it is 1, the temperature conversion is complete.
- Step 14** Read the bit[11:0] of [THS\\_DATA](#), and calculate the THS temperature based on section 3.10.3.4 “[THS Temperature Conversion Formula](#)”.

### 3.10.5 Register List

Module Name	Base Address
THS	0x02009400

Register Name	Offset	Description
THS_CTRL	0x0000	THS Control Register
THS_EN	0x0004	THS Enable Register
THS_PER	0x0008	THS Period Control Register
THS_DATA_INTC	0x0010	THS Data Interrupt Control Register
THS_SHUT_INTC	0x0014	THS Shut Interrupt Control Register
THS_ALARM_INTC	0x0018	THS Alarm Interrupt Control Register
THS_DATA_INTS	0x0020	THS Data Interrupt Status Register
THS_SHUT_INTS	0x0024	THS Shut Interrupt Status Register
THS_ALARMO_INTS	0x0028	THS Alarm off Interrupt Status Register
THS_ALARM_INTS	0x002C	THS Alarm Interrupt Status Register
THS_FILTER	0x0030	THS Median Filter Control Register
THS_ALARM_CTRL	0x0040	THS Alarm Threshold Control Register
THS_SHUTDOWN_CTRL	0x0080	THS Shutdown Threshold Control Register
THS_CDATA	0x00A0	THS Calibration Data
THS_DATA	0x00C0	THS Data Register

### 3.10.6 Register Description

#### 3.10.6.1 0x0000 THS Control Register (Default Value: 0x01DF\_002F)

Offset: 0x0000			Register Name: THS_CTRL
Bit	Read/Write	Default/Hex	Description
31:16	R/W	0x1DF	TACQ ADC acquire time CLK_IN/(n + 1) The default value is 2 us.
15:0	R/W	0x2F	Reserved

### 3.10.6.2 0x0004 THS Enable Register (Default Value: 0x0000\_0000)

Offset: 0x0004			Register Name: THS_EN
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	<p>THS_EN</p> <p>Enable temperature measurement sensor</p> <p>0: Disable</p> <p>1: Enable</p>

### 3.10.6.3 0x0008 THS Period Control Register (Default Value: 0x0003\_A000)

Offset: 0x0008			Register Name: THS_PER
Bit	Read/Write	Default/Hex	Description
31:12	R/W	0x3A	<p>THERMAL_PER</p> <p>Temperature measurement period</p> <p><math>4096*(n + 1)/CLK\_IN</math></p> <p>The default value is 10 ms.</p>
11:0	/	/	/

### 3.10.6.4 0x0010 THS Data Interrupt Control Register (Default Value: 0x0000\_0000)

Offset: 0x0010			Register Name: THS_DATA_INTC
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	<p>THS_DATA_IRQ_EN</p> <p>Enable the interrupt of sensor_data update</p> <p>If enabled, when the measured sensor_data is updated, it will generate an interrupt.</p> <p>0: Disabled</p> <p>1: Enabled</p>

**3.10.6.5 0x0014 THS Shut Interrupt Control Register (Default Value: 0x0000\_0000)**

Offset: 0x0014			Register Name: THS_SHUT_INTC
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	<p>SHUT_INT_EN</p> <p>Enable the shutdown interrupt for the sensor</p> <p>0: Disabled 1: Enabled</p>

**3.10.6.6 0x0018 THS Alarm Interrupt Control Register (Default Value: 0x0000\_0000)**

Offset: 0x0018			Register Name: THS_ALARM_INTC
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	<p>ALARM_INT_EN</p> <p>Enable the alarm interrupt for the sensor</p> <p>0: Disabled 1: Enabled</p>

**3.10.6.7 0x0020 THS Data Interrupt Status Register (Default Value: 0x0000\_0000)**

Offset: 0x0020			Register Name: THS_DATA_INTS
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W1C	0x0	<p>THS_DATA_IRQ_STS</p> <p>Indicates the pending status of the sensor's data interrupt. Write 1 to clear the pending status.</p> <p>0: No effect 1: Pending</p>

**3.10.6.8 0x0024 THS Shut Interrupt Status Register (Default Value: 0x0000\_0000)**

Offset: 0x0024			Register Name: THS_SHUT_INTS
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W1C	0x0	<p>SHUT_INT_STS</p> <p>Indicates the pending status of the sensor's shutdown interrupt.</p> <p>Write 1 to clear the pending status.</p> <p>0: No effect</p> <p>1: Pending</p>

**3.10.6.9 0x0028 THS Alarm Off Interrupt Status Register (Default Value: 0x0000\_0000)**

Offset: 0x0028			Register Name: THS_ALARM_INTS
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W1C	0x0	<p>ALARM_OFF_STS</p> <p>Alarm interrupt off pending for sensor</p> <p>Write 1 to clear the pending status.</p> <p>0: No effect</p> <p>1: Pending</p>

**3.10.6.10 0x002C THS Alarm Interrupt Status Register (Default Value: 0x0000\_0000)**

Offset: 0x002C			Register Name: THS_ALARM_INTS
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W1C	0x0	<p>ALARM_INT_STS</p> <p>Alarm interrupt pending for sensor</p> <p>Write 1 to clear the pending status.</p> <p>0: No effect</p> <p>1: Pending</p>

**3.10.6.11 0x0030 Median Filter Control Register (Default Value: 0x0000\_0001)**

Offset: 0x0030			Register Name: THS_FILTER
Bit	Read/Write	Default/Hex	Description
31:3	/	/	/
2	R/W	0x0	FILTER_EN Filter enable 0: Disabled 1: Enabled
1:0	R/W	0x1	FILTER_TYPE Averaging filter type 00: 2 01: 4 10: 8 11: 16

**3.10.6.12 0x0040 THS Alarm Threshold Control Register (Default Value: 0x05A0\_0684)**

Offset: 0x0040			Register Name: THS_ALARM_CTRL
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27:16	R/W	0x5A0	ALARM_T_HOT Thermal sensor alarm threshold for hot temperature
15:12	/	/	/
11:0	R/W	0x684	ALARM_T_HYST Thermal sensor alarm threshold for hysteresis temperature

**3.10.6.13 0x0080 THS Shutdown Threshold Control Register (Default Value: 0x0000\_04E9)**

Offset: 0x0080			Register Name: THS_SHUTDOWN_CTRL
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
11:0	R/W	0x4E9	SHUT_T_HOT Thermal sensor shutdown threshold for hot temperature

**3.10.6.14 0x00A0 THS Calibration Data Register (Default Value: 0x0000\_0800)**

Offset: 0x00A0			Register Name: THS_CDATA
Bit	Read/Write	Default/Hex	Description
31:12	/	/	/
11:0	R/W	0x800	THS_CDATA Thermal sensor calibration data

**3.10.6.15 0x00C0 THS Data Register (Default Value: 0x0000\_0000)**

Offset: 0x00C0			Register Name: THS_DATA
Bit	Read/Write	Default/Hex	Description
31:12	/	/	/
11:0	R	0x0	THS_DATA Temperature measurement data of sensor

## 3.11 IOMMU

### 3.11.1 Overview

The I/O Memory management unit (IOMMU) is designed for product specific memory requirements. It maps the virtual address (sent by peripheral access memory) to the physical address. The IOMMU allows multiple ways to manage the location of physical address, and it can use physical address which has potentially conflict mapping for different processes to allocate memory space, and also allow application of non-continuous address mapping to continuous virtual address space.

#### Features:

- Supports virtual address to physical address mapping by hardware implementation
- Supports VE, DE, G2D, DI parallel address mapping
- Supports VE, DE, G2D, DI bypass function independently
- Supports VE, DE, G2D, DI pre-fetch independently
- Supports VE, DE, G2D, DI interrupt handing mechanism independently
- Supports 2 levels TLB (level1 TLB for special using, and level2 TLB for sharing)
- Supports TLB Fully cleared and Partially disabled
- Supports trigger PTW access when TLB miss
- Supports checking the permission

### 3.11.2 Block Diagram

The internal module of IOMMU mainly has the following parts.

**Micro TLB:** level1 TLB, 64 words. Each peripheral corresponds to a TLB, which caching the level2 page table for the peripheral.

**Macro TLB:** level2 TLB, 4K words. Each peripheral shares a level2 TLB for caching the level2 page table.

**Prefetch Logic:** Each Micro TLB corresponds to a Prefetch Logic. By monitoring each master device to predict the bus access, the level2 page table corresponding to the address to be accessed can be read from memory and stored in the level2 TLB to improve hit ratio.

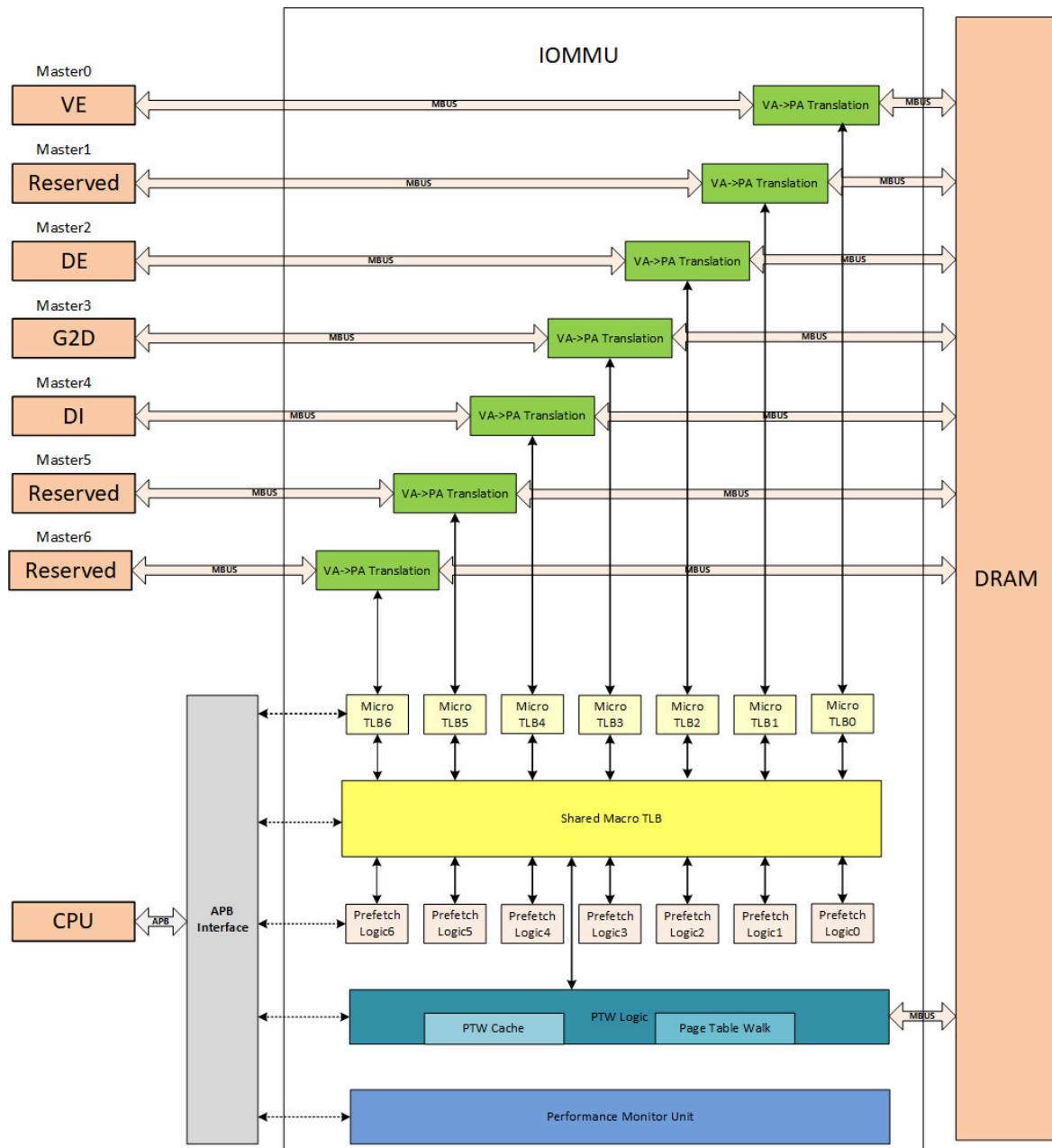
**PTW Logic:** Page Table Walk, mainly contains PTW Cache and PTW. The PTW Cache is used to store the level1 page table; when the virtual address VA missed in the level1 and level2 TLB, it will trigger the PTW. PTW Cache can store 512 level1 page tables, that is, 512 words.

**PMU:** Performance Monitoring Unit, which is used to count hit efficiency and latency.

**APB Interface:** IOMMU register instantiation module. CPU reads and writes the IOMMU register by APB bus.

The following figure shows the internal block diagram of IOMMU.

**Figure 3-29 IOMMU Block Diagram**



**Table 3-17 Correspondence Relation between Master and Module**

Master Number	Module
Master0	VE
Master1	Reserved
Master2	DE

Master Number	Module
Master3	G2D
Master4	DI
Master5	Reserved
Master6	Reserved

### 3.11.3 Functional Description

#### 3.11.3.1 Initialization

- Release the IOMMU reset signal by writing 1 to the bit[31] of the **IOMMU Reset Register**;
- Write the base address of the first TLB to the **IOMMU Translation Table Base Register**;
- Set the **IOMMU Interrupt Enable Register**;
- Enable the IOMMU by configuring the **IOMMU Enable Register** in the final.

#### 3.11.3.2 Address Changing

In the process of address mapping, The peripheral virtual address VA[31:12] are retrieved in the Level1 TLB, when TLB hits, the mapping finished, or they are retrieved in the Level2 TLB in the same way. If TLB hits, it will write the hit mapping to the Level1 TLB, and hits in Level1 TLB. If Level1 and Level2 TLB fail to be retrieved, it will trigger the PTW. After opening peripheral bypass function by setting IOMMU Bypass Register, IOMMU will not map the address, and it will output the virtual address as physical address. The typical scenarios are as follows.

##### Micro TLB hit

- Step 1** The master device sends a transfer command to send the address to the corresponding Micro TLB, and searches the Level2 page table corresponding to virtual address;
- Step 2** If Micro TLB hits, it will return a corresponding physical addresses and the Level2 page table of permission Index;
- Step 3** Address transform module converts the virtual address into physical address, and checks the permissions at the same time. If pass, the transfer is completed.

### Micro TLB miss, Macro TLB hit

- Step 1** The master device sends a transfer command to send the address to the corresponding Micro TLB, and searches the Level2 page table corresponding to virtual address;
- Step 2** If Micro TLB misses, then continue to search Macro TLB;
- Step 3** If Macro TLB hits, it will return the Level2 page table to Micro TLB;
- Step 4** Micro TLB receives the page table and puts it to Micro TLB (if this Micro TLB is full, there will happen the replace activity), at the same time, the page table entry is sent to address translation module;
- Step 5** Address transform module converts the virtual address into physical address, and checks the permissions at the same time. If pass, the transfer is completed.

### Micro TLB miss, Macro TLB miss, PTW Cache hit

- Step 1** The master device sends a transfer command to send the address to the corresponding Micro TLB, and searches the Level2 page table corresponding to virtual address;
- Step 2** If Micro TLB misses, then continue to search Macro TLB;
- Step 3** If Macro TLB misses, then it will send the request to the PTW to return the corresponding page table;
- Step 4** PTW first accesses PTW Cache, confirms that the required Level1 page table exists in the PTW Cache, sends the page table to PTW logic;
- Step 5** PTW logic returns the corresponding Level2 page table from memory page table according to Level1 page table, checks the effectiveness, and sends to Macro TLB;
- Step 6** Macro TLB stores the Level2 page table (there may happen the replace activity), and will return the Level2 page table to Micro TLB;
- Step 7** Micro TLB receives the page table entries and puts it to the Micro TLB (if this Micro TLB is full, there will happen the replace activity), and sends page table entries to address translation module;
- Step 8** Address transform module converts the virtual address into physical address, and checks the permissions at the same time. If pass, the transfer is completed.

### Micro TLB miss, Macro TLB miss, PTW Cache miss

- Step 1** The master device sends a transfer command to send the address to the corresponding Micro TLB, and searches the Level2 page table corresponding to virtual address;
- Step 2** If Micro TLB misses, then continue to search Macro TLB;
- Step 3** If Macro TLB misses, there will send the request to the PTW to return the corresponding page table;

- 
- Step 4** PTW accesses PTW Cache, the Level1 page table is unnecessary;
- Step 5** PTW accesses memory to get the corresponding Level1 page table and stores it to the PTW Cache (there may happen the replace activity);
- Step 6** PTW logic returns the corresponding Level2 page table from memory page table according to Level1 page table, checks the effectiveness, and sends to Macro TLB;
- Step 7** Macro TLB stores the Level2 page table (there may happen the replace activity), and returns the Level 2 page table to Micro TLB;
- Step 8** Micro TLB receives the page table entries and puts it to the Micro TLB (if this Micro TLB is full, there will happen the replace activity), and sends page table entries to address translation module;
- Step 9** Address transform module converts the virtual address into physical address, and checks the permissions at the same time. If pass, the transfer is completed.

#### Permission error

- Step 1** Permission checking always performs in the address conversion;
- Step 2** Once the permission checking makes mistake, the new access of the master suspends, but continues before this access;
- Step 3** Set the error status register;
- Step 4** Trigger interrupt.

#### Invalid Level1 page table

- Step 1** Invalid Level1 page table is checked when PTW logic reads the new level page table from memory;
- Step 2** The PTW reads sequentially two page table entries from the memory (64-bit data, a complete cache line), and stores in the PTW cache;
- Step 3** If the current page table is detected invalid, then the error flag is set, and the interrupt is triggered, the cache line need to be invalidated.



#### NOTE

- Invalid page table has two situations: the reading target page table from the memory is invalid; and the page table stored in PTW Cache with target page table is found to be invalid after using;
- If a page table is invalid, then the total cache line (that is two page tables) need to be invalidated.

### Invalid Level2 page table

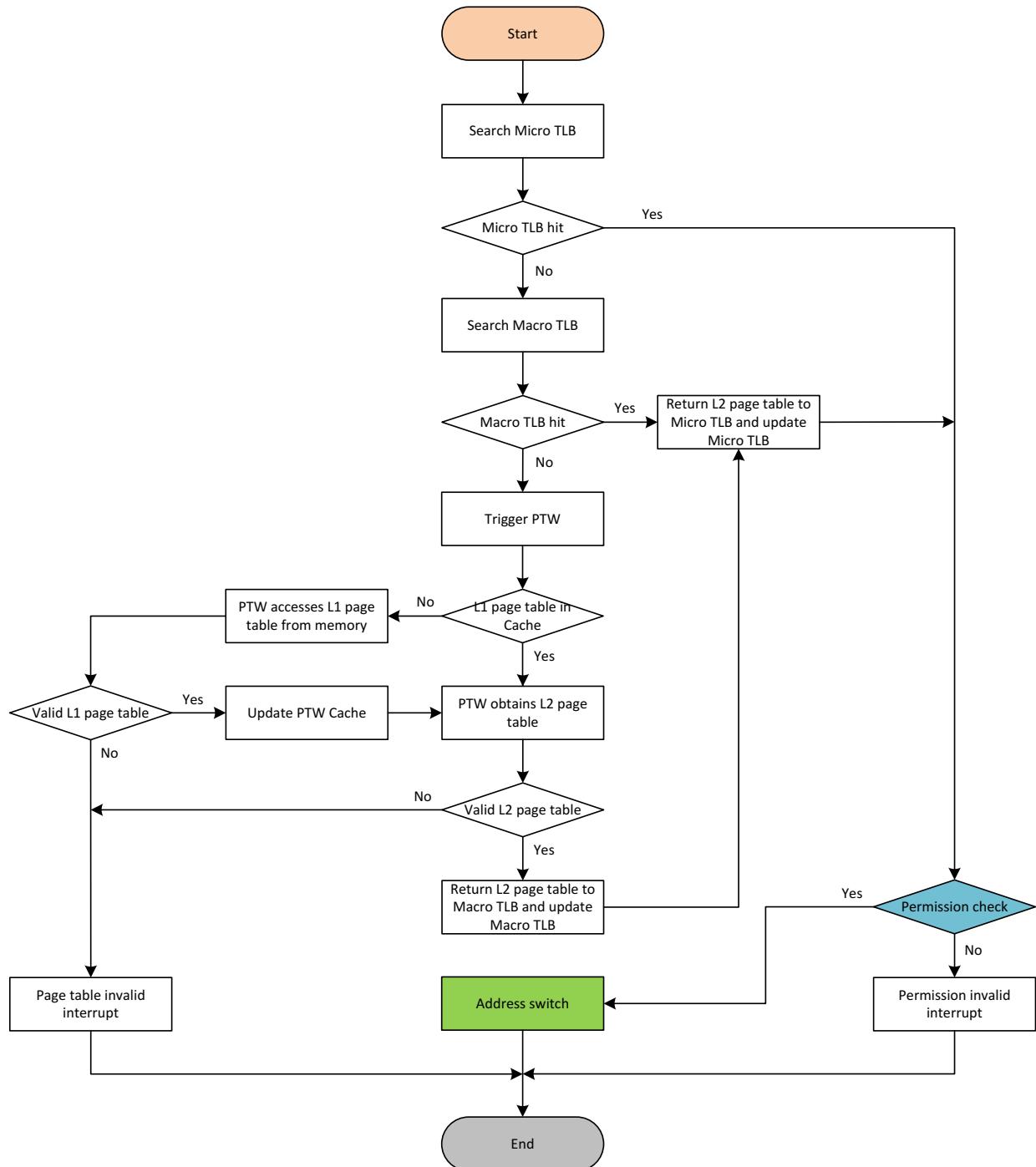
- Step 1** Invalid Level2 page table checks when Macro TLB reads the new level page table from memory;
  - Step 2** The Macro TLB reads sequentially two page table entries from the memory (64-bit data, a complete cache line), and stores in the Macro TLB;
  - Step 3** If the current page table is detected invalid, then the error flag is set, and the interrupt is triggered, the cache line need to be invalidated.
- 



#### NOTE

- Invalid page table has two situations: the target page table read from the memory is invalid; and the page table stored in Macro TLB with target page table is found to be invalid after using.
  - If a page table is invalid, then the total cache line (that is two page tables) needs to be invalidated.
- 

The internal address switch process shows in the following figure.

**Figure 3-30 Internal Switch Process**

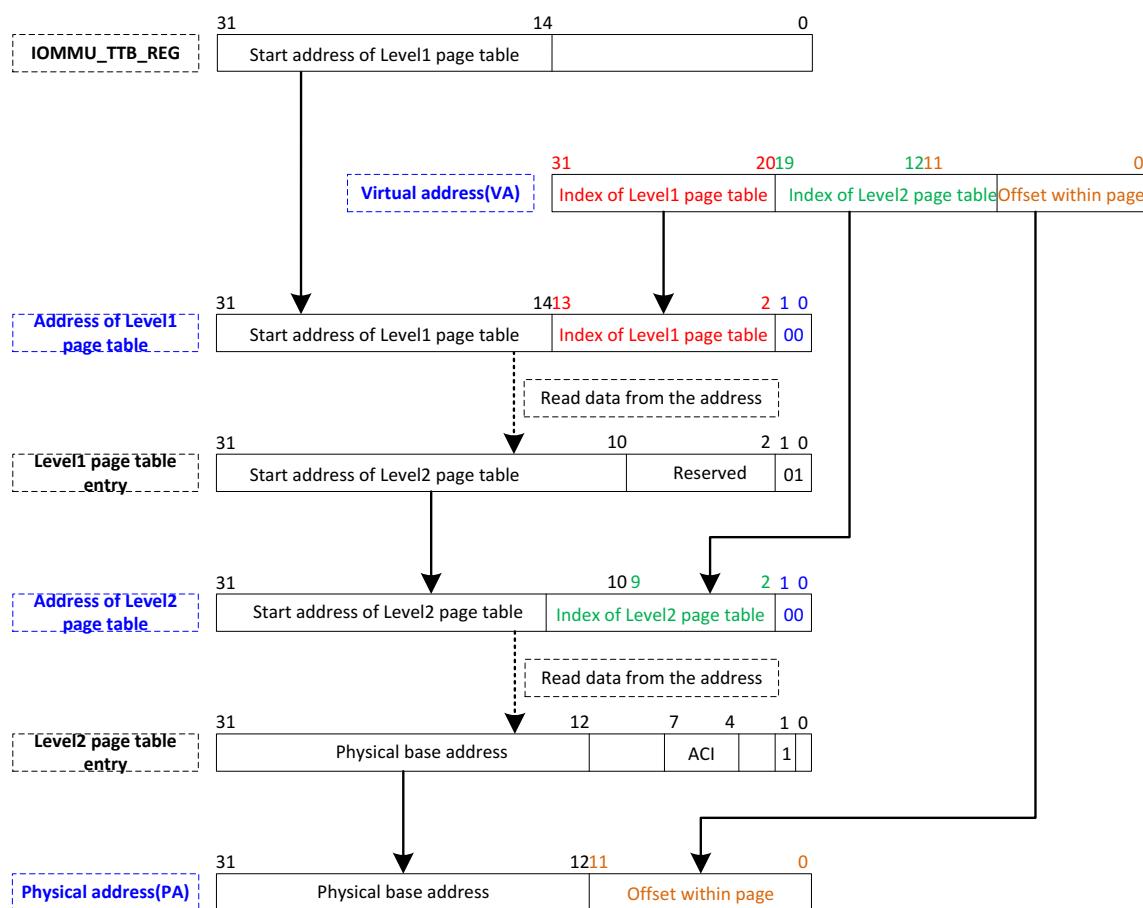
### 3.11.3.3 VA-PA Mapping

IOMMU page table is defined as Level2 mapping, the first level is 1M address space mapping, the second level is 4K address space. This version does not support 1K, 16K and other page table size. IOMMU supports a page table only, its meaning is:

- All peripherals connected to IOMMU use the same virtual address space;
- The virtual address space of the peripherals can overlap;
- Different virtual addresses can map to the same physical address space;

Base address of the page table is defined by software, and it needs 16 KB address alignment; Page table of the Level2 table item needs 1 KB address alignment. The following figure shows a complete VA-PA address translation process.

**Figure 3-31 VA-PA Switch Process**



### 3.11.3.4 Clear and Invalidate TLB

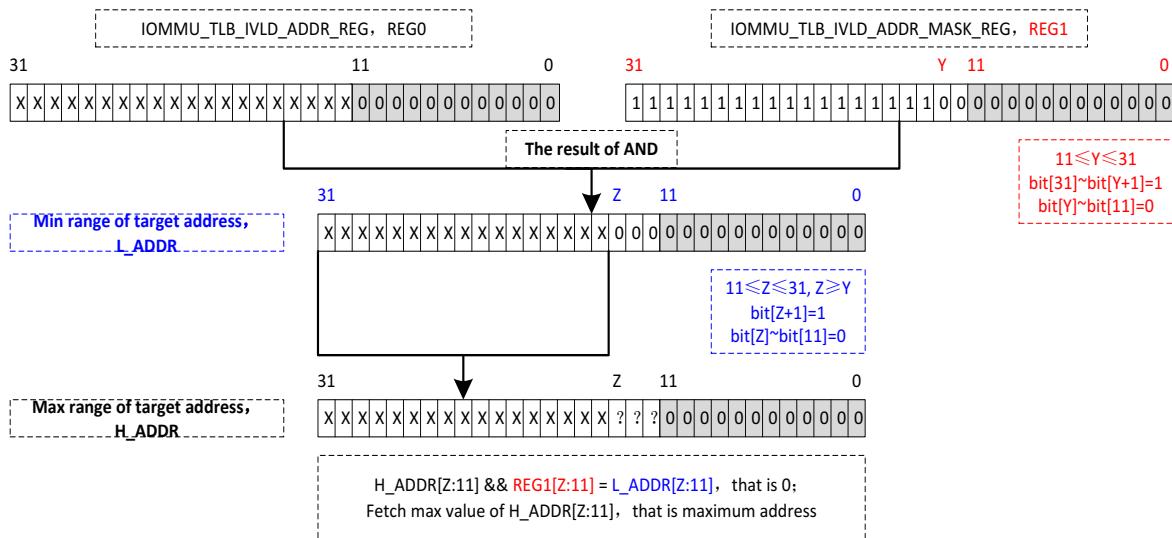
When the content of multi page tables refreshes or the address of page table changes, all VA-PA mapping which has been cached in TLB will no longer be valid, then you need configure **IOMMU TLB Flush Enable Register** to clear the TLB or PTW Cache. First suspend the access to TLB or Cache, then configure the corresponding Flush bit of **IOMMU TLB Flush Enable Register**, after the operation takes effect, the related peripherals can continue to send new access memory operations.

When some page table is invalid or incorrect mapping, you can set the TLB Invalidations relevant register to invalidate some TLB VA-PA mapping pairs. There are two modes to invalidate the TLB operation.

### (1) Mode0 (Old mode)

- Step 1** Set **IOMMU TLB Invalid Mode Select Register** to 0 to select mode0;
- Step 2** Write target address to **IOMMU TLB Invalid Address Register**;
- Step 3** Set configuration values to **IOMMU TLB Invalid Address Mask Register**, the requirements are as follows:
  - The value of **IOMMU TLB Invalid Address Mask Register** cannot be less than the **IOMMU TLB Invalid Address Register**.
  - The higher bit of **IOMMU TLB Invalid Address Mask Register** must be continuous 1, the lower bit must be continuous 0. For example, 0xfffff000, 0xffffe000, 0xffffc000, 0xffff8000, 0xffff0000 belongs to the legal value; and 0xffffd000, 0xffffb000, 0xffffa000, 0xffff9000, 0xffff7000 belongs to the illegal values.
- Step 4** Configure **IOMMU TLB Invalid Enable Register** to enable the invalid operation. Among the way to determine the invalid address is to get maximum valid bit and determine target address range by target address AND mask address. The following figure shows this process.

Figure 3-32 Invalid TLB Address Range



For example:

- When the value of **IOMMU TLB Invalid Address Mask Register** is 0xFFFFF000 by default, the result of AND is target address, that is, only target address is invalid.
- When the value of **IOMMU TLB Invalid Address Mask Register** is 0xFFFF0000, the value of **IOMMU TLB Invalid Address Register** is 0xEEEE1000, then the range of target address is from 0xEEEE0000 to 0xEEEF000.

- c) When the value of **IOMMU TLB Invalid Address Mask Register** is 0xFFFFC000, the value of **IOMMU TLB Invalid Address Register** is 0xEEEE8000, then the range of target address is from 0xEEEE8000 to 0xEEEB000.
- d) When the value of **IOMMU TLB Invalid Address Mask Register** is 0xFFFF8000, the value of **IOMMU TLB Invalid Address Register** is 0xEEEC000, then the range of target address is from 0xEEEE8000 to 0xEEEF000.
- e) When the value of **IOMMU TLB Invalid Address Mask Register** is 0xFFFFC000, the value of **IOMMU TLB Invalid Address Register** is 0xEEE0000, then the range of target address is from 0xEEE0000 to 0xEEE3000.

## (2) Mode1 (New mode)

- Step 1** Set **IOMMU TLB Invalid Mode Select Register** to 1 to select mode1;
- Step 2** Set the starting address of invalid TLB by **IOMMU TLB Invalid Start Address Register**, and set the ending address of invalid TLB by **IOMMU TLB Invalid Start Address Register**;
- Step 3** Configure **IOMMU TLB Invalid Enable Register** to enable invalid operation, then the related TLB operation is invalidated.

### 3.11.3.5 Clear and Invalidate PTW Cache

There are two modes to invalidate the PTW cache operation.

## (1) Mode0 (Old mode)

- Step 1** Set **IOMMU PC Invalid Mode Select Register** to 0 to select mode0;
- Step 2** Set the address register that needs to be invalidated to **IOMMU PC Invalid Address Register** (the addresses need to be aligned with 1 MB);
- Step 3** Configure **IOMMU PC Invalid Enable Register** to enable the invalid operation. That is, the PTW cache operation of a cacheline is invalidated.

## (2) Mode1 (New mode)

- Step 1** Set **IOMMU PC Invalid Mode Select Register** to 1 to select mode1;
- Step 2** Set the starting address of invalid TLB by **IOMMU PTW Invalid Start Address Register**, and set the ending address of invalid TLB by **IOMMU PC Invalid Start Address Register**;

**Step 3** Configure **IOMMU PC Invalidation Enable Register** to enable invalid operation, then to invalid the related PWM cache operation is completed.

### 3.11.3.6 Page Table Format

#### Level1 Page Table

The format of Level1 page table is as follows.

**Figure 3-33 Level1 Page Table Format**

31	10 9	2 1 0
Start address of Level2 page table	Reserved	01

Bit[31:10]: Base address of Level2 page table;

Bit[9:2]: Reserved;

Bit[1:0]: 01 is valid page table; others are fault;

#### Level2 Page Table

The format of Level2 page table is as follows.

**Figure 3-34 Level1 Page Table Format**

31	12	7	4	1 0
Physical base address		ACI		1

Bit[31:12]: Physical address of 4K address;

Bit[11:8]: Reserved;

Bit[7:4]: ACI, permission control index; correspond to permission control bit of **IOMMU Domain Authority Control Register**;

Bit[3:2]: Reserved;

Bit[1]: 1 is valid page table; 0 is fault;

Bit[0]: Reserved

#### Permission Index

The read/write access control of series register such as **IOMMU Domain Authority Control Register** is as follows.

**Figure 3-35 Read/Write Permission Control**

13	12	11	10	9	8	7	6	5	4	3	2	1	0
W	R	W	R	W	R	W	R	W	R	W	R	W	R

Bit[1:0]/Bit[17:16]: Master0 read/write permission control;

Bit[3:2]/Bit[19:18]: Master1 read/write permission control, reserved;

Bit[5:4]/Bit[21:20]: Master2 read/write permission control;

Bit[7:6]/Bit[23:22]: Master3 read/write permission control;

Bit[9:8]/Bit[25:24]: Master4 read/write permission control;

Bit[11:10]/Bit[27:26]: Master5 read/write permission control, reserved;

Bit[13:12]/Bit[29:28]: Master6 read/write permission control, reserved.

The value of **IOMMU Domain Authority Control Register** is read-only by default. Other registers can configure through system requirement. In address switch process, the corresponding relation between ACI and Domain is as follows.

**Table 3-18 Relation between ACI and Domain**

ACI	Domain	Register
0	Domain 0	IOMMU Domain Authority Control Register 0
1	Domain 1	IOMMU Domain Authority Control Register 0
2	Domain 2	IOMMU Domain Authority Control Register 1
3	Domain 3	IOMMU Domain Authority Control Register 1
4	Domain 4	IOMMU Domain Authority Control Register 2
5	Domain 5	IOMMU Domain Authority Control Register 2
6	Domain 6	IOMMU Domain Authority Control Register 3
7	Domain 7	IOMMU Domain Authority Control Register 3
8	Domain 8	IOMMU Domain Authority Control Register 4
9	Domain 9	IOMMU Domain Authority Control Register 4
10	Domain 10	IOMMU Domain Authority Control Register 5
11	Domain 11	IOMMU Domain Authority Control Register 5
12	Domain 12	IOMMU Domain Authority Control Register 6
13	Domain 13	IOMMU Domain Authority Control Register 6
14	Domain 14	IOMMU Domain Authority Control Register 7
15	Domain 15	IOMMU Domain Authority Control Register 7

---

After enabled **IOMMU Domain Authority Overwrite Register**, the read/write control permission can override all **IOMMU Domain Authority Control Register**.

### 3.11.4 Programming Guidelines

#### 3.11.4.1 Resetting IOMMU

Before the IOMMU module software reset operation, make sure IOMMU is never opened, or all bus operations are completed, or DRAM and peripherals already open the corresponding switch, to shield the influence of IOMMU reset.

#### 3.11.4.2 Enabling IOMMU

Before opening the IOMMU address mapping function, the **Translation Table Base Register** should be correctly configured, or all the masters are in the bypass state, or all the masters do not send the bus command.

#### 3.11.4.3 Configuring TTB

Operating the register must close the address mapping function of IOMMU, that is, the IOMMU\_ENABLE\_REG[0] is 0; or the bypass function of all masters is set to 1; or the state of the TX bus command is none.

#### 3.11.4.4 Clearing TTB

In the Flush operation, all TLB/Cache access will be suspended; but the operation entered the TLB will continue to complete before the Flush starts.

#### 3.11.4.5 Reading/Writing VA Data

For target virtual address, read and write the corresponding physical address data to make sure whether IOMMU module address mapping function is normal. First, make sure to read or write, and then configure the target virtual address or write data, then start to read or write function, check whether the results are as expected after the operation is finished.

### 3.11.4.6 Using PMU Statistics Function

When PMU function is used for the first time, set **IOMMU PMU Enable Register** to enable statistics function; when reading the relevant Register, clear the enable bit of **IOMMU PMU Enable Register**; when PMU function is used next time, first **IOMMU PMU Clear Register** is set, after counter is cleared, set the enable bit of **IOMMU PMU Enable Register**.

Given a Level2 page table administers continuous 4 KB address, if Micro TLB misses in continuous virtual address, there may need to return a Level2 page table to hit from Macro TLB; but the hit number is not recorded in the Macro TLB hit and Micro TLB hit related register. So the true hit rate calculation is as follows:

$$\text{Hit Rate} = N1/M1 + (1-N1/M1)*N2/M2$$

N1: Micro TLB hit number

M1: Micro TLB access number

N2: Macro TLB hit number

M2: Macro TLB access number

### 3.11.5 Register List

Module Name	Base Address
IOMMU	0x02010000

Register Name	Offset	Description
IOMMU_RESET_REG	0x0010	IOMMU Reset Register
IOMMU_ENABLE_REG	0x0020	IOMMU Enable Register
IOMMU_BYPASS_REG	0x0030	IOMMU Bypass Register
IOMMU_AUTO_GATING_REG	0x0040	IOMMU Auto Gating Register
IOMMU_WBUF_CTRL_REG	0x0044	IOMMU Write Buffer Control Register
IOMMU_OOO_CTRL_REG	0x0048	IOMMU Out of Order Control Register
IOMMU_4KB_BDY_PRT_CTRL_REG	0x004C	IOMMU 4KB Boundary Protect Control Register
IOMMU_TTB_REG	0x0050	IOMMU Translation Table Base Register
IOMMU_TLB_ENABLE_REG	0x0060	IOMMU TLB Enable Register
IOMMU_TLB_PREFETCH_REG	0x0070	IOMMU TLB Prefetch Register
IOMMU_TLB_FLUSH_ENABLE_REG	0x0080	IOMMU TLB Flush Enable Register
IOMMU_TLB_IVLD_MODE_SEL_REG	0x0084	IOMMU TLB Invalidation Mode Select Register
IOMMU_TLB_IVLD_STA_ADDR_REG	0x0088	IOMMU TLB Invalidation Start Address Register
IOMMU_TLB_IVLD_END_ADDR_REG	0x008C	IOMMU TLB Invalidation End Address Register

<b>Register Name</b>	<b>Offset</b>	<b>Description</b>
IOMMU_TLB_IVLD_ADDR_REG	0x0090	IOMMU TLB Invalidiation Address Register
IOMMU_TLB_IVLD_ADDR_MASK_REG	0x0094	IOMMU TLB Invalidiation Address Mask Register
IOMMU_TLB_IVLD_ENABLE_REG	0x0098	IOMMU TLB Invalidiation Enable Register
IOMMU_PC_IVLD_MODE_SEL_REG	0x009C	IOMMU PC Invalidation Mode Select Register
IOMMU_PC_IVLD_ADDR_REG	0x00A0	IOMMU PC Invalidation Address Register
IOMMU_PC_IVLD_STA_ADDR_REG	0x00A4	IOMMU PC Invalidation Start Address Register
IOMMU_PC_IVLD_ENABLE_REG	0x00A8	IOMMU PC Invalidation Enable Register
IOMMU_PC_IVLD_END_ADDR_REG	0x00AC	IOMMU PC Invalidation End Address Register
IOMMU_DM_AUT_CTRL0_REG	0x00B0	IOMMU Domain Authority Control 0 Register
IOMMU_DM_AUT_CTRL1_REG	0x00B4	IOMMU Domain Authority Control 1 Register
IOMMU_DM_AUT_CTRL2_REG	0x00B8	IOMMU Domain Authority Control 2 Register
IOMMU_DM_AUT_CTRL3_REG	0x00BC	IOMMU Domain Authority Control 3 Register
IOMMU_DM_AUT_CTRL4_REG	0x00C0	IOMMU Domain Authority Control 4 Register
IOMMU_DM_AUT_CTRL5_REG	0x00C4	IOMMU Domain Authority Control 5 Register
IOMMU_DM_AUT_CTRL6_REG	0x00C8	IOMMU Domain Authority Control 6 Register
IOMMU_DM_AUT_CTRL7_REG	0x00CC	IOMMU Domain Authority Control 7 Register
IOMMU_DM_AUT_OVWT_REG	0x00D0	IOMMU Domain Authority Overwrite Register
IOMMU_INT_ENABLE_REG	0x0100	IOMMU Interrupt Enable Register
IOMMU_INT_CLR_REG	0x0104	IOMMU Interrupt Clear Register
IOMMU_INT_STA_REG	0x0108	IOMMU Interrupt Status Register
IOMMU_INT_ERR_ADDR0_REG	0x0110	IOMMU Interrupt Error Address 0 Register
IOMMU_INT_ERR_ADDR1_REG	0x0114	IOMMU Interrupt Error Address 1 Register
IOMMU_INT_ERR_ADDR2_REG	0x0118	IOMMU Interrupt Error Address 2 Register
IOMMU_INT_ERR_ADDR3_REG	0x011C	IOMMU Interrupt Error Address 3 Register
IOMMU_INT_ERR_ADDR4_REG	0x0120	IOMMU Interrupt Error Address 4 Register
IOMMU_INT_ERR_ADDR5_REG	0x0124	IOMMU Interrupt Error Address 5 Register
IOMMU_INT_ERR_ADDR6_REG	0x0128	IOMMU Interrupt Error Address 6 Register
IOMMU_INT_ERR_ADDR7_REG	0x0130	IOMMU Interrupt Error Address 7 Register
IOMMU_INT_ERR_ADDR8_REG	0x0134	IOMMU Interrupt Error Address 8 Register
IOMMU_INT_ERR_DATA0_REG	0x0150	IOMMU Interrupt Error Data 0 Register
IOMMU_INT_ERR_DATA1_REG	0x0154	IOMMU Interrupt Error Data 1 Register
IOMMU_INT_ERR_DATA2_REG	0x0158	IOMMU Interrupt Error Data 2 Register
IOMMU_INT_ERR_DATA3_REG	0x015C	IOMMU Interrupt Error Data 3 Register
IOMMU_INT_ERR_DATA4_REG	0x0160	IOMMU Interrupt Error Data 4 Register

<b>Register Name</b>	<b>Offset</b>	<b>Description</b>
IOMMU_INT_ERR_DATA5_REG	0x0164	IOMMU Interrupt Error Data 5 Register
IOMMU_INT_ERR_DATA6_REG	0x0168	IOMMU Interrupt Error Data 6 Register
IOMMU_INT_ERR_DATA7_REG	0x0170	IOMMU Interrupt Error Data 7 Register
IOMMU_INT_ERR_DATA8_REG	0x0174	IOMMU Interrupt Error Data 8 Register
IOMMU_L1PG_INT_REG	0x0180	IOMMU L1 Page Table Interrupt Register
IOMMU_L2PG_INT_REG	0x0184	IOMMU L2 Page Table Interrupt Register
IOMMU_VA_REG	0x0190	IOMMU Virtual Address Register
IOMMU_VA_DATA_REG	0x0194	IOMMU Virtual Address Data Register
IOMMU_VA_CONFIG_REG	0x0198	IOMMU Virtual Address Configuration Register
IOMMU_PMU_ENABLE_REG	0x0200	IOMMU PMU Enable Register
IOMMU_PMU_CLR_REG	0x0210	IOMMU PMU Clear Register
IOMMU_PMU_ACCESS_LOW0_REG	0x0230	IOMMU PMU Access Low 0 Register
IOMMU_PMU_ACCESS_HIGH0_REG	0x0234	IOMMU PMU Access High 0 Register
IOMMU_PMU_HIT_LOW0_REG	0x0238	IOMMU PMU Hit Low 0 Register
IOMMU_PMU_HIT_HIGH0_REG	0x023C	IOMMU PMU Hit High 0 Register
IOMMU_PMU_ACCESS_LOW1_REG	0x0240	IOMMU PMU Access Low 1 Register
IOMMU_PMU_ACCESS_HIGH1_REG	0x0244	IOMMU PMU Access High 1 Register
IOMMU_PMU_HIT_LOW1_REG	0x0248	IOMMU PMU Hit Low 1 Register
IOMMU_PMU_HIT_HIGH1_REG	0x024C	IOMMU PMU Hit High 1 Register
IOMMU_PMU_ACCESS_LOW2_REG	0x0250	IOMMU PMU Access Low 2 Register
IOMMU_PMU_ACCESS_HIGH2_REG	0x0254	IOMMU PMU Access High 2 Register
IOMMU_PMU_HIT_LOW2_REG	0x0258	IOMMU PMU Hit Low 2 Register
IOMMU_PMU_HIT_HIGH2_REG	0x025C	IOMMU PMU Hit High 2 Register
IOMMU_PMU_ACCESS_LOW3_REG	0x0260	IOMMU PMU Access Low 3 Register
IOMMU_PMU_ACCESS_HIGH3_REG	0x0264	IOMMU PMU Access High 3 Register
IOMMU_PMU_HIT_LOW3_REG	0x0268	IOMMU PMU Hit Low 3 Register
IOMMU_PMU_HIT_HIGH3_REG	0x026C	IOMMU PMU Hit High 3 Register
IOMMU_PMU_ACCESS_LOW4_REG	0x0270	IOMMU PMU Access Low 4 Register
IOMMU_PMU_ACCESS_HIGH4_REG	0x0274	IOMMU PMU Access High 4 Register
IOMMU_PMU_HIT_LOW4_REG	0x0278	IOMMU PMU Hit Low 4 Register
IOMMU_PMU_HIT_HIGH4_REG	0x027C	IOMMU PMU Hit High 4 Register
IOMMU_PMU_ACCESS_LOW5_REG	0x0280	IOMMU PMU Access Low 5 Register
IOMMU_PMU_ACCESS_HIGH5_REG	0x0284	IOMMU PMU Access High 5 Register
IOMMU_PMU_HIT_LOW5_REG	0x0288	IOMMU PMU Hit Low 5 Register

Register Name	Offset	Description
IOMMU_PMU_HIT_HIGH5_REG	0x028C	IOMMU PMU Hit High 5 Register
IOMMU_PMU_ACCESS_LOW6_REG	0x0290	IOMMU PMU Access Low 6 Register
IOMMU_PMU_ACCESS_HIGH6_REG	0x0294	IOMMU PMU Access High 6 Register
IOMMU_PMU_HIT_LOW6_REG	0x0298	IOMMU PMU Hit Low 6 Register
IOMMU_PMU_HIT_HIGH6_REG	0x029C	IOMMU PMU Hit High 6 Register
IOMMU_PMU_ACCESS_LOW7_REG	0x02D0	IOMMU PMU Access Low 7 Register
IOMMU_PMU_ACCESS_HIGH7_REG	0x02D4	IOMMU PMU Access High 7 Register
IOMMU_PMU_HIT_LOW7_REG	0x02D8	IOMMU PMU Hit Low 7 Register
IOMMU_PMU_HIT_HIGH7_REG	0x02DC	IOMMU PMU Hit High 7 Register
IOMMU_PMU_ACCESS_LOW8_REG	0x02E0	IOMMU PMU Access Low 8 Register
IOMMU_PMU_ACCESS_HIGH8_REG	0x02E4	IOMMU PMU Access High 8 Register
IOMMU_PMU_HIT_LOW8_REG	0x02E8	IOMMU PMU Hit Low 8 Register
IOMMU_PMU_HIT_HIGH8_REG	0x02EC	IOMMU PMU Hit High 8 Register
IOMMU_PMU_TL_LOW0_REG	0x0300	IOMMU Total Latency Low 0 Register
IOMMU_PMU_TL_HIGH0_REG	0x0304	IOMMU Total Latency High 0 Register
IOMMU_PMU_ML0_REG	0x0308	IOMMU Max Latency 0 Register
IOMMU_PMU_TL_LOW1_REG	0x0310	IOMMU Total Latency Low 1 Register
IOMMU_PMU_TL_HIGH1_REG	0x0314	IOMMU Total Latency High 1 Register
IOMMU_PMU_ML1_REG	0x0318	IOMMU Max Latency 1 Register
IOMMU_PMU_TL_LOW2_REG	0x0320	IOMMU Total Latency Low 2 Register
IOMMU_PMU_TL_HIGH2_REG	0x0324	IOMMU Total Latency High 2 Register
IOMMU_PMU_ML2_REG	0x0328	IOMMU Max Latency 2 Register
IOMMU_PMU_TL_LOW3_REG	0x0330	IOMMU Total Latency Low 3 Register
IOMMU_PMU_TL_HIGH3_REG	0x0334	IOMMU Total Latency High 3 Register
IOMMU_PMU_ML3_REG	0x0338	IOMMU Max Latency 3 Register
IOMMU_PMU_TL_LOW4_REG	0x0340	IOMMU Total Latency Low 4 Register
IOMMU_PMU_TL_HIGH4_REG	0x0344	IOMMU Total Latency High 4 Register
IOMMU_PMU_ML4_REG	0x0348	IOMMU Max Latency 4 Register
IOMMU_PMU_TL_LOW5_REG	0x0350	IOMMU Total Latency Low 5 Register
IOMMU_PMU_TL_HIGH5_REG	0x0354	IOMMU Total Latency High 5 Register
IOMMU_PMU_ML5_REG	0x0358	IOMMU Max Latency 5 Register
IOMMU_PMU_TL_LOW6_REG	0x0360	IOMMU Total Latency Low 6 Register
IOMMU_PMU_TL_HIGH6_REG	0x0364	IOMMU Total Latency High 6 Register
IOMMU_PMU_ML6_REG	0x0368	IOMMU Max Latency 6 Register

### 3.11.6 Register Description

#### 3.11.6.1 0x0010 IOMMU Reset Register (Default Value: 0x8003\_007F)

Offset: 0x0010			Register Name: IOMMU_RESET_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x1	<p>IOMMU_RESET IOMMU Software Reset Switch 0: Set reset signal 1: Release reset signal Before the IOMMU software reset operation, ensure IOMMU never be opened; or all bus operations are completed; or DRAM and the peripherals have opened the corresponding switch, for shielding the effects of IOMMU reset.</p>
30:18	/	/	/
17	R/W	0x1	<p>PC_RST PTW Cache Reset PTW Cache address convert lane software reset switch. 0: Set reset signal 1: Release reset signal When PTW Cache occurs abnormal, the bit is used to reset PTW Cache individually.</p>
16	R/W	0x1	<p>MTLB_RST Macrotlb Reset Macro TLB address convert lane software reset switch. 0: Set reset signal 1: Release reset signal When PTW Cache occurs abnormal, the bit is used to reset PTW Cache individually.</p>
15:7	/	/	/
6	R/W	0x1	<p>M6_RST Master6 Reset Master6 address convert lane software reset switch. 0: Set reset signal 1: Release reset signal When Master6 occurs abnormal, the bit is used to reset PTW Cache individually. <b>Note: This bit is not used.</b></p>

Offset: 0x0010			Register Name: IOMMU_RESET_REG
Bit	Read/Write	Default/Hex	Description
5	R/W	0x1	<p>M5_RST Master5 Reset Master5 address convert lane software reset switch. 0: Set reset signal 1: Release reset signal When Master5 occurs abnormal, the bit is used to reset PTW Cache individually.</p> <p><b>Note: This bit is not used.</b></p>
4	R/W	0x1	<p>M4_RST Master4 Reset Master4 address convert lane software reset switch. 0: Set reset signal 1: Release reset signal When Master4 occurs abnormal, the bit is used to reset PTW Cache individually.</p>
3	R/W	0x1	<p>M3_RST Master3 Reset Master3 address convert lane software reset switch. 0: Set reset signal 1: Release reset signal When Master3 occurs abnormal, the bit is used to reset PTW Cache individually.</p>
2	R/W	0x1	<p>M2_RST Master2 Reset Master2 address convert lane software reset switch. 0: Set reset signal 1: Release reset signal When Master2 occurs abnormal, the bit is used to reset PTW Cache individually.</p>
1	R/W	0x1	<p>M1_RST Master1 Reset Master1 address convert lane software reset switch. 0: Set reset signal 1: Release reset signal When Master1 occurs abnormal, the bit is used to reset PTW Cache individually.</p> <p><b>Note: This bit is not used.</b></p>
0	R/W	0x1	M0_RST

Offset: 0x0010			Register Name: IOMMU_RESET_REG
Bit	Read/Write	Default/Hex	Description
			<p>Master0 Reset Master0 address convert lane software reset switch. 0: Set reset signal 1: Release reset signal When Master0 occurs abnormal, the bit is used to reset PTW Cache individually.</p>

### 3.11.6.2 0x0020 IOMMU Enable Register (Default Value: 0x0000\_0000)

Offset: 0x0020			Register Name: IOMMU_ENABLE_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	<p>ENABLE IOMMU module enable switch 0: Disable IOMMU 1: Enable IOMMU Before IOMMU address mapping function opens, configure the Translation Table Base register; or ensure all masters are in bypass status or no the status of sending bus demand(such as reset)</p>

### 3.11.6.3 0x0030 IOMMU Bypass Register (Default Value: 0x0000\_007F)

Offset: 0x0030			Register Name: IOMMU_BYPASS_REG
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6	R/W	0x1	<p>M6_BP Master6 bypass switch After bypass function is opened, IOMMU can not map the address of Master6 sending, and directly output the virtual address to MBUS as physical address. 0: Disable bypass function 1: Enable bypass function <b>Note: The bit is not used.</b></p>
5	R/W	0x1	<p>M5_BP Master5 bypass switch</p>

Offset: 0x0030			Register Name: IOMMU_BYPASS_REG
Bit	Read/Write	Default/Hex	Description
			<p>After bypass function is opened, IOMMU can not map the address of Master5 sending, and directly output the virtual address to MBUS as physical address.</p> <p>0: Disable bypass function 1: Enable bypass function</p> <p><b>Note: This bit is not used.</b></p>
4	R/W	0x1	<p>M4_BP</p> <p>Master4 bypass switch</p> <p>After bypass function is opened, IOMMU can not map the address of Master4 sending, and directly output the virtual address to MBUS as physical address.</p> <p>0: Disable bypass function 1: Enable bypass function</p>
3	R/W	0x1	<p>M3_BP</p> <p>Master3 bypass switch</p> <p>After bypass function is opened, IOMMU can not map the address of Master3 sending, and directly output the virtual address to MBUS as physical address.</p> <p>0: Disable bypass function 1: Enable bypass function</p>
2	R/W	0x1	<p>M2_BP</p> <p>Master2 bypass switch</p> <p>After bypass function is opened, IOMMU can not map the address of Master2 sending, and directly output the virtual address to MBUS as physical address.</p> <p>0: Disable bypass function 1: Enable bypass function</p>
1	R/W	0x1	<p>M1_BP</p> <p>Master1 bypass switch</p> <p>After bypass function is opened, IOMMU can not map the address of Master1 sending, and directly output the virtual address to MBUS as physical address.</p> <p>0: Disable bypass function 1: Enable bypass function</p> <p><b>Note: This bit is not used.</b></p>
0	R/W	0x1	<p>M0_BP</p> <p>Master0 bypass switch</p>

Offset: 0x0030			Register Name: IOMMU_BYPASS_REG
Bit	Read/Write	Default/Hex	Description
			<p>After bypass function is opened, IOMMU can not map the address of Master0 sending, and directly output the virtual address to MBUS as physical address.</p> <p>0: Disable bypass function 1: Enable bypass function</p>

 **NOTE**

- Operating the register belongs to non-accurate timing sequence control function. That is, before the function is valid, master operation will complete address mapping function, and any subsequent operation will not perform address mapping.
- It is suggested that master is in reset state or in no any bus operation before operating the register.

#### 3.11.6.4 0x0040 IOMMU Auto Gating Register (Default Value: 0x0000\_0001)

Offset: 0x0040			Register Name: IOMMU_AUTO_GATING_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x1	<p>IOMMU_AUTO_GATING IOMMU circuit auto gating control The purpose is to decrease power consumption of the module. 0: Disable auto gating function 1: Enable auto gating function</p>

#### 3.11.6.5 0x0044 IOMMU Write Buffer Control Register (Default Value: 0x0000\_007F)

Offset: 0x0044			Register Name: IOMMU_WBUF_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6	R/W	0x1	Reserved
5	R/W	0x1	Reserved
4	R/W	0x1	Reserved
3	R/W	0x1	Reserved
2	R/W	0x1	Reserved

Offset: 0x0044			Register Name: IOMMU_WBUF_CTRL_REG
Bit	Read/Write	Default/Hex	Description
1	R/W	0x1	Reserved
0	R/W	0x1	Reserved

### 3.11.6.6 0x0048 IOMMU Out Of Order Control Register (Default Value: 0x0000\_007F)

Offset: 0x0048			Register Name: IOMMU_OOO_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6	R/W	0x1	M6_OOO_CTRL Master6 out-of-order control bit 0: Disable out-of-order 1: Enable out-of-order <b>Note: This bit is not used.</b>
5	R/W	0x1	M5_OOO_CTRL Master5 out-of-order control bit 0: Disable out-of-order 1: Enable out-of-order <b>Note: This bit is not used.</b>
4	R/W	0x1	M4_OOO_CTRL Master4 out-of-order control bit 0: Disable out-of-order 1: Enable out-of-order
3	R/W	0x1	M3_OOO_CTRL Master3 out-of-order control bit 0: Disable out-of-order 1: Enable out-of-order
2	R/W	0x1	M2_OOO_CTRL Master2 out-of-order control bit 0: Disable out-of-order 1: Enable out-of-order
1	R/W	0x1	M1_OOO_CTRL Master1 out-of-order control bit 0: Disable out-of-order 1: Enable out-of-order <b>Note: This bit is not used.</b>

Offset: 0x0048			Register Name: IOMMU_OOO_CTRL_REG
Bit	Read/Write	Default/Hex	Description
0	R/W	0x1	M0_OOO_CTRL Master0 out-of-order control bit 0: Disable out-of-order 1: Enable out-of-order

### 3.11.6.7 0x004C IOMMU 4KB Boundary Protect Control Register (Default Value: 0x0000\_007F)



When the virtual address sent by master is over the 4 KB boundary, 4 KB protection unit will split it into two serial access.

Offset: 0x004C			Register Name: IOMMU_4KB_BDY_PRT_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6	R/W	0x1	M6_4KB_BDY_PRT_CTRL Master6 4 KB boundary protect control bit 0: Disable 4 KB boundary protect 1: Enable 4 KB boundary protect <b>Note: This bit is not used.</b>
5	R/W	0x1	M5_4KB_BDY_PRT_CTRL Master4 4 KB boundary protect control bit 0: Disable 4 KB boundary protect 1: Enable 4 KB boundary protect <b>Note: This bit is not used.</b>
4	R/W	0x1	M4_4KB_BDY_PRT_CTRL Master4 4 KB boundary protect control bit 0: Disable 4 KB boundary protect 1: Enable 4 KB boundary protect
3	R/W	0x1	M3_4KB_BDY_PRT_CTRL Master3 4 KB boundary protect control bit 0: Disable 4 KB boundary protect 1: Enable 4 KB boundary protect
2	R/W	0x1	M2_4KB_BDY_PRT_CTRL

Offset: 0x004C			Register Name: IOMMU_4KB_BDY_PRT_CTRL_REG
Bit	Read/Write	Default/Hex	Description
			Master2 4 KB boundary protect control bit 0: Disable 4 KB boundary protect 1: Enable 4 KB boundary protect
1	R/W	0x1	M1_4KB_BDY_PRT_CTRL Master1 4 KB boundary protect control bit 0: Disable 4 KB boundary protect 1: Enable 4 KB boundary protect <b>Note: This bit is not used.</b>
0	R/W	0x1	M0_4KB_BDY_PRT_CTRL Master0 4 KB boundary protect control bit 0: Disable 4 KB boundary protect 1: Enable 4 KB boundary protect

### 3.11.6.8 0x0050 IOMMU Translation Table Base Register (Default Value: 0x0000\_0000)

Offset: 0x0050			Register Name: IOMMU_TTB_REG
Bit	Read/Write	Default/Hex	Description
31:14	R/W	0x0	TTB Translation Table Base Level1 page table starting address, aligned to 16 KB. When operating the register, IOMMU address mapping function must be closed, namely IOMMU_ENABLE_REG is 0; Or Bypass function of all main equipment is set to 1, or no the state of transfer bus commands (such as setting).
13:0	/	/	/

### 3.11.6.9 0x0060 IOMMU TLB Enable Register (Default Value: 0x0003\_007F)

Offset: 0x0060			Register Name: IOMMU_TLB_ENABLE_REG
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
17	R/W	0x1	PTW_CACHE_ENABLE PTW Cache enable bit 0: Disable 1: Enable

Offset: 0x0060			Register Name: IOMMU_TLB_ENABLE_REG
Bit	Read/Write	Default/Hex	Description
16	R/W	0x1	<p>MACRO_TLB_ENABLE Macro TLB enable bit 0: Disable 1: Enable</p>
15:7	/	/	/
6	R/W	0x1	<p>MICRO_TLB6_ENABLE Micro TLB6 enable bit 0: Disable 1: Enable</p> <p><b>Note: This bit is not used.</b></p>
5	R/W	0x1	<p>MICRO_TLB5_ENABLE Micro TLB5 enable bit 0: Disable 1: Enable</p> <p><b>Note: This bit is not used.</b></p>
4	R/W	0x1	<p>MICRO_TLB4_ENABLE Micro TLB4 enable bit 0: Disable 1: Enable</p>
3	R/W	0x1	<p>MICRO_TLB3_ENABLE Micro TLB3 enable bit 0: Disable 1: Enable</p>
2	R/W	0x1	<p>MICRO_TLB2_ENABLE Micro TLB2 enable bit 0: Disable 1: Enable</p>
1	R/W	0x1	<p>MICRO_TLB1_ENABLE Micro TLB1 enable bit 0: Disable 1: Enable</p> <p><b>Note: This bit is not used.</b></p>
0	R/W	0x1	<p>MICRO_TLB0_ENABLE Micro TLB0 enable bit 0: Disable 1: Enable</p>

### 3.11.6.10 0x0070 IOMMU TLB Prefetch Register (Default Value: 0x0000\_0000)

Offset: 0x0070			Register Name: IOMMU_TLB_PREFETCH_REG
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
17	R/W	0x1	<p>PF_VL_PT_TO_PC Prefetch Value Pagetable to PTW Cache 0: Disable 1: Enable If the function is enabled, the prefetch function will not update the invalid Level1 page table to PTW cache.</p>
16	R/W	0x1	<p>PF_VL_PT_TO_MT Prefetch Value Pagetable to Macro TLB 0: Disable 1: Enable If the function is enabled, the prefetch function will not update the invalid Level2 page table to Macro TLB.</p>
15:7	/	/	/
6	R/W	0x0	<p>MI_TLB6_PF Micro TLB6 prefetch enable 0: Disable 1: Enable <b>Note: This bit is not used.</b></p>
5	R/W	0x0	<p>MI_TLB5_PF Micro TLB5 prefetch enable 0: Disable 1: Enable <b>Note: This bit is not used.</b></p>
4	R/W	0x0	<p>MI_TLB4_PF Micro TLB4 prefetch enable 0: Disable 1: Enable</p>
3	R/W	0x0	<p>MI_TLB3_PF Micro TLB3 prefetch enable 0: Disable 1: Enable</p>

Offset: 0x0070			Register Name: IOMMU_TLB_PREFETCH_REG
Bit	Read/Write	Default/Hex	Description
			<b>Note: If G2D accesses DDR, it is suggested that the prefetch function is disabled.</b>
2	R/W	0x0	MI_TLB2_PF Micro TLB2 prefetch enable 0: Disable 1: Enable
1	R/W	0x0	MI_TLB1_PF Micro TLB1 prefetch enable 0: Disable 1: Enable <b>Note: This bit is not used.</b>
0	R/W	0x0	MI_TLB0_PF Micro TLB0 prefetch enable 0: Disable 1: Enable

### 3.11.6.11 0x0080 IOMMU TLB Flush Enable Register (Default Value: 0x0000\_0000)

When performing flush operations, all TLB/Cache access will be paused.

Before flush starts, the operation that has entered TLB continues to complete.

Offset: 0x0080			Register Name: IOMMU_TLB_FLUSH_ENABLE_REG
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
17	R/WAC	0x0	PC_FS PTW Cache Flush Clear PTW Cache 0: No clear operation or clear operation is completed 1: Enable clear operation After the Flush operation is completed, the bit can clear automatically.
16	R/WAC	0x0	MA_TLB_FS Macro TLB Flush Clear Macro TLB 0: No clear operation or clear operation is completed 1: Enable clear operation

Offset: 0x0080			Register Name: IOMMU_TLB_FLUSH_ENABLE_REG
Bit	Read/Write	Default/Hex	Description
			After the Flush operation is completed, the bit can clear automatically.
15:7	/	/	/
6	R/WAC	0x0	<p>MI_TLB6_FS Micro TLB6 Flush Clear Micro TLB6</p> <p>0: No clear operation or clear operation is completed 1: Enable clear operation</p> <p>After the Flush operation is completed, the bit can clear automatically.</p> <p><b>Note: This bit is not used.</b></p>
5	R/WAC	0x0	<p>MI_TLB5_FS Micro TLB5 Flush Clear Micro TLB5</p> <p>0: No clear operation or clear operation is completed 1: Enable clear operation</p> <p>After the Flush operation is completed, the bit can clear automatically.</p> <p><b>Note: This bit is not used.</b></p>
4	R/WAC	0x0	<p>MI_TLB4_FS Micro TLB4 Flush Clear Micro TLB4</p> <p>0: No clear operation or clear operation is completed 1: Enable clear operation</p> <p>After the Flush operation is completed, the bit can clear automatically.</p>
3	R/WAC	0x0	<p>MI_TLB3_FS Micro TLB3 Flush Clear Micro TLB3</p> <p>0: No clear operation or clear operation is completed 1: Enable clear operation</p> <p>After the Flush operation is completed, the bit can clear automatically.</p>
2	R/WAC	0x0	<p>MI_TLB2_FS Micro TLB2 Flush Clear Micro TLB2</p> <p>0: No clear operation or clear operation is completed</p>

Offset: 0x0080			Register Name: IOMMU_TLB_FLUSH_ENABLE_REG
Bit	Read/Write	Default/Hex	Description
			<p>1: Enable clear operation After the Flush operation is completed, the bit can clear automatically.</p>
1	R/WAC	0x0	<p>MI_TLB1_FS Micro TLB1 Flush Clear Micro TLB1 0: No clear operation or clear operation is completed 1: Enable clear operation After the Flush operation is completed, the bit can clear automatically. <b>Note: This bit is not used.</b></p>
0	R/WAC	0x0	<p>MI_TLB0_FS Micro TLB0 Flush Clear Micro TLB0 0: No clear operation or clear operation is completed 1: Enable clear operation After the Flush operation is completed, the bit can clear automatically.</p>

### 3.11.6.12 0x0084 IOMMU TLB Invalidation Mode Select Register (Default Value: 0x0000\_0000)

Offset: 0x0084			Register Name: IOMMU_TLB_IVLD_MODE_SEL_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	<p>TLB_IVLD_MODE_SEL 0: Invalidate TLB by using the Mask mode 1: Invalidate TLB by using the Start and End mode</p>

### 3.11.6.13 0x0088 IOMMU TLB Invalidation Start Address Register (Default Value: 0x0000\_0000)

Offset: 0x0088			Register Name: IOMMU_TLB_IVLD_STA_ADDR_REG
Bit	Read/Write	Default/Hex	Description
31:12	R/W	0x0	<p>TLB_IVLD_STA_ADDR TLB invalid start address, 4 KB aligned.</p>
11:0	/	/	/

### 3.11.6.14 0x008C IOMMU TLB Invalid End Address Register (Default Value: 0x0000\_0000)

Offset: 0x008C			Register Name: IOMMU_TLB_IVLD_END_ADDR_REG
Bit	Read/Write	Default/Hex	Description
31:12	R/W	0x0	TLB_IVLD_END_ADDR TLB invalid end address, 4 KB aligned.
11:0	/	/	/

### 3.11.6.15 0x0090 IOMMU TLB Invalid Address Register (Default Value: 0x0000\_0000)

Offset: 0x0090			Register Name: IOMMU_TLB_IVLD_ADDR_REG
Bit	Read/Write	Default/Hex	Description
31:12	R/W	0x0	TLB_IVLD_ADDR TLB invalid address, 4 KB aligned
11:0	/	/	/

**Operation:**

- 1) Set the virtual address that needs to be operated in **IOMMU\_TLB\_IVLD\_ADDR\_REG**.
- 2) Set the mask of virtual address that needs to be operated in **IOMMU\_TLB\_IVLD\_ADDR\_MASK\_REG**.
- 3) Write '1' to **IOMMU\_TLB\_IVLD\_ENABLE\_REG[0]**.
- 4) Read **IOMMU\_TLB\_IVLD\_ENABLE\_REG[0]**, when it is '0', it indicates that invalidation behavior is finished.



**NOTE**

- When performing invalidation operation, TLB/Cache operation has not affected.
- After or before invalidation operation starts, there is no absolute relationship between the same address switch operation and invalidation operation.

### 3.11.6.16 0x0094 IOMMU TLB Invalid Address Mask Register (Default Value: 0x0000\_0000)

Offset: 0x0094			Register Name: IOMMU_TLB_IVLD_ADDR_MASK_REG
Bit	Read/Write	Default/Hex	Description
31:12	R/W	0x0	TLB_IVLD_ADDR_MASK TLB invalid address mask register, 4 KB aligned
11:0	/	/	/

### 3.11.6.17 0x0098 IOMMU TLB Invalidation Enable Register (Default Value: 0x0000\_0000)

Offset: 0x0098			Register Name: IOMMU_TLB_IVLD_ENABLE_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/WAC	0x0	<p>TLB_IVLD_ENABLE Enable TLB invalidation operation 0: No operation or operation is completed 1: Enable invalidation operation</p> <p>After invalidation operation is completed, the bit can clear automatically.</p> <p>When operating invalidation operation, TLB/Cache operation has not affected.</p> <p>After or before invalidation operation starts, there is no absolute relationship between the same address switch operation and invalidation operation.</p>

### 3.11.6.18 0x009C IOMMU PC Invalidation Mode Select Register (Default Value: 0x0000\_0000)

Offset: 0x009C			Register Name: IOMMU_PC_IVLD_MODE_SEL_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	<p>PC_IVLD_MODE_SEL PTW Cache Invalid Mode Select 0: Invalidate PTW by using the Mask mode 1: Invalidate PTW by using the Start and End mode</p>

### 3.11.6.19 0x00A0 IOMMU PC Invalidation Address Register (Default Value: 0x0000\_0000)

Offset: 0x00A0			Register Name: IOMMU_PC_IVLD_ADDR_REG
Bit	Read/Write	Default/Hex	Description
31:20	R/W	0x0	PC_IVLD_ADDR PTW Cache invalid address, 1 MB aligned.
19:0	/	/	/

**3.11.6.20 0x00A4 IOMMU PC Invalidation Start Address Register (Default Value: 0x0000\_0000)**

Offset: 0x00A4			Register Name: IOMMU_PC_IVLD_STA_ADDR_REG
Bit	Read/Write	Default/Hex	Description
31:20	R/W	0x0	PC_IVLD_SA PTW Cache invalid start address, 1 MB aligned.
19:0	/	/	/

**3.11.6.21 0x00A8 IOMMU PC Invalidation Enable Register (Default Value: 0x0000\_0000)**

Offset: 0x00A8			Register Name: IOMMU_PC_IVLD_ENABLE_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/WAC	0x0	PC_IVLD_ENABLE Enable PTW Cache invalidation operation 0: No operation or operation is completed 1: Enable invalidation operation After invalidation operation is completed, the bit can clear automatically. After or before invalidation operation starts, there is no absolute relationship between the same address switch operation and invalidation operation.

**3.11.6.22 0x00AC IOMMU PC Invalidation End Address Register (Default Value: 0x0000\_0000)**

Offset: 0x00AC			Register Name: IOMMU_PC_IVLD_END_ADDR_REG
Bit	Read/Write	Default/Hex	Description
31:20	R/W	0x0	PC_IVLD_EA PTW Cache invalid end address, 1 MB aligned.
19:0	/	/	/

**3.11.6.23 0x00B0 IOMMU Domain Authority Control 0 Register (Default Value: 0x0000\_0000)**

Software can set 15 different permission control types in IOMMU\_DM\_AUT\_CTRL\_REG0–7. A default access control type is DOMAIN0. The read/write operation of DOMAIN1–15 is unlimited by default.

Software needs to set the index of the permission control domain corresponding to the page table item in the bit[7:4] of the Level2 page table, the default value is 0 (use domain0), that is, the read/write operation is not controlled.

Setting REG\_ARD\_OVWT can mask the Domain control defined by IOMMU\_DM\_AUT\_CTRL\_REG0~7. All Level2 page table types are covered by the type of REG\_ARD\_OVWT. The read/write operation is permitted by default.

Offset: 0x00B0			Register Name: IOMMU_DM_AUT_CTRL0_REG
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29	R/W	0x0	<p>DM1_M6_WT_AUT_CTRL Domain1 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited</p> <p><b>Note: The bit is not used.</b></p>
28	R/W	0x0	<p>DM1_M6_RD_AUT_CTRL Domain1 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited</p> <p><b>Note: The bit is not used.</b></p>
27	R/W	0x0	<p>DM1_M5_WT_AUT_CTRL Domain1 write permission control for master5 0: The write-operation is permitted 1: The write-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
26	R/W	0x0	<p>DM1_M5_RD_AUT_CTRL Domain1 read permission control for master5 0: The read-operation is permitted 1: The read-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
25	R/W	0x0	<p>DM1_M4_WT_AUT_CTRL Domain1 write permission control for master4 0: The write-operation is permitted 1: The write-operation is prohibited</p>
24	R/W	0x0	<p>DM1_M4_RD_AUT_CTRL Domain1 read permission control for master4 0: The read-operation is permitted 1: The read-operation is prohibited</p>
23	R/W	0x0	<p>DM1_M3_WT_AUT_CTRL Domain1 write permission control for master3 0: The write-operation is permitted</p>

Offset: 0x00B0			Register Name: IOMMU_DM_AUT_CTRL0_REG
Bit	Read/Write	Default/Hex	Description
			1: The write-operation is prohibited
22	R/W	0x0	DM1_M3_RD_AUT_CTRL Domain1 read permission control for master3 0: The read-operation is permitted 1: The read-operation is prohibited
21	R/W	0x0	DM1_M2_WT_AUT_CTRL Domain1 write permission control for master2 0: The write-operation is permitted 1: The write-operation is prohibited
20	R/W	0x0	DM1_M2_RD_AUT_CTRL Domain1 read permission control for master2 0: The read-operation is permitted 1: The read-operation is prohibited
19	R/W	0x0	DM1_M1_WT_AUT_CTRL Domain1 write permission control for master1 0: The write-operation is permitted 1: The write-operation is prohibited <b>Note: This bit is not used.</b>
18	R/W	0x0	DM1_M1_RD_AUT_CTRL Domain1 read permission control for master1 0: The read-operation is permitted 1: The read-operation is prohibited <b>Note: This bit is not used.</b>
17	R/W	0x0	DM1_M0_WT_AUT_CTRL Domain1 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited
16	R/W	0x0	DM1_M0_RD_AUT_CTRL Domain1 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited
15:14	/	/	/
13	R	0x0	DM0_M6_WT_AUT_CTRL Domain0 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited

Offset: 0x00B0			Register Name: IOMMU_DM_AUT_CTRL0_REG
Bit	Read/Write	Default/Hex	Description
			<b>Note: The bit is not used.</b>
12	R	0x0	DM0_M6_RD_AUT_CTRL Domain0 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited <b>Note: The bit is not used.</b>
11	R	0x0	DM0_M5_WT_AUT_CTRL Domain0 write permission control for master5 0: The write-operation is permitted 1: The write-operation is prohibited <b>Note: This bit is not used.</b>
10	R	0x0	DM0_M5_RD_AUT_CTRL Domain0 read permission control for master5 0: The read-operation is permitted 1: The read-operation is prohibited <b>Note: This bit is not used.</b>
9	R	0x0	DM0_M4_WT_AUT_CTRL Domain0 write permission control for master4 0: The write-operation is permitted 1: The write-operation is prohibited
8	R	0x0	DM0_M4_RD_AUT_CTRL Domain0 read permission control for master4 0: The read-operation is permitted 1: The read-operation is prohibited
7	R	0x0	DM0_M3_WT_AUT_CTRL Domain0 write permission control for master3 0: The write-operation is permitted 1: The write-operation is prohibited
6	R	0x0	DM0_M3_RD_AUT_CTRL Domain0 read permission control for master3 0: The read-operation is permitted 1: The read-operation is prohibited
5	R	0x0	DM0_M2_WT_AUT_CTRL Domain0 write permission control for master2 0: The write-operation is permitted 1: The write-operation is prohibited

Offset: 0x00B0			Register Name: IOMMU_DM_AUT_CTRL0_REG
Bit	Read/Write	Default/Hex	Description
4	R	0x0	<p>DM0_M2_RD_AUT_CTRL Domain0 read permission control for master2 0: The read-operation is permitted 1: The read-operation is prohibited</p>
3	R	0x0	<p>DM0_M1_WT_AUT_CTRL Domain0 write permission control for master1 0: The write-operation is permitted 1: The write-operation is prohibited <b>Note: This bit is not used.</b></p>
2	R	0x0	<p>DM0_M1_RD_AUT_CTRL Domain0 read permission control for master1 0: The read-operation is permitted 1: The read-operation is prohibited <b>Note: This bit is not used.</b></p>
1	R	0x0	<p>DM0_M0_WT_AUT_CTRL Domain0 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited</p>
0	R	0x0	<p>DM0_M0_RD_AUT_CTRL Domain0 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited</p>

### 3.11.6.24 0x00B4 IOMMU Domain Authority Control 1 Register (Default Value: 0x0000\_0000)

Offset: 0x00B4			Register Name: IOMMU_DM_AUT_CTRL1_REG
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29	R/W	0x0	<p>DM3_M6_WT_AUT_CTRL Domain3 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited <b>Note: The bit is not used.</b></p>
28	R/W	0x0	<p>DM3_M6_RD_AUT_CTRL Domain3 read permission control for master6 0: The read-operation is permitted</p>

Offset: 0x00B4			Register Name: IOMMU_DM_AUT_CTRL1_REG
Bit	Read/Write	Default/Hex	Description
			<p>1: The read-operation is prohibited</p> <p><b>Note: The bit is not used.</b></p>
27	R/W	0x0	<p>DM3_M5_WT_AUT_CTRL</p> <p>Domain3 write permission control for master5</p> <p>0: The write-operation is permitted</p> <p>1: The write-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
26	R/W	0x0	<p>DM3_M5_RD_AUT_CTRL</p> <p>Domain3 read permission control for master5</p> <p>0: The read-operation is permitted</p> <p>1: The read-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
25	R/W	0x0	<p>DM3_M4_WT_AUT_CTRL</p> <p>Domain3 write permission control for master4</p> <p>0: The write-operation is permitted</p> <p>1: The write-operation is prohibited</p>
24	R/W	0x0	<p>DM3_M4_RD_AUT_CTRL</p> <p>Domain3 read permission control for master4</p> <p>0: The read-operation is permitted</p> <p>1: The read-operation is prohibited</p>
23	R/W	0x0	<p>DM3_M3_WT_AUT_CTRL</p> <p>Domain3 write permission control for master3</p> <p>0: The write-operation is permitted</p> <p>1: The write-operation is prohibited</p>
22	R/W	0x0	<p>DM3_M3_RD_AUT_CTRL</p> <p>Domain3 read permission control for master3</p> <p>0: The read-operation is permitted</p> <p>1: The read-operation is prohibited</p>
21	R/W	0x0	<p>DM3_M2_WT_AUT_CTRL</p> <p>Domain3 write permission control for master2</p> <p>0: The write-operation is permitted</p> <p>1: The write-operation is prohibited</p>
20	R/W	0x0	<p>DM3_M2_RD_AUT_CTRL</p> <p>Domain3 read permission control for master2</p> <p>0: The read-operation is permitted</p> <p>1: The read-operation is prohibited</p>

Offset: 0x00B4			Register Name: IOMMU_DM_AUT_CTRL1_REG
Bit	Read/Write	Default/Hex	Description
19	R/W	0x0	<p>DM3_M1_WT_AUT_CTRL Domain3 write permission control for master1 0: The write-operation is permitted 1: The write-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
18	R/W	0x0	<p>DM3_M1_RD_AUT_CTRL Domain3 read permission control for master1 0: The read-operation is permitted 1: The read-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
17	R/W	0x0	<p>DM3_M0_WT_AUT_CTRL Domain3 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited</p>
16	R/W	0x0	<p>DM3_M0_RD_AUT_CTRL Domain3 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited</p>
15:14	/	/	/
13	R/W	0x0	<p>DM2_M6_WT_AUT_CTRL Domain2 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited</p> <p><b>Note: The bit is not used.</b></p>
12	R/W	0x0	<p>DM2_M6_RD_AUT_CTRL Domain2 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited</p> <p><b>Note: The bit is not used.</b></p>
11	R/W	0x0	<p>DM2_M5_WT_AUT_CTRL Domain2 write permission control for master5 0: The write-operation is permitted 1: The write-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
10	R/W	0x0	<p>DM2_M5_RD_AUT_CTRL Domain2 read permission control for master5 0: The read-operation is permitted</p>

Offset: 0x00B4			Register Name: IOMMU_DM_AUT_CTRL1_REG
Bit	Read/Write	Default/Hex	Description
			<p>1: The read-operation is prohibited  <b>Note: This bit is not used.</b></p>
9	R/W	0x0	<p>DM2_M4_WT_AUT_CTRL          Domain2 write permission control for master4          0: The write-operation is permitted          1: The write-operation is prohibited</p>
8	R/W	0x0	<p>DM2_M4_RD_AUT_CTRL          Domain2 read permission control for master4          0: The read-operation is permitted          1: The read-operation is prohibited</p>
7	R/W	0x0	<p>DM2_M3_WT_AUT_CTRL          Domain2 write permission control for master3          0: The write-operation is permitted          1: The write-operation is prohibited</p>
6	R/W	0x0	<p>DM2_M3_RD_AUT_CTRL          Domain2 read permission control for master3          0: The read-operation is permitted          1: The read-operation is prohibited</p>
5	R/W	0x0	<p>DM2_M2_WT_AUT_CTRL          Domain2 write permission control for master2          0: The write-operation is permitted          1: The write-operation is prohibited</p>
4	R/W	0x0	<p>DM2_M2_RD_AUT_CTRL          Domain2 read permission control for master2          0: The read-operation is permitted          1: The read-operation is prohibited</p>
3	R/W	0x0	<p>DM2_M1_WT_AUT_CTRL          Domain2 write permission control for master1          0: The write-operation is permitted          1: The write-operation is prohibited  <b>Note: This bit is not used.</b></p>
2	R/W	0x0	<p>DM2_M1_RD_AUT_CTRL          Domain2 read permission control for master1          0: The read-operation is permitted          1: The read-operation is prohibited  <b>Note: This bit is not used.</b></p>

Offset: 0x00B4			Register Name: IOMMU_DM_AUT_CTRL1_REG
Bit	Read/Write	Default/Hex	Description
1	R/W	0x0	<p>DM2_M0_WT_AUT_CTRL Domain2 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited</p>
0	R/W	0x0	<p>DM2_M0_RD_AUT_CTRL Domain2 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited</p>

### 3.11.6.25 0x00B8 IOMMU Domain Authority Control 2 Register (Default Value: 0x0000\_0000)

Offset: 0x00B8			Register Name: IOMMU_DM_AUT_CTRL2_REG
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29	R/W	0x0	<p>DM5_M6_WT_AUT_CTRL Domain5 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited <b>Note: The bit is not used.</b></p>
28	R/W	0x0	<p>DM5_M6_RD_AUT_CTRL Domain5 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited <b>Note: The bit is not used.</b></p>
27	R/W	0x0	<p>DM5_M5_WT_AUT_CTRL Domain5 write permission control for master5 0: The write-operation is permitted 1: The write-operation is prohibited <b>Note: This bit is not used.</b></p>
26	R/W	0x0	<p>DM5_M5_RD_AUT_CTRL Domain5 read permission control for master5 0: The read-operation is permitted 1: The read-operation is prohibited <b>Note: This bit is not used.</b></p>
25	R/W	0x0	<p>DM5_M4_WT_AUT_CTRL Domain5 write permission control for master4</p>

Offset: 0x00B8			Register Name: IOMMU_DM_AUT_CTRL2_REG
Bit	Read/Write	Default/Hex	Description
			0: The write-operation is permitted 1: The write-operation is prohibited
24	R/W	0x0	DM5_M4_RD_AUT_CTRL Domain5 read permission control for master4 0: The read-operation is permitted 1: The read-operation is prohibited
23	R/W	0x0	DM5_M3_WT_AUT_CTRL Domain5 write permission control for master3 0: The write-operation is permitted 1: The write-operation is prohibited
22	R/W	0x0	DM5_M3_RD_AUT_CTRL Domain5 read permission control for master3 0: The read-operation is permitted 1: The read-operation is prohibited
21	R/W	0x0	DM5_M2_WT_AUT_CTRL Domain5 write permission control for master2 0: The write-operation is permitted 1: The write-operation is prohibited
20	R/W	0x0	DM5_M2_RD_AUT_CTRL Domain5 read permission control for master2 0: The read-operation is permitted 1: The read-operation is prohibited
19	R/W	0x0	DM5_M1_WT_AUT_CTRL Domain5 write permission control for master1 0: The write-operation is permitted 1: The write-operation is prohibited <b>Note: This bit is not used.</b>
18	R/W	0x0	DM5_M1_RD_AUT_CTRL Domain5 read permission control for master1 0: The read-operation is permitted 1: The read-operation is prohibited <b>Note: This bit is not used.</b>
17	R/W	0x0	DM5_M0_WT_AUT_CTRL Domain5 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited

Offset: 0x00B8			Register Name: IOMMU_DM_AUT_CTRL2_REG
Bit	Read/Write	Default/Hex	Description
16	R/W	0x0	<p>DM5_M0_RD_AUT_CTRL Domain5 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited</p>
15:14	/	/	/
13	R/W	0x0	<p>DM4_M6_WT_AUT_CTRL Domain4 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited <b>Note: The bit is not used.</b></p>
12	R/W	0x0	<p>DM4_M6_RD_AUT_CTRL Domain4 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited <b>Note: The bit is not used.</b></p>
11	R/W	0x0	<p>DM4_M5_WT_AUT_CTRL Domain4 write permission control for master5 0: The write-operation is permitted 1: The write-operation is prohibited <b>Note: This bit is not used.</b></p>
10	R/W	0x0	<p>DM4_M5_RD_AUT_CTRL Domain4 read permission control for master5 0: The read-operation is permitted 1: The read-operation is prohibited <b>Note: This bit is not used.</b></p>
9	R/W	0x0	<p>DM4_M4_WT_AUT_CTRL Domain4 write permission control for master4 0: The write-operation is permitted 1: The write-operation is prohibited</p>
8	R/W	0x0	<p>DM4_M4_RD_AUT_CTRL Domain4 read permission control for master4 0: The read-operation is permitted 1: The read-operation is prohibited</p>
7	R/W	0x0	<p>DM4_M3_WT_AUT_CTRL Domain4 write permission control for master3 0: The write-operation is permitted 1: The write-operation is prohibited</p>

Offset: 0x00B8			Register Name: IOMMU_DM_AUT_CTRL2_REG
Bit	Read/Write	Default/Hex	Description
6	R/W	0x0	<p>DM4_M3_RD_AUT_CTRL Domain4 read permission control for master3 0: The read-operation is permitted 1: The read-operation is prohibited</p>
5	R/W	0x0	<p>DM4_M2_WT_AUT_CTRL Domain4 write permission control for master2 0: The write-operation is permitted 1: The write-operation is prohibited</p>
4	R/W	0x0	<p>DM4_M2_RD_AUT_CTRL Domain4 read permission control for master2 0: The read-operation is permitted 1: The read-operation is prohibited</p>
3	R/W	0x0	<p>DM4_M1_WT_AUT_CTRL Domain4 write permission control for master1 0: The write-operation is permitted 1: The write-operation is prohibited <b>Note: This bit is not used.</b></p>
2	R/W	0x0	<p>DM4_M1_RD_AUT_CTRL Domain4 read permission control for master1 0: The read-operation is permitted 1: The read-operation is prohibited <b>Note: This bit is not used.</b></p>
1	R/W	0x0	<p>DM4_M0_WT_AUT_CTRL Domain4 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited</p>
0	R/W	0x0	<p>DM4_M0_RD_AUT_CTRL Domain4 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited</p>

### 3.11.6.26 0x00BC IOMMU Domain Authority Control 3 Register (Default Value: 0x0000\_0000)

Offset: 0x00BC			Register Name: IOMMU_DM_AUT_CTRL3_REG
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/

Offset: 0x00BC			Register Name: IOMMU_DM_AUT_CTRL3_REG
Bit	Read/Write	Default/Hex	Description
29	R/W	0x0	<p>DM7_M6_WT_AUT_CTRL Domain7 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited</p> <p><b>Note: The bit is not used.</b></p>
28	R/W	0x0	<p>DM7_M6_RD_AUT_CTRL Domain7 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited</p> <p><b>Note: The bit is not used.</b></p>
27	R/W	0x0	<p>DM7_M5_WT_AUT_CTRL Domain7 write permission control for master5 0: The write-operation is permitted 1: The write-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
26	R/W	0x0	<p>DM7_M5_RD_AUT_CTRL Domain7 read permission control for master5 0: The read-operation is permitted 1: The read-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
25	R/W	0x0	<p>DM7_M4_WT_AUT_CTRL Domain7 write permission control for master4 0: The write-operation is permitted 1: The write-operation is prohibited</p>
24	R/W	0x0	<p>DM7_M4_RD_AUT_CTRL Domain7 read permission control for master4 0: The read-operation is permitted 1: The read-operation is prohibited</p>
23	R/W	0x0	<p>DM7_M3_WT_AUT_CTRL Domain7 write permission control for master3 0: The write-operation is permitted 1: The write-operation is prohibited</p>
22	R/W	0x0	<p>DM7_M3_RD_AUT_CTRL Domain7 read permission control for master3 0: The read-operation is permitted 1: The read-operation is prohibited</p>
21	R/W	0x0	DM7_M2_WT_AUT_CTRL

Offset: 0x00BC			Register Name: IOMMU_DM_AUT_CTRL3_REG
Bit	Read/Write	Default/Hex	Description
			Domain7 write permission control for master2 0: The write-operation is permitted 1: The write-operation is prohibited
20	R/W	0x0	DM7_M2_RD_AUT_CTRL Domain7 read permission control for master2 0: The read-operation is permitted 1: The read-operation is prohibited
19	R/W	0x0	DM7_M1_WT_AUT_CTRL Domain7 write permission control for master1 0: The write-operation is permitted 1: The write-operation is prohibited <b>Note: This bit is not used.</b>
18	R/W	0x0	DM7_M1_RD_AUT_CTRL Domain7 read permission control for master1 0: The read-operation is permitted 1: The read-operation is prohibited <b>Note: This bit is not used.</b>
17	R/W	0x0	DM7_M0_WT_AUT_CTRL Domain7 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited
16	R/W	0x0	DM7_M0_RD_AUT_CTRL Domain7 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited
15:14	/	/	/
13	R/W	0x0	DM6_M6_WT_AUT_CTRL Domain6 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited <b>Note: The bit is not used.</b>
12	R/W	0x0	DM6_M6_RD_AUT_CTRL Domain6 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited <b>Note: The bit is not used.</b>

Offset: 0x00BC			Register Name: IOMMU_DM_AUT_CTRL3_REG
Bit	Read/Write	Default/Hex	Description
11	R/W	0x0	<p>DM6_M5_WT_AUT_CTRL Domain6 write permission control for master5 0: The write-operation is permitted 1: The write-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
10	R/W	0x0	<p>DM6_M5_RD_AUT_CTRL Domain6 read permission control for master5 0: The read-operation is permitted 1: The read-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
9	R/W	0x0	<p>DM6_M4_WT_AUT_CTRL Domain6 write permission control for master4 0: The write-operation is permitted 1: The write-operation is prohibited</p>
8	R/W	0x0	<p>DM6_M4_RD_AUT_CTRL Domain6 read permission control for master4 0: The read-operation is permitted 1: The read-operation is prohibited</p>
7	R/W	0x0	<p>DM6_M3_WT_AUT_CTRL Domain6 write permission control for master3 0: The write-operation is permitted 1: The write-operation is prohibited</p>
6	R/W	0x0	<p>DM6_M3_RD_AUT_CTRL Domain6 read permission control for master3 0: The read-operation is permitted 1: The read-operation is prohibited</p>
5	R/W	0x0	<p>DM6_M2_WT_AUT_CTRL Domain6 write permission control for master2 0: The write-operation is permitted 1: The write-operation is prohibited</p>
4	R/W	0x0	<p>DM6_M2_RD_AUT_CTRL Domain6 read permission control for master2 0: The read-operation is permitted 1: The read-operation is prohibited</p>
3	R/W	0x0	<p>DM6_M1_WT_AUT_CTRL Domain6 write permission control for master1 0: The write-operation is permitted</p>

Offset: 0x00BC			Register Name: IOMMU_DM_AUT_CTRL3_REG
Bit	Read/Write	Default/Hex	Description
			<p>1: The write-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
2	R/W	0x0	<p>DM6_M1_RD_AUT_CTRL Domain6 read permission control for master1</p> <p>0: The read-operation is permitted</p> <p>1: The read-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
1	R/W	0x0	<p>DM6_M0_WT_AUT_CTRL Domain6 write permission control for master0</p> <p>0: The write-operation is permitted</p> <p>1: The write-operation is prohibited</p>
0	R/W	0x0	<p>DM6_M0_RD_AUT_CTRL Domain6 read permission control for master0</p> <p>0: The read-operation is permitted</p> <p>1: The read-operation is prohibited</p>

### 3.11.6.27 0x00C0 IOMMU Domain Authority Control 4 Register (Default Value: 0x0000\_0000)

Offset: 0x00C0			Register Name: IOMMU_DM_AUT_CTRL4_REG
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29	R/W	0x0	<p>DM9_M6_WT_AUT_CTRL Domain9 write permission control for master6</p> <p>0: The write-operation is permitted</p> <p>1: The write-operation is prohibited</p> <p><b>Note: The bit is not used.</b></p>
28	R/W	0x0	<p>DM9_M6_RD_AUT_CTRL Domain9 read permission control for master6</p> <p>0: The read-operation is permitted</p> <p>1: The read-operation is prohibited</p> <p><b>Note: The bit is not used.</b></p>
27	R/W	0x0	<p>DM9_M5_WT_AUT_CTRL Domain9 write permission control for master5</p> <p>0: The write-operation is permitted</p> <p>1: The write-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>

Offset: 0x00C0			Register Name: IOMMU_DM_AUT_CTRL4_REG
Bit	Read/Write	Default/Hex	Description
26	R/W	0x0	<p>DM9_M5_RD_AUT_CTRL Domain9 read permission control for master5 0: The read-operation is permitted 1: The read-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
25	R/W	0x0	<p>DM9_M4_WT_AUT_CTRL Domain9 write permission control for master4 0: The write-operation is permitted 1: The write-operation is prohibited</p>
24	R/W	0x0	<p>DM9_M4_RD_AUT_CTRL Domain9 read permission control for master4 0: The read-operation is permitted 1: The read-operation is prohibited</p>
23	R/W	0x0	<p>DM9_M3_WT_AUT_CTRL Domain9 write permission control for master3 0: The write-operation is permitted 1: The write-operation is prohibited</p>
22	R/W	0x0	<p>DM9_M3_RD_AUT_CTRL Domain9 read permission control for master3 0: The read-operation is permitted 1: The read-operation is prohibited</p>
21	R/W	0x0	<p>DM9_M2_WT_AUT_CTRL Domain9 write permission control for master2 0: The write-operation is permitted 1: The write-operation is prohibited</p>
20	R/W	0x0	<p>DM9_M2_RD_AUT_CTRL Domain9 read permission control for master2 0: The read-operation is permitted 1: The read-operation is prohibited</p>
19	R/W	0x0	<p>DM9_M1_WT_AUT_CTRL Domain9 write permission control for master1 0: The write-operation is permitted 1: The write-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
18	R/W	0x0	<p>DM9_M1_RD_AUT_CTRL Domain9 read permission control for master1 0: The read-operation is permitted</p>

Offset: 0x00C0			Register Name: IOMMU_DM_AUT_CTRL4_REG
Bit	Read/Write	Default/Hex	Description
			1: The read-operation is prohibited <b>Note: This bit is not used.</b>
17	R/W	0x0	DM9_M0_WT_AUT_CTRL Domain9 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited
16	R/W	0x0	DM9_M0_RD_AUT_CTRL Domain9 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited
15:14	/	/	/
13	R/W	0x0	DM8_M6_WT_AUT_CTRL Domain8 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited <b>Note: The bit is not used.</b>
12	R/W	0x0	DM8_M6_RD_AUT_CTRL Domain8 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited <b>Note: The bit is not used.</b>
11	R/W	0x0	DM8_M5_WT_AUT_CTRL Domain8 write permission control for master5 0: The write-operation is permitted 1: The write-operation is prohibited <b>Note: This bit is not used.</b>
10	R/W	0x0	DM8_M5_RD_AUT_CTRL Domain8 read permission control for master5 0: The read-operation is permitted 1: The read-operation is prohibited <b>Note: This bit is not used.</b>
9	R/W	0x0	DM8_M4_WT_AUT_CTRL Domain8 write permission control for master4 0: The write-operation is permitted 1: The write-operation is prohibited
8	R/W	0x0	DM8_M4_RD_AUT_CTRL

Offset: 0x00C0			Register Name: IOMMU_DM_AUT_CTRL4_REG
Bit	Read/Write	Default/Hex	Description
			Domain8 read permission control for master4 0: The read-operation is permitted 1: The read-operation is prohibited
7	R/W	0x0	DM8_M3_WT_AUT_CTRL Domain8 write permission control for master3 0: The write-operation is permitted 1: The write-operation is prohibited
6	R/W	0x0	DM8_M3_RD_AUT_CTRL Domain8 read permission control for master3 0: The read-operation is permitted 1: The read-operation is prohibited
5	R/W	0x0	DM8_M2_WT_AUT_CTRL Domain8 write permission control for master2 0: The write-operation is permitted 1: The write-operation is prohibited
4	R/W	0x0	DM8_M2_RD_AUT_CTRL Domain8 read permission control for master2 0: The read-operation is permitted 1: The read-operation is prohibited
3	R/W	0x0	DM8_M1_WT_AUT_CTRL Domain8 write permission control for master1 0: The write-operation is permitted 1: The write-operation is prohibited <b>Note: This bit is not used.</b>
2	R/W	0x0	DM8_M1_RD_AUT_CTRL Domain8 read permission control for master1 0: The read-operation is permitted 1: The read-operation is prohibited <b>Note: This bit is not used.</b>
1	R/W	0x0	DM8_M0_WT_AUT_CTRL Domain8 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited
0	R/W	0x0	DM8_M0_RD_AUT_CTRL Domain8 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited

### 3.11.6.28 0x00C4 IOMMU Domain Authority Control 5 Register (Default Value: 0x0000\_0000)

Offset: 0x00C4			Register Name: IOMMU_DM_AUT_CTRL5_REG
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29	R/W	0x0	<p>DM11_M6_WT_AUT_CTRL Domain11 write permission control for master6</p> <p>0: The write-operation is permitted 1: The write-operation is prohibited</p> <p><b>Note: The bit is not used.</b></p>
28	R/W	0x0	<p>DM11_M6_RD_AUT_CTRL Domain11 read permission control for master6</p> <p>0: The read-operation is permitted 1: The read-operation is prohibited</p> <p><b>Note: The bit is not used.</b></p>
27	R/W	0x0	<p>DM11_M5_WT_AUT_CTRL Domain11 write permission control for master5</p> <p>0: The write-operation is permitted 1: The write-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
26	R/W	0x0	<p>DM11_M5_RD_AUT_CTRL Domain11 read permission control for master5</p> <p>0: The read-operation is permitted 1: The read-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
25	R/W	0x0	<p>DM11_M4_WT_AUT_CTRL Domain11 write permission control for master4</p> <p>0: The write-operation is permitted 1: The write-operation is prohibited</p>
24	R/W	0x0	<p>DM11_M4_RD_AUT_CTRL Domain11 read permission control for master4</p> <p>0: The read-operation is permitted 1: The read-operation is prohibited</p>
23	R/W	0x0	<p>DM11_M3_WT_AUT_CTRL Domain11 write permission control for master3</p> <p>0: The write-operation is permitted 1: The write-operation is prohibited</p>

Offset: 0x00C4			Register Name: IOMMU_DM_AUT_CTRL5_REG
Bit	Read/Write	Default/Hex	Description
22	R/W	0x0	<p>DM11_M3_RD_AUT_CTRL Domain11 read permission control for master3 0: The read-operation is permitted 1: The read-operation is prohibited</p>
21	R/W	0x0	<p>DM11_M2_WT_AUT_CTRL Domain11 write permission control for master2 0: The write-operation is permitted 1: The write-operation is prohibited</p>
20	R/W	0x0	<p>DM11_M2_RD_AUT_CTRL Domain11 read permission control for master2 0: The read-operation is permitted 1: The read-operation is prohibited</p>
19	R/W	0x0	<p>DM11_M1_WT_AUT_CTRL Domain11 write permission control for master1 0: The write-operation is permitted 1: The write-operation is prohibited <b>Note: This bit is not used.</b></p>
18	R/W	0x0	<p>DM11_M1_RD_AUT_CTRL Domain11 read permission control for master1 0: The read-operation is permitted 1: The read-operation is prohibited <b>Note: This bit is not used.</b></p>
17	R/W	0x0	<p>DM11_M0_WT_AUT_CTRL Domain11 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited</p>
16	R/W	0x0	<p>DM11_M0_RD_AUT_CTRL Domain11 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited</p>
15:14	/	/	/
13	R/W	0x0	<p>DM10_M6_WT_AUT_CTRL Domain10 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited <b>Note: The bit is not used.</b></p>

Offset: 0x00C4			Register Name: IOMMU_DM_AUT_CTRL5_REG
Bit	Read/Write	Default/Hex	Description
12	R/W	0x0	<p>DM10_M6_RD_AUT_CTRL Domain10 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited</p> <p><b>Note: The bit is not used.</b></p>
11	R/W	0x0	<p>DM10_M5_WT_AUT_CTRL Domain10 write permission control for master5 0: The write-operation is permitted 1: The write-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
10	R/W	0x0	<p>DM10_M5_RD_AUT_CTRL Domain10 read permission control for master5 0: The read-operation is permitted 1: The read-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
9	R/W	0x0	<p>DM10_M4_WT_AUT_CTRL Domain10 write permission control for master4 0: The write-operation is permitted 1: The write-operation is prohibited</p>
8	R/W	0x0	<p>DM10_M4_RD_AUT_CTRL Domain10 read permission control for master4 0: The read-operation is permitted 1: The read-operation is prohibited</p>
7	R/W	0x0	<p>DM10_M3_WT_AUT_CTRL Domain10 write permission control for master3 0: The write-operation is permitted 1: The write-operation is prohibited</p>
6	R/W	0x0	<p>DM10_M3_RD_AUT_CTRL Domain10 read permission control for master3 0: The read-operation is permitted 1: The read-operation is prohibited</p>
5	R/W	0x0	<p>DM10_M2_WT_AUT_CTRL Domain10 write permission control for master2 0: The write-operation is permitted 1: The write-operation is prohibited</p>
4	R/W	0x0	<p>DM10_M2_RD_AUT_CTRL Domain10 read permission control for master2</p>

Offset: 0x00C4			Register Name: IOMMU_DM_AUT_CTRL5_REG
Bit	Read/Write	Default/Hex	Description
			0: The read-operation is permitted 1: The read-operation is prohibited
3	R/W	0x0	DM10_M1_WT_AUT_CTRL Domain10 write permission control for master1 0: The write-operation is permitted 1: The write-operation is prohibited <b>Note: This bit is not used.</b>
2	R/W	0x0	DM10_M1_RD_AUT_CTRL Domain10 read permission control for master1 0: The read-operation is permitted 1: The read-operation is prohibited <b>Note: This bit is not used.</b>
1	R/W	0x0	DM10_M0_WT_AUT_CTRL Domain10 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited
0	R/W	0x0	DM10_M0_RD_AUT_CTRL Domain10 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited

### 3.11.6.29 0x00C8 IOMMU Domain Authority Control 6 Register (Default Value: 0x0000\_0000)

Offset: 0x00C8			Register Name: IOMMU_DM_AUT_CTRL6_REG
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29	R/W	0x0	DM13_M6_WT_AUT_CTRL Domain13 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited
28	R/W	0x0	DM13_M6_RD_AUT_CTRL Domain13 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited
27	R/W	0x0	DM13_M5_WT_AUT_CTRL Domain13 write permission control for master5

Offset: 0x00C8			Register Name: IOMMU_DM_AUT_CTRL6_REG
Bit	Read/Write	Default/Hex	Description
			<p>0: The write-operation is permitted 1: The write-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
26	R/W	0x0	<p>DM13_M5_RD_AUT_CTRL Domain13 read permission control for master5 0: The read-operation is permitted 1: The read-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
25	R/W	0x0	<p>DM13_M4_WT_AUT_CTRL Domain13 write permission control for master4 0: The write-operation is permitted 1: The write-operation is prohibited</p>
24	R/W	0x0	<p>DM13_M4_RD_AUT_CTRL Domain13 read permission control for master4 0: The read-operation is permitted 1: The read-operation is prohibited</p>
23	R/W	0x0	<p>DM13_M3_WT_AUT_CTRL Domain13 write permission control for master3 0: The write-operation is permitted 1: The write-operation is prohibited</p>
22	R/W	0x0	<p>DM13_M3_RD_AUT_CTRL Domain13 read permission control for master3 0: The read-operation is permitted 1: The read-operation is prohibited</p>
21	R/W	0x0	<p>DM13_M2_WT_AUT_CTRL Domain13 write permission control for master2 0: The write-operation is permitted 1: The write-operation is prohibited</p>
20	R/W	0x0	<p>DM13_M2_RD_AUT_CTRL Domain13 read permission control for master2 0: The read-operation is permitted 1: The read-operation is prohibited</p>
19	R/W	0x0	<p>DM13_M1_WT_AUT_CTRL Domain13 write permission control for master1 0: The write-operation is permitted 1: The write-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>

Offset: 0x00C8			Register Name: IOMMU_DM_AUT_CTRL6_REG
Bit	Read/Write	Default/Hex	Description
18	R/W	0x0	<p>DM13_M1_RD_AUT_CTRL Domain13 read permission control for master1 0: The read-operation is permitted 1: The read-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
17	R/W	0x0	<p>DM13_M0_WT_AUT_CTRL Domain13 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited</p>
16	R/W	0x0	<p>DM13_M0_RD_AUT_CTRL Domain13 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited</p>
15:14	/	/	/
13	R/W	0x0	<p>DM12_M6_WT_AUT_CTRL Domain12 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
12	R/W	0x0	<p>DM12_M6_RD_AUT_CTRL Domain12 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
11	R/W	0x0	<p>DM12_M5_WT_AUT_CTRL Domain12 write permission control for master5 0: The write-operation is permitted 1: The write-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
10	R/W	0x0	<p>DM12_M5_RD_AUT_CTRL Domain12 read permission control for master5 0: The read-operation is permitted 1: The read-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
9	R/W	0x0	<p>DM12_M4_WT_AUT_CTRL Domain12 write permission control for master4 0: The write-operation is permitted</p>

Offset: 0x00C8			Register Name: IOMMU_DM_AUT_CTRL6_REG
Bit	Read/Write	Default/Hex	Description
			1: The write-operation is prohibited
8	R/W	0x0	<p>DM12_M4_RD_AUT_CTRL Domain12 read permission control for master4</p> <p>0: The read-operation is permitted 1: The read-operation is prohibited</p>
7	R/W	0x0	<p>DM12_M3_WT_AUT_CTRL Domain12 write permission control for master3</p> <p>0: The write-operation is permitted 1: The write-operation is prohibited</p>
6	R/W	0x0	<p>DM12_M3_RD_AUT_CTRL Domain12 read permission control for master3</p> <p>0: The read-operation is permitted 1: The read-operation is prohibited</p>
5	R/W	0x0	<p>DM12_M2_WT_AUT_CTRL Domain12 write permission control for master2</p> <p>0: The write-operation is permitted 1: The write-operation is prohibited</p>
4	R/W	0x0	<p>DM12_M2_RD_AUT_CTRL Domain12 read permission control for master2</p> <p>0: The read-operation is permitted 1: The read-operation is prohibited</p>
3	R/W	0x0	<p>DM12_M1_WT_AUT_CTRL Domain12 write permission control for master1</p> <p>0: The write-operation is permitted 1: The write-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
2	R/W	0x0	<p>DM12_M1_RD_AUT_CTRL Domain12 read permission control for master1</p> <p>0: The read-operation is permitted 1: The read-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
1	R/W	0x0	<p>DM12_M0_WT_AUT_CTRL Domain12 write permission control for master0</p> <p>0: The write-operation is permitted 1: The write-operation is prohibited</p>
0	R/W	0x0	DM12_M0_RD_AUT_CTRL

Offset: 0x00C8			Register Name: IOMMU_DM_AUT_CTRL6_REG
Bit	Read/Write	Default/Hex	Description
			Domain12 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited

### 3.11.6.30 0x00CC IOMMU Domain Authority Control 7 Register (Default Value: 0x0000\_0000)

Offset: 0x00CC			Register Name: IOMMU_DM_AUT_CTRL7_REG
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29	R/W	0x0	DM15_M6_WT_AUT_CTRL Domain15 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited <b>Note: The bit is not used.</b>
28	R/W	0x0	DM15_M6_RD_AUT_CTRL Domain15 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited <b>Note: The bit is not used.</b>
27	R/W	0x0	DM15_M5_WT_AUT_CTRL Domain15 write permission control for master5 0: The write-operation is permitted 1: The write-operation is prohibited <b>Note: This bit is not used.</b>
26	R/W	0x0	DM15_M5_RD_AUT_CTRL Domain15 read permission control for master5 0: The read-operation is permitted 1: The read-operation is prohibited <b>Note: This bit is not used.</b>
25	R/W	0x0	DM15_M4_WT_AUT_CTRL Domain15 write permission control for master4 0: The write-operation is permitted 1: The write-operation is prohibited
24	R/W	0x0	DM15_M4_RD_AUT_CTRL Domain15 read permission control for master4 0: The read-operation is permitted

Offset: 0x00CC			Register Name: IOMMU_DM_AUT_CTRL7_REG
Bit	Read/Write	Default/Hex	Description
			1: The read-operation is prohibited
23	R/W	0x0	DM15_M3_WT_AUT_CTRL Domain15 write permission control for master3 0: The write-operation is permitted 1: The write-operation is prohibited
22	R/W	0x0	DM15_M3_RD_AUT_CTRL Domain15 read permission control for master3 0: The read-operation is permitted 1: The read-operation is prohibited
21	R/W	0x0	DM15_M2_WT_AUT_CTRL Domain15 write permission control for master2 0: The write-operation is permitted 1: The write-operation is prohibited
20	R/W	0x0	DM15_M2_RD_AUT_CTRL Domain15 read permission control for master2 0: The read-operation is permitted 1: The read-operation is prohibited
19	R/W	0x0	DM15_M1_WT_AUT_CTRL Domain15 write permission control for master1 0: The write-operation is permitted 1: The write-operation is prohibited <b>Note: This bit is not used.</b>
18	R/W	0x0	DM15_M1_RD_AUT_CTRL Domain15 read permission control for master1 0: The read-operation is permitted 1: The read-operation is prohibited <b>Note: This bit is not used.</b>
17	R/W	0x0	DM15_M0_WT_AUT_CTRL Domain15 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited
16	R/W	0x0	DM15_M0_RD_AUT_CTRL Domain15 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited
15:14	/	/	/

Offset: 0x00CC			Register Name: IOMMU_DM_AUT_CTRL7_REG
Bit	Read/Write	Default/Hex	Description
13	R/W	0x0	<p>DM14_M6_WT_AUT_CTRL Domain14 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited</p> <p><b>Note: The bit is not used.</b></p>
12	R/W	0x0	<p>DM14_M6_RD_AUT_CTRL Domain14 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited</p> <p><b>Note: The bit is not used.</b></p>
11	R/W	0x0	<p>DM14_M5_WT_AUT_CTRL Domain14 write permission control for master5 0: The write-operation is permitted 1: The write-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
10	R/W	0x0	<p>DM14_M5_RD_AUT_CTRL Domain14 read permission control for master5 0: The read-operation is permitted 1: The read-operation is prohibited</p> <p><b>Note: This bit is not used.</b></p>
9	R/W	0x0	<p>DM14_M4_WT_AUT_CTRL Domain14 write permission control for master4 0: The write-operation is permitted 1: The write-operation is prohibited</p>
8	R/W	0x0	<p>DM14_M4_RD_AUT_CTRL Domain14 read permission control for master4 0: The read-operation is permitted 1: The read-operation is prohibited</p>
7	R/W	0x0	<p>DM14_M3_WT_AUT_CTRL Domain14 write permission control for master3 0: The write-operation is permitted 1: The write-operation is prohibited</p>
6	R/W	0x0	<p>DM14_M3_RD_AUT_CTRL Domain14 read permission control for master3 0: The read-operation is permitted 1: The read-operation is prohibited</p>
5	R/W	0x0	DM14_M2_WT_AUT_CTRL

Offset: 0x00CC			Register Name: IOMMU_DM_AUT_CTRL7_REG
Bit	Read/Write	Default/Hex	Description
			Domain14 write permission control for master2 0: The write-operation is permitted 1: The write-operation is prohibited
4	R/W	0x0	DM14_M2_RD_AUT_CTRL Domain14 read permission control for master2 0: The read-operation is permitted 1: The read-operation is prohibited
3	R/W	0x0	DM14_M1_WT_AUT_CTRL Domain14 write permission control for master1 0: The write-operation is permitted 1: The write-operation is prohibited <b>Note: This bit is not used.</b>
2	R/W	0x0	DM14_M1_RD_AUT_CTRL Domain14 read permission control for master1 0: The read-operation is permitted 1: The read-operation is prohibited <b>Note: This bit is not used.</b>
1	R/W	0x0	DM14_M0_WT_AUT_CTRL Domain14 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited
0	R/W	0x0	DM14_M0_RD_AUT_CTRL Domain14 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited

### 3.11.6.31 0x00D0 IOMMU Domain Authority Overwrite Register (Default Value: 0x0000\_0000)

Setting the REG\_ARD\_OVWT can mask the Domain control defined by IOMMU\_DM\_AUT\_CTRL\_REG0–7. All the property of Level2 are covered by the property defined in REG\_ARD\_OVWT. Allow read and write for all by default.

Offset: 0x00D0			Register Name: IOMMU_DM_AUT_OVWT_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	DM_AUT_OVWT_ENABLE Domain write/read permission overwrite enable 0: Disable

Offset: 0x00D0			Register Name: IOMMU_DM_AUT_OVWT_REG
Bit	Read/Write	Default/Hex	Description
			1: Enable
30:14	/	/	/
13	R/W	0x0	M6_WT_AUT_OVWT_CTRL Master6 write permission overwrite control 0: The write-operation is permitted 1: The write-operation is prohibited <b>Note: The bit is not used.</b>
12	R/W	0x0	M6_RD_AUT_OVWT_CTRL Master6 read permission overwrite control 0: The read-operation is permitted 1: The read-operation is prohibited <b>Note: The bit is not used.</b>
11	R/W	0x0	M5_WT_AUT_OVWT_CTRL Master5 write permission overwrite control 0: The write-operation is permitted 1: The write-operation is prohibited <b>Note: This bit is not used.</b>
10	R/W	0x0	M5_RD_AUT_OVWT_CTRL Master5 read permission overwrite control 0: The read-operation is permitted 1: The read-operation is prohibited <b>Note: This bit is not used.</b>
9	R/W	0x0	M4_WT_AUT_OVWT_CTRL Master5 write permission overwrite control 0: The write-operation is permitted 1: The write-operation is prohibited
8	R/W	0x0	M4_RD_AUT_OVWT_CTRL Master5 read permission overwrite control 0: The read-operation is permitted 1: The read-operation is prohibited
7	R/W	0x0	M3_WT_AUT_OVWT_CTRL Master3 write permission overwrite control 0: The write-operation is permitted 1: The write-operation is prohibited
6	R/W	0x0	M3_RD_AUT_OVWT_CTRL Master3 read permission overwrite control

Offset: 0x00D0			Register Name: IOMMU_DM_AUT_OVWT_REG
Bit	Read/Write	Default/Hex	Description
			0: The read-operation is permitted 1: The read-operation is prohibited
5	R/W	0x0	M2_WT_AUT_OVWT_CTRL Master2 write permission overwrite control 0: The write-operation is permitted 1: The write-operation is prohibited
4	R/W	0x0	M2_RD_AUT_OVWT_CTRL Master2 read permission overwrite control 0: The read-operation is permitted 1: The read-operation is prohibited
3	R/W	0x0	M1_WT_AUT_OVWT_CTRL Master1 write permission overwrite control 0: The write-operation is permitted 1: The write-operation is prohibited <b>Note: This bit is not used.</b>
2	R/W	0x0	M1_RD_AUT_OVWT_CTRL Master1 read permission overwrite control 0: The read-operation is permitted 1: The read-operation is prohibited <b>Note: This bit is not used.</b>
1	R/W	0x0	M0_WT_AUT_OVWT_CTRL Master0 write permission overwrite control 0: The write-operation is permitted 1: The write-operation is prohibited
0	R/W	0x0	M0_RD_AUT_OVWT_CTRL Master0 read permission overwrite control 0: The read-operation is permitted 1: The read-operation is prohibited

### 3.11.6.32 0x0100 IOMMU Interrupt Enable Register (Default Value: 0x0000\_0000)

Invalid page table and permission error can not make one device or multi-devices in system work normally.

Permission error usually happens in MicroTLB. The error generates interrupt and waits for processing through software.

Invalid page table usually happens in MacroTLB. The error can not influence the access of other devices. So the error page table needs go back the way it comes, but the error should not be written in each level TLB.

Offset: 0x0100			Register Name: IOMMU_INT_ENABLE_REG
Bit	Read/Write	Default/Hex	Description
31:21	/	/	/
20	R/W	0x0	<p>DBG_PF_L2_IV_PT_EN. Debug or Prefetch Invalid Page Table Enable 0: Mask interrupt 1: Enable interrupt</p>
19	R/W	0x0	<p>DBG_PF_PC_IV_L1_PT_EN. Debug or Prefetch PTW Cache Invalid Level1 Page Table Enable 0: Mask interrupt 1: Enable interrupt</p>
18	R/W	0x0	<p>DBG_PF_DRAM_IV_L1_PT_EN. Debug or Prefetch DRAM Invalid Level1 Page Table Enable 0: Mask interrupt 1: Enable interrupt</p>
17	R/W	0x0	<p>L2_PAGE_TABLE_INVALID_EN Level2 page table invalid interrupt enable 0: Mask interrupt 1: Enable interrupt</p>
16	R/W	0x0	<p>L1_PAGE_TABLE_INVALID_EN Level1 page table invalid interrupt enable 0: Mask interrupt 1: Enable interrupt</p>
15:7	/	/	/
6	R/W	0x0	<p>MICRO_TLB6_INVALID_EN Micro TLB6 permission invalid interrupt enable 0: Mask interrupt 1: Enable interrupt <b>Note: This bit is not used.</b></p>
5	R/W	0x0	<p>MICRO_TLB5_INVALID_EN Micro TLB5 permission invalid interrupt enable 0: Mask interrupt 1: Enable interrupt <b>Note: This bit is not used.</b></p>
4	R/W	0x0	<p>MICRO_TLB4_INVALID_EN Micro TLB4 permission invalid interrupt enable</p>

Offset: 0x0100			Register Name: IOMMU_INT_ENABLE_REG
Bit	Read/Write	Default/Hex	Description
			0: Mask interrupt 1: Enable interrupt
3	R/W	0x0	MICRO_TLB3_INVALID_EN Micro TLB3 permission invalid interrupt enable 0: Mask interrupt 1: Enable interrupt
2	R/W	0x0	MICRO_TLB2_INVALID_EN Micro TLB2 permission invalid interrupt enable 0: Mask interrupt 1: Enable interrupt
1	R/W	0x0	MICRO_TLB1_INVALID_EN Micro TLB1 permission invalid interrupt enable 0: Mask interrupt 1: Enable interrupt <b>Note: This bit is not used.</b>
0	R/W	0x0	MICRO_TLB0_INVALID_EN Micro TLB0 permission invalid interrupt enable 0: Mask interrupt 1: Enable interrupt

### 3.11.6.33 0x0104 IOMMU Interrupt Clear Register (Default Value: 0x0000\_0000)

Offset: 0x0104			Register Name: IOMMU_INT_CLR_REG
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
17	W	0x0	L2_PAGE_TABLE_INVALID_CLR Level2 page table invalid interrupt clear bit 0: Invalid operation 1: Clear interrupt
16	W	0x0	L1_PAGE_TABLE_INVALID_CLR Level1 page table invalid interrupt clear bit 0: Invalid operation 1: Clear interrupt
15:7	/	/	/
6	W	0x0	MICRO_TLB6_INVALID_CLR Micro TLB6 permission invalid interrupt clear bit

Offset: 0x0104			Register Name: IOMMU_INT_CLR_REG
Bit	Read/Write	Default/Hex	Description
			0: Invalid operation 1: Clear interrupt <b>Note: The bit is not used.</b>
5	W	0x0	<b>MICRO_TLB5_INVALID_CLR</b> Micro TLB5 permission invalid interrupt clear bit 0: Invalid operation 1: Clear interrupt <b>Note: This bit is not used.</b>
4	W	0x0	<b>MICRO_TLB4_INVALID_CLR</b> Micro TLB4 permission invalid interrupt clear bit 0: Invalid operation 1: Clear interrupt
3	W	0x0	<b>MICRO_TLB3_INVALID_CLR</b> Micro TLB3 permission invalid interrupt clear bit 0: Invalid operation 1: Clear interrupt
2	W	0x0	<b>MICRO_TLB2_INVALID_CLR</b> Micro TLB2 permission invalid interrupt clear bit 0: Invalid operation 1: Clear interrupt
1	W	0x0	<b>MICRO_TLB1_INVALID_CLR</b> Micro TLB1 permission invalid interrupt clear bit 0: Invalid operation 1: Clear interrupt <b>Note: This bit is not used.</b>
0	W	0x0	<b>MICRO_TLB0_INVALID_CLR</b> Micro TLB0 permission invalid interrupt clear bit 0: Invalid operation 1: Clear interrupt

### 3.11.6.34 0x0108 IOMMU Interrupt Status Register (Default Value: 0x0000\_0000)

Offset: 0x0108			Register Name: IOMMU_INT_STA_REG
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
17	R	0x0	L2_PAGE_TABLE_INVALID_STA

Offset: 0x0108			Register Name: IOMMU_INT_STA_REG
Bit	Read/Write	Default/Hex	Description
			<p>Level2 page table invalid interrupt status bit 0: Interrupt does not happen or interrupt is cleared 1: Interrupt happens</p>
16	R	0x0	<p>L1_PAGE_TABLE_INVALID_STA Level1 page table invalid interrupt status bit 0: Interrupt does not happen or interrupt is cleared 1: Interrupt happens</p>
15:7	/	/	/
6	R	0x0	<p>MICRO_TLB6_INVALID_STA Micro TLB6 permission invalid interrupt status bit 0: Interrupt does not happen or interrupt is cleared 1: Interrupt happens <b>Note: The bit is not used.</b></p>
5	R	0x0	<p>MICRO_TLB5_INVALID_STA Micro TLB5 permission invalid interrupt status bit 0: Interrupt does not happen or interrupt is cleared 1: Interrupt happens <b>Note: This bit is not used.</b></p>
4	R	0x0	<p>MICRO_TLB4_INVALID_STA Micro TLB4 permission invalid interrupt status bit 0: Interrupt does not happen or interrupt is cleared 1: Interrupt happens</p>
3	R	0x0	<p>MICRO_TLB3_INVALID_STA Micro TLB3 permission invalid interrupt status bit 0: Interrupt does not happen or interrupt is cleared 1: Interrupt happens</p>
2	R	0x0	<p>MICRO_TLB2_INVALID_STA Micro TLB2 permission invalid interrupt status bit 0: Interrupt does not happen or interrupt is cleared 1: Interrupt happens</p>
1	R	0x0	<p>MICRO_TLB1_INVALID_STA Micro TLB1 permission invalid interrupt status bit 0: Interrupt does not happen or interrupt is cleared 1: Interrupt happens <b>Note: This bit is not used.</b></p>
0	R	0x0	MICRO_TLB0_INVALID_STA

Offset: 0x0108			Register Name: IOMMU_INT_STA_REG
Bit	Read/Write	Default/Hex	Description
			Micro TLBO permission invalid interrupt status bit 0: Interrupt does not happen or interrupt is cleared 1: Interrupt happens

### 3.11.6.35 0x0110 IOMMU Interrupt Error Address 0 Register (Default Value: 0x0000\_0000)

Offset: 0x0110			Register Name: IOMMU_INT_ERR_ADDR0_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	INT_ERR_ADDR0 Virtual address that caused Micro TLB0 to interrupt

### 3.11.6.36 0x0114 IOMMU Interrupt Error Address 1 Register (Default Value: 0x0000\_0000)

Offset: 0x0114			Register Name: IOMMU_INT_ERR_ADDR1_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	INT_ERR_ADDR1 Virtual address that caused Micro TLB1 to interrupt <b>Note:</b> This field is not used.

### 3.11.6.37 0x0118 IOMMU Interrupt Error Address 2 Register (Default Value: 0x0000\_0000)

Offset: 0x0118			Register Name: IOMMU_INT_ERR_ADDR2_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	INT_ERR_ADDR2 Virtual address that caused Micro TLB2 to interrupt

### 3.11.6.38 0x011C IOMMU Interrupt Error Address 3 Register (Default Value: 0x0000\_0000)

Offset: 0x011C			Register Name: IOMMU_INT_ERR_ADDR3_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	INT_ERR_ADDR3 Virtual address that caused Micro TLB3 to interrupt

### 3.11.6.39 0x0120 IOMMU Interrupt Error Address 4 Register (Default Value: 0x0000\_0000)

Offset: 0x0120			Register Name: IOMMU_INT_ERR_ADDR4_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	INT_ERR_ADDR4 Virtual address that caused Micro TLB4 to interrupt

### 3.11.6.40 0x0124 IOMMU Interrupt Error Address 5 Register (Default Value: 0x0000\_0000)

Offset: 0x0124			Register Name: IOMMU_INT_ERR_ADDR5_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	INT_ERR_ADDR5 Virtual address that caused Micro TLB5 to interrupt <b>Note: This field is not used.</b>

### 3.11.6.41 0x0128 IOMMU Interrupt Error Address 6 Register (Default Value: 0x0000\_0000)

Offset: 0x0128			Register Name: IOMMU_INT_ERR_ADDR6_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	INT_ERR_ADDR6 Virtual address that caused Micro TLB6 to interrupt <b>Note: This field is not used.</b>

### 3.11.6.42 0x0130 IOMMU Interrupt Error Address 7 Register (Default Value: 0x0000\_0000)

Offset: 0x0130			Register Name: IOMMU_INT_ERR_ADDR7_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	INT_ERR_ADDR7 Virtual address that caused L1 page table to interrupt

### 3.11.6.43 0x0134 IOMMU Interrupt Error Address 8 Register (Default Value: 0x0000\_0000)

Offset: 0x0134			Register Name: IOMMU_INT_ERR_ADDR8_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	INT_ERR_ADDR8 Virtual address that caused L2 page table to interrupt

### 3.11.6.44 0x0150 IOMMU Interrupt Error Data 0 Register (Default Value: 0x0000\_0000)

Offset: 0x0150			Register Name: IOMMU_INT_ERR_DATA0_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	INT_ERR_DATA0 Corresponding page table of virtual address that caused Micro TLB0 to interrupt

### 3.11.6.45 0x0154 IOMMU Interrupt Error Data 1 Register (Default Value: 0x0000\_0000)

Offset: 0x0154			Register Name: IOMMU_INT_ERR_DATA1_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	INT_ERR_DATA1 Corresponding page table of virtual address that caused Micro TLB1 to interrupt <b>Note:</b> This field is not used.

### 3.11.6.46 0x0158 IOMMU Interrupt Error Data 2 Register (Default Value: 0x0000\_0000)

Offset: 0x0158			Register Name: IOMMU_INT_ERR_DATA2_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	INT_ERR_DATA2 Corresponding page table of virtual address that caused Micro TLB2 to interrupt

### 3.11.6.47 0x015C IOMMU Interrupt Error Data 3 Register (Default Value: 0x0000\_0000)

Offset: 0x015C			Register Name: IOMMU_INT_ERR_DATA3_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	INT_ERR_DATA3 Corresponding page table of virtual address that caused Micro TLB3 to interrupt

### 3.11.6.48 0x0160 IOMMU Interrupt Error Data 4 Register (Default Value: 0x0000\_0000)

Offset: 0x0160			Register Name: IOMMU_INT_ERR_DATA4_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	<p>INT_ERR_DATA4</p> <p>Corresponding page table of virtual address that caused Micro TLB4 to interrupt</p>

### 3.11.6.49 0x0164 IOMMU Interrupt Error Data 5 Register (Default Value: 0x0000\_0000)

Offset: 0x0164			Register Name: IOMMU_INT_ERR_DATA5_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	<p>INT_ERR_DATA5</p> <p>Corresponding page table of virtual address that caused Micro TLB5 to interrupt</p> <p><b>Note: This field is not used.</b></p>

### 3.11.6.50 0x0168 IOMMU Interrupt Error Data 6 Register (Default Value: 0x0000\_0000)

Offset: 0x0168			Register Name: IOMMU_INT_ERR_DATA6_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	<p>INT_ERR_DATA6</p> <p>Corresponding page table of virtual address that caused Micro TLB6 to interrupt</p> <p><b>Note: This field is not used.</b></p>

### 3.11.6.51 0x0170 IOMMU Interrupt Error Data 7 Register (Default Value: 0x0000\_0000)

Offset: 0x0170			Register Name: IOMMU_INT_ERR_DATA7_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	<p>INT_ERR_DATA7</p> <p>Corresponding page table of virtual address that caused L1 page table to interrupt</p>

### 3.11.6.52 0x0174 IOMMU Interrupt Error Data 8 Register (Default Value: 0x0000\_0000)

Offset: 0x0174			Register Name: IOMMU_INT_ERR_DATA8_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	INT_ERR_DATA8 Corresponding page table of virtual address that caused L2 page table to interrupt

### 3.11.6.53 0x0180 IOMMU L1 Page Table Interrupt Register (Default Value: 0x0000\_0000)

Offset: 0x0180			Register Name: IOMMU_L1PG_INT_REG
Bit	Read/Write	Default/Hex	Description
31	R	0x0	DBG_MODE_L1PG_INT Debug mode address switch causes L1 page table to occur interrupt.
30:7	/	/	/
6	R	0x0	MASTER6_L1PG_INT Master6 address switch causes L1 page table to occur interrupt. <b>Note: The bit is not used.</b>
5	R	0x0	MASTER5_L1PG_INT Master5 address switch causes L1 page table to occur interrupt. <b>Note: This bit is not used.</b>
4	R	0x0	MASTER4_L1PG_INT Master4 address switch causes L1 page table to occur interrupt.
3	R	0x0	MASTER3_L1PG_INT Master3 address switch causes L1 page table to occur interrupt.
2	R	0x0	MASTER2_L1PG_INT Master2 address switch causes L1 page table to occur interrupt.
1	R	0x0	MASTER1_L1PG_INT Master1 address switch causes L1 page table to occur interrupt. <b>Note: This bit is not used.</b>
0	R	0x0	MASTER0_L1PG_INT Master0 address switch causes L1 page table to occur interrupt.

**3.11.6.54 0x0184 IOMMU L2 Page Table Interrupt Register (Default Value: 0x0000\_0000)**

Offset: 0x0184			Register Name: IOMMU_L2PG_INT_REG
Bit	Read/Write	Default/Hex	Description
31	R	0x0	DBG_MODE_L2PG_INT Debug mode address switch causes L2 page table to occur interrupt.
30:7	/	/	/
6	R	0x0	MASTER6_L2PG_INT Master6 address switch causes L2 page table to occur interrupt. <b>Note: The bit is not used.</b>
5	R	0x0	MASTER5_L2PG_INT Master5 address switch causes L2 page table to occur interrupt. <b>Note: This bit is not used.</b>
4	R	0x0	MASTER4_L2PG_INT Master4 address switch causes L2 page table to occur interrupt.
3	R	0x0	MASTER3_L2PG_INT Master3 address switch causes L2 page table to occur interrupt.
2	R	0x0	MASTER2_L2PG_INT Master2 address switch causes L2 page table to occur interrupt.
1	R	0x0	MASTER1_L2PG_INT Master1 address switch causes L2 page table to occur interrupt. <b>Note: This bit is not used.</b>
0	R	0x0	MASTER0_L2PG_INT Master0 address switch causes L2 page table to occur interrupt.

**3.11.6.55 0x0190 IOMMU Virtual Address Register (Default Value: 0x0000\_0000)**

Offset: 0x0190			Register Name: IOMMU_VA_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	VA Virtual address of read/write

### 3.11.6.56 0x0194 IOMMU Virtual Address Data Register (Default Value: 0x0000\_0000)

Offset: 0x0194			Register Name: IOMMU_VA_DATA_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	VA_DATA Data corresponding to read/write virtual address

### 3.11.6.57 0x0198 IOMMU Virtual Address Configuration Register (Default Value: 0x0000\_0000)

Offset: 0x0198			Register Name: IOMMU_VA_CONFIG_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	MODE_SEL 0: Prefetch 1: Debug Mode It is used to chose Prefetch mode or Debug mode.
31:9	/	/	/
8	R/W	0x0	VA_CONFIG Virtual Address Configuration 0: Read operation 1: Write operation
7:1	/	/	/
0	R/WAC	0x0	VA_CONFIG_START 0: No operation or operation is completed 1: Start After the operation is completed, the bit can clear automatically.

Read operation process:

- Write IOMMU\_VA\_REG[31:0];
- Write IOMMU\_VA\_CONFIG\_REG[8] to 0;
- Write IOMMU\_VA\_CONFIG\_REG[0] to 1 to start read-process;
- Query IOMMU\_VA\_CONFIG\_REG[0] until it is 0;
- Read IOMMU\_VA\_DATA\_REG[31:0];

Write operation process:

- Write IOMMU\_VA\_REG[31:0];
- Write IOMMU\_VA\_DATA\_REG[31:0];
- Write IOMMU\_VA\_CONFIG\_REG[8] to 1;
- Write IOMMU\_VA\_CONFIG\_REG[0] to 1 to start write-process;
- Query IOMMU\_VA\_CONFIG\_REG[0] until it is 0;

### 3.11.6.58 0x0200 IOMMU PMU Enable Register (Default Value: 0x0000\_0000)

Offset: 0x0200			Register Name: IOMMU_PMU_ENABLE_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	PMU_ENABLE 0: Disable statistical function 1: Enable statistical function

### 3.11.6.59 0x0210 IOMMU PMU Clear Register (Default Value: 0x0000\_0000)

Offset: 0x0210			Register Name: IOMMU_PMU_CLR_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/WAC	0x0	PMU_CLR 0: No clear operation or clear operation is completed 1: Clear counter data After the operation is completed, the bit can clear automatically.

### 3.11.6.60 0x0230 IOMMU PMU Access Low 0 Register (Default Value: 0x0000\_0000)

Offset: 0x0230			Register Name: IOMMU_PMU_ACCESS_LOW0_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_ACCESS_LOW0 Record total number of Micro TLB0 access, lower 32-bit register

### 3.11.6.61 0x0234 IOMMU PMU Access High 0 Register (Default Value: 0x0000\_0000)

Offset: 0x0234			Register Name: IOMMU_PMU_ACCESS_HIGH0_REG
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R	0x0	PMU_ACCESS_HIGH0 Record total number of Micro TLB0 access, higher 11-bit register

**3.11.6.62 0x0238 IOMMU PMU Hit Low 0 Register (Default Value: 0x0000\_0000)**

Offset: 0x0238			Register Name: IOMMU_PMU_HIT_LOW0_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_HIT_LOW0 Record total number of Micro TLB0 hit, lower 32-bit register

**3.11.6.63 0x023C IOMMU PMU Hit High 0 Register (Default Value: 0x0000\_0000)**

Offset: 0x023C			Register Name: IOMMU_PMU_HIT_HIGH0_REG
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R	0x0	PMU_HIT_HIGH0 Record total number of Micro TLB0 hit, higher 11-bit register

**3.11.6.64 0x0240 IOMMU PMU Access Low 1 Register (Default Value: 0x0000\_0000)**

Offset: 0x0240			Register Name: IOMMU_PMU_ACCESS_LOW1_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_ACCESS_LOW1 Record total number of Micro TLB1 access, lower 32-bit register <b>Note: This bit is not used.</b>

**3.11.6.65 0x0244 IOMMU PMU Access High 1 Register (Default Value: 0x0000\_0000)**

Offset: 0x0244			Register Name: IOMMU_PMU_ACCESS_HIGH1_REG
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R	0x0	PMU_ACCESS_HIGH1 Record total number of Micro TLB1 access, higher 11-bit register <b>Note: This field is not used.</b>

### 3.11.6.66 0x0248 IOMMU PMU Hit Low 1 Register (Default Value: 0x0000\_0000)

Offset: 0x0248			Register Name: IOMMU_PMU_HIT_LOW1_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_HIT_LOW1 Record total number of Micro TLB1 hit, lower 32-bit register <b>Note: This field is not used.</b>

### 3.11.6.67 0x024C IOMMU PMU Hit High 1 Register (Default Value: 0x0000\_0000)

Offset: 0x024C			Register Name: IOMMU_PMU_HIT_HIGH1_REG
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R	0x0	PMU_HIT_HIGH1 Record total number of Micro TLB1 hit, higher 11-bit register <b>Note: This field is not used.</b>

### 3.11.6.68 0x0250 IOMMU PMU Access Low 2 Register (Default Value: 0x0000\_0000)

Offset: 0x0250			Register Name: IOMMU_PMU_ACCESS_LOW2_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_ACCESS_LOW2 Record total number of Micro TLB2 access, lower 32-bit register

### 3.11.6.69 0x0254 IOMMU PMU Access High 2 Register (Default Value: 0x0000\_0000)

Offset: 0x0254			Register Name: IOMMU_PMU_ACCESS_HIGH2_REG
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R	0x0	PMU_ACCESS_HIGH2 Record total number of Micro TLB2 access, higher 11-bit register

**3.11.6.70 0x0258 IOMMU PMU Hit Low 2 Register (Default Value: 0x0000\_0000)**

Offset: 0x0258			Register Name: IOMMU_PMU_HIT_LOW2_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_HIT_LOW2 Record total number of Micro TLB2 hit, lower 32-bit register

**3.11.6.71 0x025C IOMMU PMU Hit High 2 Register (Default Value: 0x0000\_0000)**

Offset: 0x025C			Register Name: IOMMU_PMU_HIT_HIGH2_REG
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R	0x0	PMU_HIT_HIGH2 Record total number of Micro TLB2 hit, higher 11-bit register

**3.11.6.72 0x0260 IOMMU PMU Access Low 3 Register (Default Value: 0x0000\_0000)**

Offset: 0x0260			Register Name: IOMMU_PMU_ACCESS_LOW3_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_ACCESS_LOW3 Record total number of Micro TLB3 access, lower 32-bit register

**3.11.6.73 0x0264 IOMMU PMU Access High 3 Register (Default Value: 0x0000\_0000)**

Offset: 0x0264			Register Name: IOMMU_PMU_ACCESS_HIGH3_REG
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R	0x0	PMU_ACCESS_HIGH3 Record total number of Micro TLB3 access, higher 11-bit register

**3.11.6.74 0x0268 IOMMU PMU Hit Low 3 Register (Default Value: 0x0000\_0000)**

Offset: 0x0268			Register Name: IOMMU_PMU_HIT_LOW3_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_HIT_LOW3 Record total number of Micro TLB3 hit, lower 32-bit register

**3.11.6.75 0x026C IOMMU PMU Hit High 3 Register (Default Value: 0x0000\_0000)**

Offset: 0x026C			Register Name: IOMMU_PMU_HIT_HIGH3_REG
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R	0x0	PMU_HIT_HIGH3 Record total number of Micro TLB3 hit, higher 11-bit register

**3.11.6.76 0x0270 IOMMU PMU Access Low 4 Register (Default Value: 0x0000\_0000)**

Offset: 0x0270			Register Name: IOMMU_PMU_ACCESS_LOW4_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_ACCESS_LOW4 Record total number of Micro TLB4 access, lower 32-bit register

**3.11.6.77 0x0274 IOMMU PMU Access High 4 Register (Default Value: 0x0000\_0000)**

Offset: 0x0274			Register Name: IOMMU_PMU_ACCESS_HIGH4_REG
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R	0x0	PMU_ACCESS_HIGH4 Record total number of Micro TLB4 access, higher 11-bit register

**3.11.6.78 0x0278 IOMMU PMU Hit Low 4 Register (Default Value: 0x0000\_0000)**

Offset: 0x0278			Register Name: IOMMU_PMU_HIT_LOW4_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_HIT_LOW4 Record total number of Micro TLB4 hit, lower 32-bit register

### 3.11.6.79 0x027C IOMMU PMU Hit High 4 Register (Default Value: 0x0000\_0000)

Offset: 0x027C			Register Name: IOMMU_PMU_HIT_HIGH4_REG
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R	0x0	PMU_HIT_HIGH4 Record total number of Micro TLB4 hit, higher 11-bit register

### 3.11.6.80 0x0280 IOMMU PMU Access Low 5 Register (Default Value: 0x0000\_0000)

Offset: 0x0280			Register Name: IOMMU_PMU_ACCESS_LOW5_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_ACCESS_LOW5 Record total number of Micro TLB5 access, lower 32-bit register <b>Note: This field is not used.</b>

### 3.11.6.81 0x0284 IOMMU PMU Access High 5 Register (Default Value: 0x0000\_0000)

Offset: 0x0284			Register Name: IOMMU_PMU_ACCESS_HIGH5_REG
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R	0x0	PMU_ACCESS_HIGH5 Record total number of Micro TLB5 access, higher 11-bit register <b>Note: This field is not used.</b>

### 3.11.6.82 0x0288 IOMMU PMU Hit Low 5 Register (Default Value: 0x0000\_0000)

Offset: 0x0288			Register Name: IOMMU_PMU_HIT_LOW5_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_HIT_LOW5 Record total number of Micro TLB5 hit, lower 32-bit register <b>Note: This field is not used.</b>

### 3.11.6.83 0x028C IOMMU PMU Hit High 5 Register (Default Value: 0x0000\_0000)

Offset: 0x028C			Register Name: IOMMU_PMU_HIT_HIGH5_REG
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R	0x0	PMU_HIT_HIGH5 Record total number of Micro TLB5 hit, higher 11-bit register <b>Note: This field is not used.</b>

### 3.11.6.84 0x0290 IOMMU PMU Access Low 6 Register (Default Value: 0x0000\_0000)

Offset: 0x0290			Register Name: IOMMU_PMU_ACCESS_LOW6_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_ACCESS_LOW6 Record total number of Micro TLB6 access, lower 32-bit register <b>Note: The field is not used.</b>

### 3.11.6.85 0x0294 IOMMU PMU Access High 6 Register (Default Value: 0x0000\_0000)

Offset: 0x0294			Register Name: IOMMU_PMU_ACCESS_HIGH6_REG
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R	0x0	PMU_ACCESS_HIGH6 Record total number of Micro TLB6 access, higher 11-bit register <b>Note: The field is not used.</b>

### 3.11.6.86 0x0298 IOMMU PMU Hit Low 6 Register (Default Value: 0x0000\_0000)

Offset: 0x0298			Register Name: IOMMU_PMU_HIT_LOW6_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_HIT_LOW6 Record total number of Micro TLB6 hit, lower 32-bit register <b>Note: The field is not used.</b>

**3.11.6.87 0x029C IOMMU PMU Hit High 6 Register (Default Value: 0x0000\_0000)**

Offset: 0x029C			Register Name: IOMMU_PMU_HIT_HIGH6_REG
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R	0x0	PMU_HIT_HIGH6 Record total number of Micro TLB6 hit, higher 11-bit register <b>Note:</b> The field is not used.

**3.11.6.88 0x02D0 IOMMU PMU Access Low 7 Register (Default Value: 0x0000\_0000)**

Offset: 0x02D0			Register Name: IOMMU_PMU_ACCESS_LOW7_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_ACCESS_LOW7 Record total number of Micro TLB7 access, lower 32-bit register

**3.11.6.89 0x02D4 IOMMU PMU Access High 7 Register (Default Value: 0x0000\_0000)**

Offset: 0x02D4			Register Name: IOMMU_PMU_ACCESS_HIGH7_REG
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R	0x0	PMU_ACCESS_HIGH7 Record total number of Micro TLB7 access, higher 11-bit register

**3.11.6.90 0x02D8 IOMMU PMU Hit Low 7 Register (Default Value: 0x0000\_0000)**

Offset: 0x02D8			Register Name: IOMMU_PMU_HIT_LOW7_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_HIT_LOW7 Record total number of Micro TLB7 hit, lower 32-bit register

**3.11.6.91 0x02DC IOMMU PMU Hit High 7 Register (Default Value: 0x0000\_0000)**

Offset: 0x02DC			Register Name: IOMMU_PMU_HIT_HIGH7_REG
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/

Offset: 0x02DC			Register Name: IOMMU_PMU_HIT_HIGH7_REG
Bit	Read/Write	Default/Hex	Description
10:0	R	0x0	PMU_HIT_HIGH7 Record total number of Micro TLB7 hit, higher 11-bit register

### 3.11.6.92 0x02E0 IOMMU PMU Access Low 8 Register (Default Value: 0x0000\_0000)

Offset: 0x02E0			Register Name: IOMMU_PMU_ACCESS_LOW8_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_ACCESS_LOW8 Record total number of PTW Cache access, lower 32-bit register

### 3.11.6.93 0x02E4 IOMMU PMU Access High 8 Register (Default Value: 0x0000\_0000)

Offset: 0x02E4			Register Name: IOMMU_PMU_ACCESS_HIGH8_REG
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R	0x0	PMU_ACCESS_HIGH8 Record total number of PTW Cache access, higher 11-bit register

### 3.11.6.94 0x02E8 IOMMU PMU Hit Low 8 Register (Default Value: 0x0000\_0000)

Offset: 0x02E8			Register Name: IOMMU_PMU_HIT_LOW8_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_HIT_LOW8 Record total number of PTW Cache hit, lower 32-bit register

### 3.11.6.95 0x02EC IOMMU PMU Hit High 8 Register (Default Value: 0x0000\_0000)

Offset: 0x02EC			Register Name: IOMMU_PMU_HIT_HIGH8_REG
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R	0x0	PMU_HIT_HIGH8 Record total number of PTW Cache hit, higher 11-bit register

**3.11.6.96 0x0300 IOMMU Total Latency Low 0 Register (Default Value: 0x0000\_0000)**

Offset: 0x0300			Register Name: IOMMU_PMU_TL_LOW0_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_TL_LOW0 Record total latency of Master0, lower 32-bit register

**3.11.6.97 0x0304 IOMMU Total Latency High 0 Register (Default Value: 0x0000\_0000)**

Offset: 0x0304			Register Name: IOMMU_PMU_TL_HIGH0_REG
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
17:0	R	0x0	PMU_TL_HIGH0 Record total latency of Master0, higher 18-bit register

**3.11.6.98 0x0308 IOMMU Max Latency 0 Register (Default Value: 0x0000\_0000)**

Offset: 0x0308			Register Name: IOMMU_PMU_ML0_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_ML0 Record the max latency of Master0.

**3.11.6.99 0x0310 IOMMU Total Latency Low 1 Register (Default Value: 0x0000\_0000)**

Offset: 0x0310			Register Name: IOMMU_PMU_TL_LOW1_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_TL_LOW1 Record total latency of Master1, lower 32-bit register

**3.11.6.100 0x0314 IOMMU Total Latency High 1 Register (Default Value: 0x0000\_0000)**

Offset: 0x0314			Register Name: IOMMU_PMU_TL_HIGH1_REG
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
17:0	R	0x0	PMU_TL_HIGH1 Record total latency of Master1, higher 18-bit register

**3.11.6.101 0x0318 IOMMU Max Latency 1 Register (Default Value: 0x0000\_0000)**

Offset: 0x0318			Register Name: IOMMU_PMU_ML1_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_ML1 Record the max latency of Master1.

**3.11.6.102 0x0320 IOMMU Total Latency Low 2 Register (Default Value: 0x0000\_0000)**

Offset: 0x0320			Register Name: IOMMU_PMU_TL_LOW2_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_TL_LOW2 Record total latency of Master2, lower 32-bit register

**3.11.6.103 0x0324 IOMMU Total Latency High 2 Register (Default Value: 0x0000\_0000)**

Offset: 0x0324			Register Name: IOMMU_PMU_TL_HIGH2_REG
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
17:0	R	0x0	PMU_TL_HIGH2 Record total latency of Master2, higher 18-bit register

**3.11.6.104 0x0328 IOMMU Max Latency 2 Register (Default Value: 0x0000\_0000)**

Offset: 0x0328			Register Name: IOMMU_PMU_ML2_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_ML2 Record the max latency of Master2.

**3.11.6.105 0x0330 IOMMU Total Latency Low 3 Register (Default Value: 0x0000\_0000)**

Offset: 0x0330			Register Name: IOMMU_PMU_TL_LOW3_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_TL_LOW3

Offset: 0x0330			Register Name: IOMMU_PMU_TL_LOW3_REG
Bit	Read/Write	Default/Hex	Description
			Record total latency of Master3, lower 32-bit register

### 3.11.6.106 0x0334 IOMMU Total Latency High 3 Register (Default Value: 0x0000\_0000)

Offset: 0x0334			Register Name: IOMMU_PMU_TL_HIGH3_REG
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
17:0	R	0x0	PMU_TL_HIGH3 Record total latency of Master3, higher 18-bit register

### 3.11.6.107 0x0338 IOMMU Max Latency 3 Register (Default Value: 0x0000\_0000)

Offset: 0x0338			Register Name: IOMMU_PMU_ML3_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_ML3 Record the max latency of Master3.

### 3.11.6.108 0x0340 IOMMU Total Latency Low 4 Register (Default Value: 0x0000\_0000)

Offset: 0x0340			Register Name: IOMMU_PMU_TL_LOW4_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_TL_LOW4 Record total latency of Master4, lower 32-bit register

### 3.11.6.109 0x0344 IOMMU Total Latency High 4 Register (Default Value: 0x0000\_0000)

Offset: 0x0344			Register Name: IOMMU_PMU_TL_HIGH4_REG
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
17:0	R	0x0	PMU_TL_HIGH4 Record total latency of Master4, higher 18-bit register

### 3.11.6.110 0x0348 IOMMU Max Latency 4 Register (Default Value: 0x0000\_0000)

Offset: 0x0348			Register Name: IOMMU_PMU_ML4_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_ML4 Record the max latency of Master4.

### 3.11.6.111 0x0350 IOMMU Total Latency Low 5 Register (Default Value: 0x0000\_0000)

Offset: 0x0350			Register Name: IOMMU_PMU_TL_LOW5_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_TL_LOW5 Record total latency of Master5, lower 32-bit register

### 3.11.6.112 0x0354 IOMMU Total Latency High 5 Register (Default Value: 0x0000\_0000)

Offset: 0x0354			Register Name: IOMMU_PMU_TL_HIGH5_REG
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
17:0	R	0x0	PMU_TL_HIGH5 Record total latency of Master5, higher 18-bit register

### 3.11.6.113 0x0358 IOMMU Max Latency 5 Register (Default Value: 0x0000\_0000)

Offset: 0x0358			Register Name: IOMMU_PMU_ML5_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_ML5 Record the max latency of Master5.

### 3.11.6.114 0x0360 IOMMU Total Latency Low 6 Register (Default Value: 0x0000\_0000)

Offset: 0x0360			Register Name: IOMMU_PMU_TL_LOW6_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_TL_LOW6 Record total latency of Master6, lower 32-bit register <b>Note: The field is not used.</b>

**3.11.6.115 0x0364 IOMMU Total Latency High 6 Register (Default Value: 0x0000\_0000)**

Offset: 0x0364			Register Name: IOMMU_PMU_TL_HIGH6_REG
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
17:0	R	0x0	PMU_TL_HIGH6 Record total latency of Master6, higher 18-bit register <b>Note: The field is not used.</b>

**3.11.6.116 0x0368 IOMMU Max Latency 6 Register (Default Value: 0x0000\_0000)**

Offset: 0x0368			Register Name: IOMMU_PMU_ML6_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	PMU_ML6 Record the max latency of Master6. <b>Note: The field is not used.</b>

## 3.12 RTC

### 3.12.1 Overview

The Real Time Clock (RTC) is used to implement time counter and timing wakeup functions. The RTC can display the year, month, day, week, hour, minute, second in real time. The RTC has the independent power to continue to work in system power-off.

The RTC has the following features:

- Provides a 16-bit counter for counting day, 5-bit counter for counting hour, 6-bit counter for counting minute, 6-bit counter for counting second
- Connects a 32.768 kHz low-frequency oscillator for count clock
- Timer frequency is 1 kHz
- Configurable initial value by software anytime
- Supports timing alarm, and generates interrupt and wakeup the external devices
- Eight 32-bit user registers for storing power-off information

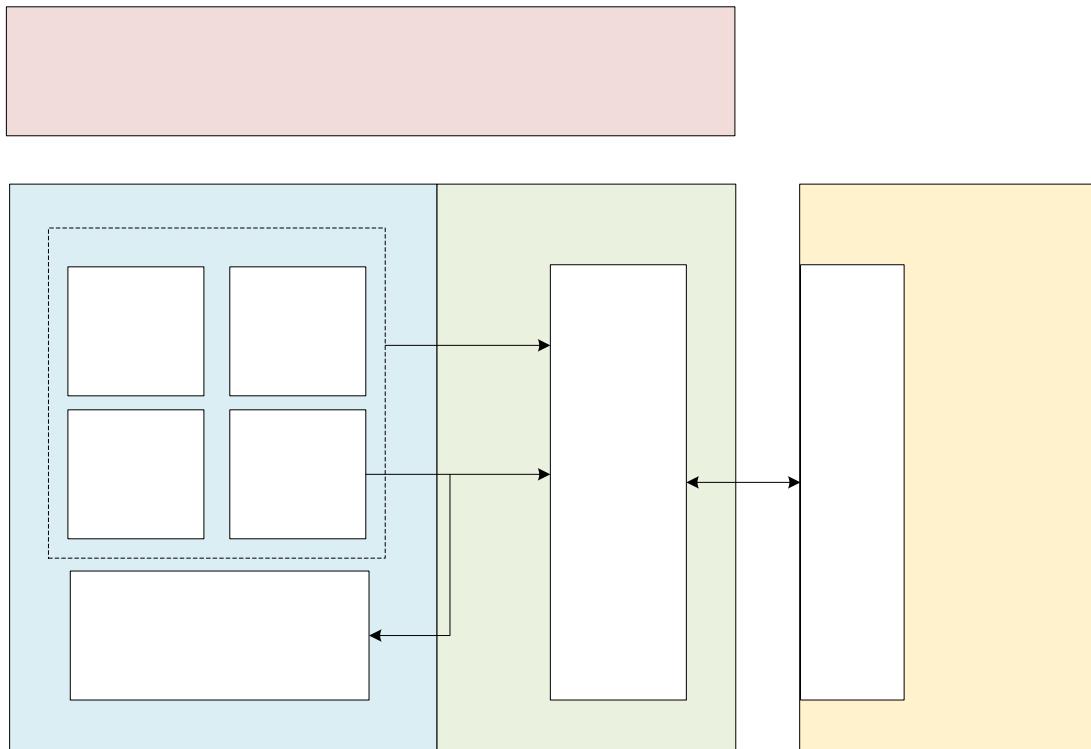


The register configuration of RTC is AHB bus, it only can support word operation, not byte operation and half-word operation.

---

### 3.12.2 Block Diagram

The following figure shows the block diagram of the RTC.

**Figure 3- 36 RTC Block Diagram**

### 3.12.3 Functional Description

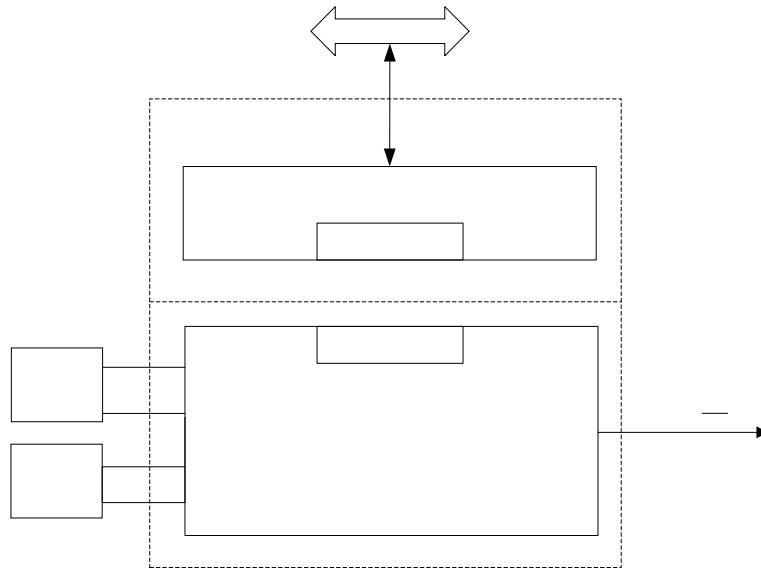
#### 3.12.3.1 External Signals

**Table 3-19 RTC External Signals**

Signal	Description	Type
X32KIN	32.768 kHz oscillator input	AI
X32KOUT	32.768 kHz oscillator output	AO
VCC-RTC	RTC high voltage, generated via external power	P

### 3.12.3.2 Typical Application

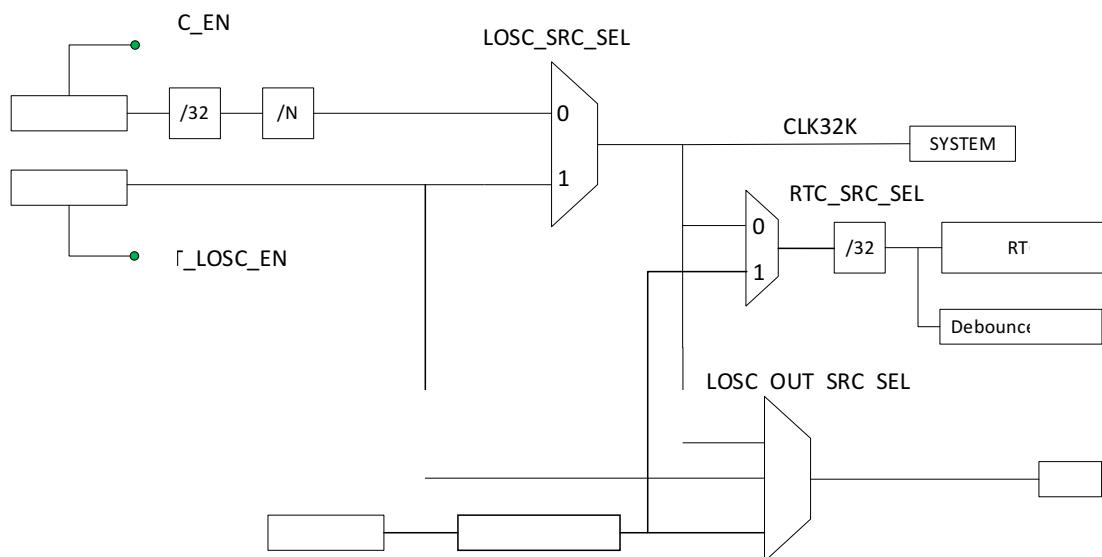
Figure 3-37 RTC Application Diagram



### 3.12.3.3 Clock Tree

The following figure shows the clock tree of the RTC.

Figure 3-38 RTC Clock Tree



- **LOSC**

The LOSC has 2 clock sources: internal RC, external low frequency crystal. The LOSC selects the internal RC by default, when the system starts, the LOSC can select by software the external low frequency crystal to provide much accuracy clock. The clock accurate of the LOSC is related to the accurate of the external low frequency crystal. Usually select 32.768 kHz crystal with  $\pm 20$  ppm frequency tolerance. When using internal RC, the clock can be changed by changing division ratio. When using external clock, the clock cannot be changed.

- **RTC**

The clock sources of RTC can be selected by related switches, including 32K divided by internal 16 MHz RC, 32K divided by external DCXO, and external 32.768 kHz crystal.

- **System 32K**

The clock sources of system 32K are from external 32.768 kHz crystal and 32K divided by the internal 16 MHz RC.

- **RTC\_32K\_FANOUT**

The clock source of RTC\_32K\_FANOUT can select CLK32K, external 32.768 kHz crystal or 32K divided by external DCXO.

### 3.12.3.4 Real Time Clock

**Figure 3-39 RTC Counter**

The 1K counter adds 1 on each rising edge of the clock. When the clock number reaches 0x3FF, 1K counter starts to count again from 0, and the second counter adds 1. The step structure of 1 kHz counter is as follows.

**Figure 3-40 RTC 1 kHz Counter Step Structure**

According to above implementation, the changing range of each counter is as follows.

**Table 3-20 RTC Counter Changing Range**

Counter	Range
Second	0 to 59
Minute	0 to 59
Hour	0 to 23
Day	0 to 65535 (The year, month, day need be transformed by software according to day counter)



Because there is no error correction mechanism in the hardware, each counter configuration should not exceed a reasonable counting range.

### 3.12.3.5 Alarm 0

The principle of alarm0 is a comparator. When RTC timer reaches the scheduled time, the RTC generates the interrupt, or outputs low level signal by NMI pin to wakeup power management chip.

The RTC only generates one interrupt when RTC timer reached the scheduled day, hour, minute and second counter, then the RTC needs to be set a new scheduled time, the next interrupt can be generated.

### 3.12.3.6 Power-off Information Storage

The RTC provides eight 32-bit general purpose register to store power-off information.

Because VCC-RTC always holds non-power-off state after VCC-RTC cold starts, when the system is in shutdown or standby scene, the CPU can judge software process by the storing information.

### 3.12.3.7 RTC\_VIO

The RTC module has a LDO, the input source of the LDO is VCC\_RTC, the output of the LDO is RTC\_VIO, the value of RTC\_VIO is adjustable, the RTC\_VIO is mainly used for internal digital logic.

•

## 3.12.4 Programming Guidelines

### 3.12.4.1 RTC Clock Control

- Step 1** Select clock source: Select clock source by the bit0 of [LOSC\\_CTRL\\_REG](#), the clock source is the internal RC oscillator by default. When the system starts, the clock source can be switched to the external 32K oscillator by software.
- Step 2** Auto switch: After enabled the bit[15:14] of [LOSC\\_CTRL\\_REG](#), the RTC automatically switches clock source to the internal oscillator when the external crystal could not output waveform, the switch status can query by the bit[1] of [LOSC\\_AUTO\\_SWT\\_STA\\_REG](#).



#### NOTE

If only configuring the bit[15] of [LOSC\\_CTRL\\_REG](#), the clock source status bit cannot be changed after the auto switch is valid, because the two functions are independent.

---

Here is the basic code samples.

```
Write (0x16aa4000,LOSC_Ctrl); //Write key field  
Write (0x16aa4001,LOSC_Ctrl); //Select the external 32K clock
```

### 3.12.4.2 RTC Calendar

- Step 1** Write time initial value: Write the current time to [RTC\\_DAY\\_REG](#) and [RTC\\_HH\\_MM\\_SS\\_REG](#).
- Step 2** After updated time, the RTC restarts to count again. The software can read the current time anytime.
-

**NOTE**

- The RTC can only provide day counter, so the current day counter need be converted to year, month, day and week by software.
- Ensure the bit[8:7] of LOSC\_CTRL\_REG is 0 before the next time configuration is performed.

Here is the basic code samples.

For example: set time to 21st, 07:08:09 and read it.

```
RTC_DAY_REG = 0x000000015;
```

```
RTC_HH_MM_SS_REG = 0x00070809; //0000 0000 000|0 0000(Hour) 00|00 0000(Minute) 00|00 0000(Second)
```

```
Read (RTC_DAY_REG);
```

```
Read (RTC_HH_MM_SS_REG);
```

### 3.12.4.3 Alarm0

**Step 1** Enable alram0 interrupt by writing [ALARM0 IRQ EN](#).

**Step 2** Set the counter comparator, write the count-down day, hour, minute, second number to [ALARM0 DAY SET REG](#) and [ALARM0 HH-MM-SS SET REG](#).

**Step 3** Enable alarm0 function by writing [ALARM0 ENABLE REG](#), then the software can query alarm count value in real time by [ALARM0 DAY SET REG](#) and [ALARM0 HH-MM-SS SET REG](#). When the setting time reaches, [ALARM0 IRQ STA REG](#) is set to 1 to generate interrupt.

**Step 4** After enter the interrupt process, write [ALARM0 IRQ STA REG](#) to clear the interrupt pending, and execute the interrupt process.

**Step 5** Resume the interrupt and continue to execute the interrupted process.

**Step 6** The power-off wakeup is generated via SoC hardware and PMIC, the software only needs to set the pending condition of alarm0, and set [ALARM CONFIG REG](#) to 1.

Here is the basic code samples.

```
irq_request(GIC_SRC_R_Alarm0, Alm0_handler);
irq_enable(GIC_SRC_R_Alarm0);
writel(1, ALARM0_DAY_SET_REG);
writel(1, RTC_HH_MM_SS_REG); //set 1 second corresponding to normal mode;
```

```
writel(1, ALMO_EN);
writel(1, ALM_CONFIG); //NMI output
while(!readl(ALMO_IRQ_STA));
writel(1, ALMO_IRQ_EN);
while(readl(ALMO_IRQ_STA));
```

### 3.12.5 Register List

Module Name	Base Address
RTC	0x07090000

Register Name	Offset	Description
LOSC_CTRL_REG	0x0000	Low Oscillator Control Register
LOSC_AUTO_SWT_STA_REG	0x0004	LOSC Auto Switch Status Register
INTOSC_CLK_PRESCAL_REG	0x0008	Internal OSC Clock Pre-scalar Register
RTC_DAY_REG	0x0010	RTC Year-Month-Day Register
RTC_HH_MM_SS_REG	0x0014	RTC Hour-Minute-Second Register
ALARMO_DAY_SET_REG	0x0020	Alarm 0 Day Setting Register
ALARMO_CUR_VLU_REG	0x0024	Alarm 0 Counter Current Value Register
ALARMO_ENABLE_REG	0x0028	Alarm 0 Enable Register
ALARMO_IRQ_EN	0x002C	Alarm 0 IRQ Enable Register
ALARMO_IRQ_STA_REG	0x0030	Alarm 0 IRQ Status Register
ALARM_CONFIG_REG	0x0050	Alarm Configuration Register
32K_FOUT_CTRL_GATING_REG	0x0060	32K Fanout Control Gating Register
GP_DATA_REG	0x0100 + N*0x04	General Purpose Register (N=0 to 7)
FBOOT_INFO_REG0	0x0120	Fast Boot Information Register0
FBOOT_INFO_REG1	0x0124	Fast Boot Information Register1
DCXO_CTRL_REG	0x0160	DCXO Control Register
RTC_VIO_REG	0x0190	RTC_VIO Regulation Register
IC_CHARA_REG	0x01F0	IC Characteristic Register
VDD_OFF_GATING_CTRL_REG	0x01F4	VDD Off Gating Control Register
EFUSE_HV_PWR SWT_CTRL_REG	0x0204	Efuse High Voltage Power Switch Control Register
RTC_SPI_CLK_CTRL_REG	0x0310	RTC SPI Clock Control Register

**NOTE**

The offset addresses less than 0x0300 are in VDD\_RTC power domain, and the offset addresses larger than or equal to 0x300 are in VDD\_SYS power domain.

### 3.12.6 Register Description

#### 3.12.6.1 0x0000 LOSC Control Register (Default Value: 0x0000\_4010)

Offset:0x0000			Register Name: LOSC_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:16	W	0x0	KEY_FIELD Key Field This field should be filled with 0x16AA, and then the bit0 and bit1 can be written with the new value.
15	R/W	0x0	LOSC_AUTO_SWT_FUNCTION LOSC auto switch function disable 0: Enable 1: Disable
14	R/W	0x1	LOSC_AUTO_SWT_32K_SEL_EN LOSC auto switch 32K clk source select enable 0: Disable. When the losc losts, the 32k clk source will not change to RC 1: Enable. When the losc losts, the 32k clk source will change to RC (LOSC_SRC_SEL will be changed from 1 to 0)
13:9	/	/	/
8	R/W	0x0	RTC_HHMMSS_ACCE RTC Hour Minute Second access After writing the RTC HH-MM-SS Register, this bit is set and it will be cleared until the real writing operation is finished. After writing the RTC HH-MM-SS Register, the RTC HH-MM-SS Register will be refreshed for at most one second. <b>Note: Make sure that the bit is 0 for time configuration.</b>
7	R/W	0x0	RTC_DAY_ACCE RTC DAY access After writing the RTC DAY register, this bit is set and it will be cleared until the real writing operation is finished.

Offset:0x0000			Register Name: LOSC_CTRL_REG
Bit	Read/Write	Default/Hex	Description
			<p>After writing the RTC DAY register, the DAY register will be refreshed for at most one second.</p> <p><b>Note:</b> Make sure that the bit is 0 for time configuration.</p>
6:5	/	/	/
4	R/W	0x1	<p>EXT_LOSC_EN External 32.768 kHz Crystal Enable 0: Disable 1: Enable</p>
3:2	R/W	0x0	<p>EXT_LOSC_GSM External 32.768 kHz Crystal GSM 00: Low 01: / 10: / 11: High When GSM is changed, the 32K oscillation circuit will arise transient instability. If the autoswitch function (bit 15) is enabled, 32K changes to RC16M with certain probability. The GSM can influence the time of 32K starting oscillation, the more the GSM, the shorter the time of starting oscillation. So modifying GSM is not recommended. If you need to modify the GSM, firstly disable the auto switch function (bit 15), with a delay of 50 us, then change the GSM, the 32K clock source is changed to external clock.</p>
1	R/W	0x0	<p>RTC_SRC_SEL RTC_TIMER Clock Source Select 0: LOSC_SRC 1: 24MDIV32K Before switching the bit, make sure that the 24MDIV32K function is enabled, that is, the bit16 of the <a href="#">32K Fanout Control Register</a> is 1.</p>
0	R/W	0x0	<p>LOSC_SRC_SEL LOSC Clock Source Select 0: Low frequency clock from 16M RC 1: External 32.768 kHz OSC</p>



If the bit[8:7] of LOSC\_CTRL\_REG is set, the RTC HH-MM-SS, DD and ALARM DD-HH-MM-SS register cannot be written.

### 3.12.6.2 0x0004 LOSC Auto Switch Status Register (Default Value: 0x0000\_0000)

Offset:0x0004			Register Name: LOSC_AUTO_SWT_STA_REG
Bit	Read/Write	Default/Hex	Description
31:3	/	/	/
2	R	0x0	EXT_LOSC_STA Work only when the auto switch function is enabled. 0: External 32.768 kHz OSC work normally 1: External 32.768 kHz OSC work abnormally
1	R/W1C	0x0	LOSC_AUTO_SWT_PEND LOSC auto switch pending 0: No effect 1: Auto switch pending, it means LOSC_SRC_SEL is changed from 1 to 0. Setting 1 to this bit will clear it.
0	R	0x0	LOSC_SRC_SEL_STA Checking LOSC clock source status 0: Low frequency clock from 16M RC 1: External 32.768 kHz OSC

### 3.12.6.3 0x0008 Internal OSC Clock Prescalar Register (Default Value: 0x0000\_000F)

Offset:0x0008			Register Name: INTOSC_CLK_PRESCAL_REG
Bit	Read/Write	Default/Hex	Description
31:5	/	/	/
4:0	R/W	0xF	INTOSC_32K_CLK_PRESCAL Internal OSC 32K Clock Prescalar value N. The clock output = Internal RC/32/N. 00000: 1 00001: 2 00002: 3 ..... 11111: 32

### 3.12.6.4 0x0010 RTC Year-Month-DAY Register (Default Value: UDF)

Offset:0x0010			Register Name: RTC_DAY_REG
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	UDF	DAY Set Day Range from 0 to 65535.

### 3.12.6.5 0x0014 RTC Hour-Minute-Second Register (Default Value: UDF)

Offset:0x0014			Register Name: RTC_HH_MM_SS_REG
Bit	Read/Write	Default/Hex	Description
31:21	/	/	/
20:16	R/W	UDF	HOUR Set hour Range from 0 to 23.
15:14	/	/	/
13:8	R/W	UDF	MINUTE Set minute Range from 0 to 59.
7:6	/	/	/
5:0	R/W	UDF	SECOND Set second Range from 0 to 59.

### 3.12.6.6 0x0020 Alarm 0 Day Setting Register (Default Value: 0x0000\_0000)

Offset:0x0020			Register Name: ALARM0_DAY_SET_REG
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0	ALARM0_COUNTER Alarm 0 Counter is based on Day.

### 3.12.6.7 0x0024 Alarm 0 Counter Current Value Register (Default Value: UDF)

Offset:0x0024			Register Name: ALARM0_CUR_VLU_REG
Bit	Read/Write	Default/Hex	Description
31:21	/	/	/
20:16	R/W	UDF	HOUR Current hour Range from 0 to 23.
15:14	/	/	/
13:8	R/W	UDF	MINUTE Current minute Range from 0 to 59.
7:6	/	/	/
5:0	R/W	UDF	SECOND Current second Range from 0 to 59.

### 3.12.6.8 0x0028 Alarm 0 Enable Register (Default Value: 0x0000\_0000)

Offset:0x0028			Register Name: ALARM0_ENABLE_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	ALM_0_EN Alarm 0 Enable 0: Disable 1: Enable

### 3.12.6.9 0x002C Alarm 0 IRQ Enable Register (Default Value: 0x0000\_0000)

Offset:0x002C			Register Name: ALARM0_IRQ_EN
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	ALARM0_IRQ_EN Alarm 0 IRQ Enable 0: Disable 1: Enable

**3.12.6.10 0x0030 Alarm 0 IRQ Status Register (Default Value: 0x0000\_0000)**

Offset:0x0030			Register Name: ALARM0_IRQ_STA_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W1C	0x0	ALARM0_IRQ_PEND Alarm 0 IRQ Pending bit 0: No effect 1: Pending, alarm 0 counter value is reached If alarm 0 irq enable is set to 1, the pending bit will be sent to the interrupt controller.

**3.12.6.11 0x0050 Alarm Configuration Register (Default Value: 0x0000\_0000)**

Offset:0x0050			Register Name: ALARM_CONFIG_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	ALARM_WAKEUP Configuration of alarm wake up output. 0: Disable alarm wake up output 1: Enable alarm wake up output

**3.12.6.12 0x0060 32K Fanout Control Gating Register (Default Value: 0x0000\_0000)**

Offset:0x0060			Register Name: 32K_FOUT_CTRL_GATING_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	HOSC_TO_32K_DIVIDER_ENABLE HOSC to 32k divider enable 0: Disable the hosc 24M to 32K divider circuit 1: Enable the hosc 24M to 32K divider circuit
15:3	/	/	/
2:1	R/W	0x0	LOSC_OUT_SRC_SEL LOSC output source select 00: RTC_32K (select by RC_CLK_SRC_SEL & LOSC_SRC_SEL) 01: LOSC 10: HOSC divided 32K

Offset:0x0060			Register Name: 32K_FOUT_CTRL_GATING_REG
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	<p>32K_FANOUT_GATING</p> <p>LOSC out gating enable</p> <p>Configuration of LOSC output, and there is no LOSC output by default.</p> <p>0: Mask LOSC output gating</p> <p>1: Enable LOSC output gating</p>

### 3.12.6.13 0x0100+N\*0x0004 General Purpose Register (Default Value: 0x0000\_0000)

Offset:0x0100+N*0x0004 (N=0 to 7)			Register Name: GP_DATA_REGN
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<p>GP_DATA</p> <p>Data [31:0]</p>



General purpose register 0 to 7 value can be stored if the RTC-VIO is larger than 0.7 V.

### 3.12.6.14 0x0120 Fast Boot Information Register0 (Default Value: 0x0000\_0000)

Offset:0x0120			Register Name: FBOOT_INFO_REG0
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<p>FBOOT_INFO0</p> <p>Fast Boot info</p> <p>Fast Boot Information 0, refer to BROM spec.</p>

### 3.12.6.15 0x0124 Fast Boot Information Register1 (Default Value: 0x0000\_0000)

Offset:0x0124			Register Name: FBOOT_INFO_REG1
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<p>FBOOT_INFO1</p> <p>Fast Boot info</p> <p>Fast Boot Information 1, refer to BROM spec.</p>

**3.12.6.16 0x0160 DCXO Control Register (Default Value: 0x883F\_10F7)**

Offset:0x0160			Register Name: DCXO_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x1	CLK_REQ_ENB Clock REQ enable 0: Enable DCXO wake up function 1: Disable DCXO wake up function
30:28	/	/	/
27:24	R/W	0x8	DCXO_ICTRL DCXO current control value
23	/	/	/
22:16	R/W	0x3F	DCXO_TRIM DCXO cap array value The capacity cell is 55 fF.
15:13	/	/	/
12:8	R/W	0x10	DCXO_BG DCXO bandgap output voltage
7	R/W	0x1	DCXO_LDO_INRUSHB DCXO LDO driving capacity signal, active high
6	R/W	0x1	XTAL_MODE Xtal mode enable signal, active high 0: For external clk input mode 1: For normal mode
5:4	R/W	0x3	DCXO_RFCLK_ENHANCE DCXO rfclk enhance Enhance driving capacity of output OUT_RF_REFCLK, 0x0 for 5 pF, 0x1 for 10 pF, 0x2 for 15 pF, 0x3 for 20 pF.
3	/	/	/
2	R/W	0x1	RSTO_DLY_SEL For Debug Use Only. It cannot configure to 0 in normal state.
1	R/W	0x1	DCXO_EN DCXO enable 1: Enable 0: Disable

Offset:0x0160			Register Name: DCXO_CTRL_REG
Bit	Read/Write	Default/Hex	Description
0	R/W	0x1	<p>CLK16M_RC_EN 1: Enable 0: Disable</p> <p>The related register configuration is necessary to ensure the reset debounce circuit has a stable clock source.</p> <p>The first time SoC starts up, by default, the reset debounce circuit of SoC uses 32K divided by RC16M. In power-off, software reads the related bit to ensure whether EXT32K is working normally, if it is normal, first switch the clock source of debounce circuit to EXT32K, then close RC16M.</p> <p>Without EXT32K scenario or external RTC scenario, software confirms firstly whether EXT32K is working normally before switching, or software does not close RC16M.</p>

### 3.12.6.17 0x0190 RTC\_VIO Regulation Register (Default Value: 0x0000\_0004)

Offset:0x0190			Register Name:RTC_VIO_REG
Bit	Read/Write	Default/Hex	Description
31:5	/	/	/
4	R/W	0x0	<p>V_SEL VDD Select 0: Resistance divider 1: Band gap</p>
3	/	/	/
2:0	R/W	0x4	<p>RTC_VIO_REGU RTC_VIO Voltage Select The RTC-VIO is provided power for RTC digital part. These bits are useful for regulating the RTC_VIO from 0.65 V to 1.3 V. 000: 1.0 V 001: 0.65 V (the configuration can cause RTC reset) 010: 0.7 V 011: 0.8 V 100: 0.9 V 101: 1.1 V 110: 1.2 V 111: 1.3 V</p>

**3.12.6.18 0x01F0 IC Characteristic Register (Default Value: 0x0000\_0000)**

Offset:0x01F0			Register Name: IC_CHAR_A_REG
Bit	Read/Write	Default/Hex	Description
31:16	R/W	0x0	<p>KEY_FIELD Key Field</p> <p>The field should be written as 0x16AA. Writing any other value in this field aborts the write-operation.</p>
15:0	R/W	0x0	<p>ID_DATA</p> <p>Return 0x16AA only if the KEY_FIELD is set as 0x16AA when read those bits, otherwise return 0x0.</p>

**3.12.6.19 0x01F4 VDD Off Gating Control Register (Default Value: 0x0000\_0021)**

Offset:0x01F4			Register Name: VDD_OFF_GATING_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:16	W	0x0	<p>KEY_FIELD Key Field</p> <p>This field should be filled with 0x16AA, and then the bit 15 can be configured.</p>
15	WAC	0x0	<p>PWROFF_GAT_RTC_CFG (For Debug Use Only) Power off gating control signal</p> <p>When using vdd_sys to RTC isolation software control, write this bit to 1. It will only be cleared by resetb release.</p>
14:12	/	/	/
11:4	R/W	0x2	<p>VCCIO_DET_SPARE</p> <p>Bit[7:5]: Reserved, default=0</p> <p>Bit[4]: Bypass debounce circuit, default=0</p> <p>Bit[3]: Enable control, default=0</p> <p>0: Disable VCC-IO detection</p> <p>1: Force the detection output</p> <p>Bit[2:0]: Gear adjustment</p> <p>000: Detection threshold is 2.5 V</p> <p>001: Detection threshold is 2.6 V</p> <p>010: Detection threshold is 2.7 V (default)</p> <p>011: Detection threshold is 2.8 V</p>

Offset:0x01F4			Register Name: VDD_OFF_GATING_CTRL_REG
Bit	Read/Write	Default/Hex	Description
			100: Detection threshold is 2.9 V 101: Detection threshold is 3 V 110: N/A 111: N/A
3:1	/	/	/
0	R/W	0x1	VCCIO_DET_BYPASS_EN 0: not bypass 1: bypass

### 3.12.6.20 0x0204 Efuse High Voltage Power Switch Control Register (Default Value: 0x0000\_0000)

Offset:0x0204			Register Name: EFUSE_HV_PWR SWT_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	EFUSE_1.8V_POWER_SWITCH_CONTROL 1: Open power switch 0: Close power switch

### 3.12.6.21 0x0310 RTC SPI Clock Control Register (Default Value: 0x0000\_0009)

Offset:0x0310			Register Name: RTC_SPI_CLK_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	RTC Reg CFG SPI Clock Gating 0: Gating 1: Not Gating  Before configuring RTC register, the clock divider of SPI needs to be configured firstly, then clock gating needs to be enabled.  <b>Note: Frequency division and clock gating can not be set at the same time.</b>
30:5	/	/	/
4:0	R/W	0x9	RTC Reg CFG SPI Clock Divider: M Actual SPI Clock = AHBS1/(M+1), (0 to 15) The default frequency of AHBS1 is 200 MHz, and the default frequency of SPI Clock is 20 MHz.

Offset:0x0310			Register Name: RTC_SPI_CLK_CTRL_REG
Bit	Read/Write	Default/Hex	Description
			<b>Note:</b> The SPI clock can not exceed 50 MHz, or else the RTC register may be abnormal.

## Contents

4	Video and Graphics .....	360
4.1	DE .....	360
4.2	DI .....	361
4.3	G2D.....	362
4.4	Video Decoding .....	363
4.4.1	Overview .....	363
4.4.2	Block Diagram .....	364
4.5	Video Encoding .....	365
4.5.1	Overview .....	365
4.5.2	Block Diagram .....	365

## Figures

Figure 4-1 DE Block Diagram.....	360
Figure 4-2 Video Decoding Block Diagram.....	364
Figure 4-3 JPGE Block Diagram .....	366

## 4 Video and Graphics

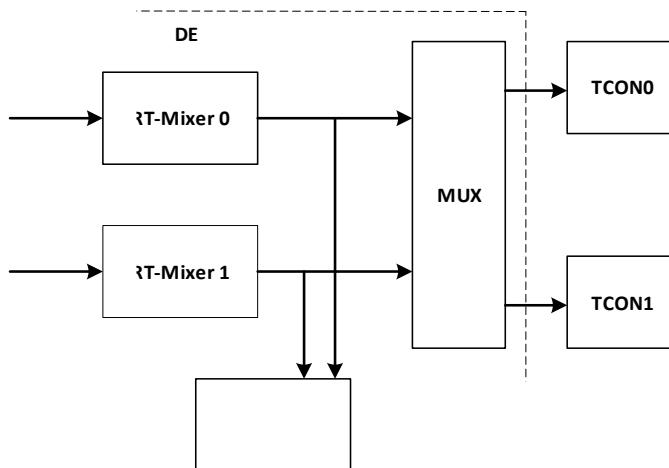
### 4.1 DE

The Display Engine (DE) is a hardware composer to transfer image layers from a local bus or a video buffer to the LCD interface. The DE supports four overlay windows to blend, and supports image post-processing in the video channel. The block diagram of DE is shown in Figure 4-1.

The DE has the following features:

- Output size up to 2048 x 2048
- Four alpha blending channels for main display, three channels for aux display
- Four overlay layers in each channel, and has an independent scaler
- Potter-duff compatible blending operation
- Supports input format Semi-planar YUV422/YUV420/YUV411 and Planar YUV422/YUV420/YUV411, ARGB8888/XRGB8888/RGB888/ARGB4444/ARGB1555/RGB565/palette
- Supports SmartColor2.0 for excellent display experience
  - Adaptive detail/edge enhancement
  - Adaptive color enhancement
  - Adaptive contrast enhancement and fresh tone rectify
- Supports write back for aux display

Figure 4-1 DE Block Diagram



## 4.2 DI

The De-interlacer (DI) converts the interlaced input video frame to progressive video frame.

The DI has the following features:

- Supports YUV420 (Planar/NV12/NV21) and YUV422 (Planar/NV16/NV61) data format
- Supports video resolution from 32x32 to 2048x1280 pixel
- Supports Inter-field interpolation/motion adaptive de-interlace method
- Performance: module clock 600M for 1080p@60Hz YUV420

## 4.3 G2D

The Graphic 2D (G2D) engine is hardware accelerator for 2D graphic.

The G2D has the following features:

- Supports layer size up to 2048 x 2048 pixels
- Supports pre-multiply alpha image data
- Supports color key
- Supports two pipes Porter-Duff alpha blending
- Supports multiple video formats 4:2:0, 4:2:2, 4:1:1 and multiple pixel formats (8/16/24/32 bits graphics layer)
- Supports memory scan order option
- Supports any format convert function
- Supports 1/16x to 32x resize ratio
- Supports 32-phase 8-tap horizontal anti-alias filter and 32-phase 4-tap vertical anti-alias filter
- Supports window clip
- Supports FillRectangle, BitBlit, StretchBlit and MaskBlit
- Supports horizontal and vertical flip, clockwise 0/90/180/270 degree rotate for normal buffer
- Supports horizontal flip, clockwise 0/90/270 degree rotate for LBC buffer

## 4.4 Video Decoding

### 4.4.1 Overview

The Video Decoding consists of Video Control Firmware (VCF) running on ARM processor and embedded hardware Video Engine (VE). VCF gets the bitstream from topper software, parses bitstream, invokes the Video Engine, and generates the decoding image sequence. The decoder image sequence is transmitted by the video output controller to the display device under the control of the topper software.

The Video Decoding has the following features:

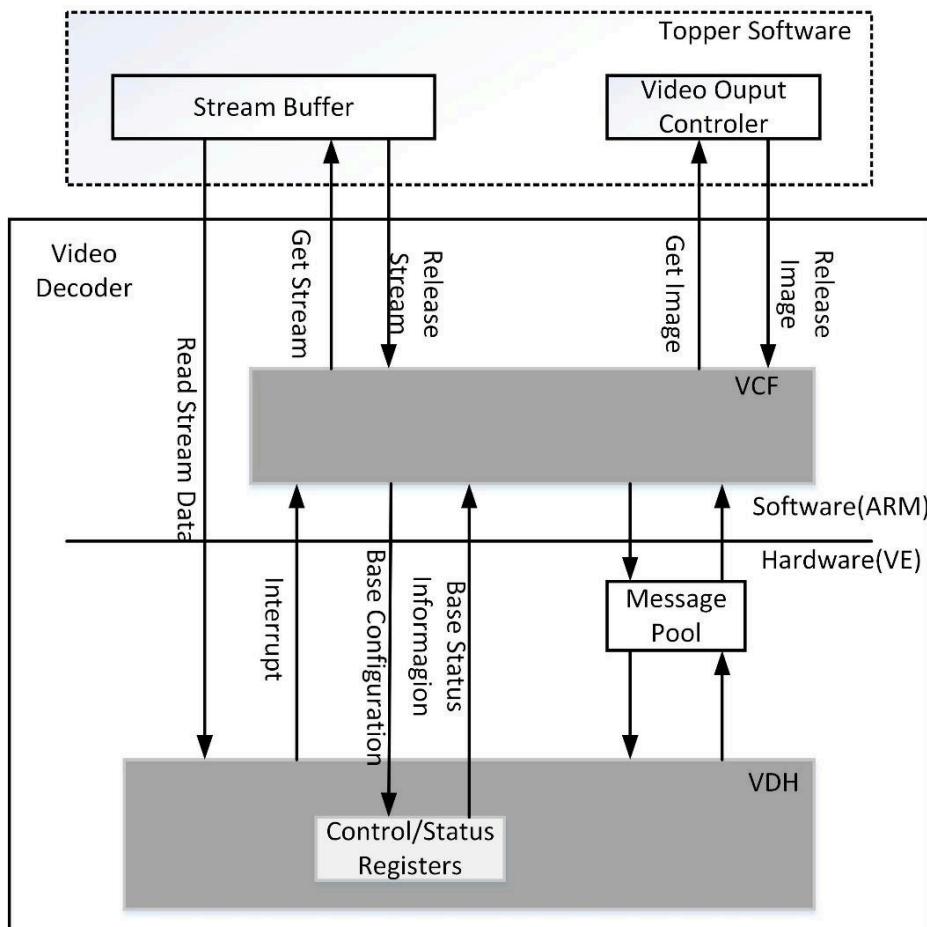
- Supports H.265 MP@L4.1
  - Maximum video resolution: 4096 x 2048
  - Maximum decoding rate: 100 Mbit/s, 4K@30fps
- Supports H.264 BP/MP/HP@L4.2
  - Maximum video resolution: 4096 x 2048
  - Maximum decoding rate: 60 Mbit/s, 4K@24fps
- Supports H.263 BP
  - Maximum video resolution: 1920 x 1080
  - Maximum decoding rate: 60 Mbit/s, 1080p@60fps
- Supports MPEG-4 SP/ASP L5
  - Maximum video resolution: 1920 x 1080
  - Maximum decoding rate: 60 Mbit/s, 1080p@60fps
- Supports MPEG-2 MP/HL
  - Maximum video resolution: 1920 x 1080
  - Maximum decoding rate: 60 Mbit/s, 1080p@60fps
- Supports MPEG-1 MP/HL
  - Maximum video resolution: 1920 x 1080
  - Maximum decoding rate: 60 Mbit/s, 1080p@60fps
- Supports JPEG/Xvid/Sorenson Spark
  - Maximum video resolution: 1920 x 1080
  - Maximum decoding rate: 60 Mbit/s, 1080p@60fps
- Supports MJPEG

- Maximum video resolution: 1920 x 1080
- Maximum decoding rate: 60 Mbit/s, 1080p@30fps

#### 4.4.2 Block Diagram

The functional block diagram of the Video Decoding is as follows.

Figure 4-2 Video Decoding Block Diagram



## 4.5 Video Encoding

### 4.5.1 Overview

The Video Encoding supports JPEG/MJPEG encoding (JPGE).

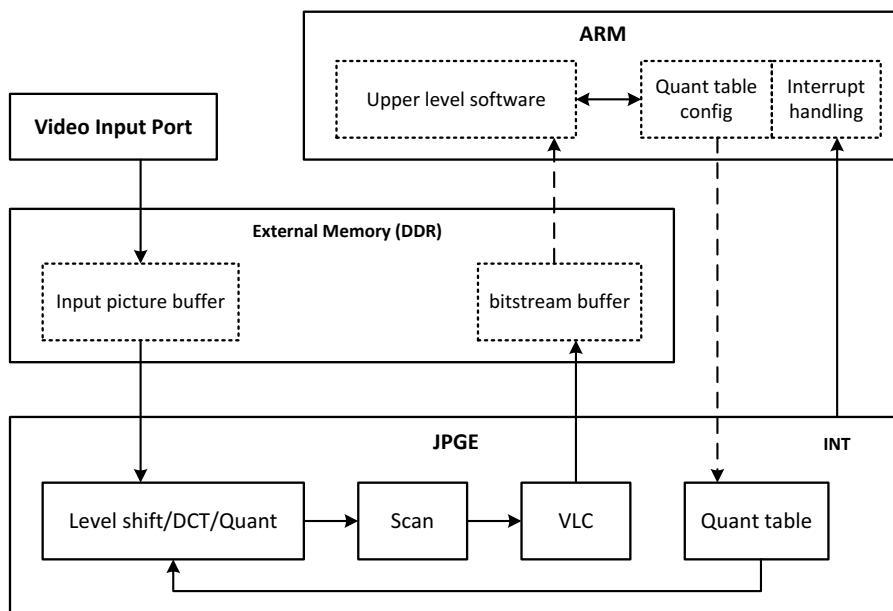
The JPGE is a high-performance JPEG encoder implemented by using hardware. It supports 64-megapixel snapshot or HD MJPEG encoding.

The JPGE has the following features:

- Supports ISO/IEC 10918-1 (CCITT T.81) baseline process (DCT sequential) encoding
- Encodes the pictures in the chrominance sampling format of YCbCr4:2:0 and YCbCr4:2:2
- Supports multiple input picture formats:
  - Semi-planar YCbCr4:2:0
  - Semi-planar YCbCr4:2:2
- Supports JPEG encoding with the performance of 1080p@60fps
- Supports configurable picture resolutions
  - Minimum picture resolution: 192 x 96
  - Maximum picture resolution: 4096 x 4096
- Supports the picture width or height step of 8
- Supports configurable quantization tables for the Y component, Cb component and Cr component respectively
- Supports OSD front-end overlapping
  - OSD overlaying before encoding for a maximum of 16 regions
  - OSD overlaying with any size and at any position (within the size and position range of the picture)
  - 16-level alpha blending
  - OSD overlaying control (enabled or disabled)
- Supports the color-to-gray function
- Supports the MJPEG output bit rate ranging from 2 kbit/s to 60 Mbit/s

### 4.5.2 Block Diagram

The functional block diagram of the JPGE is as follows.

**Figure 4-3 JPGE Block Diagram**

The JPGE realizes various protocol processing with large computation such as OSD, level shift, DCT, quantization, scanning, VLC encoding, and stream generation. The ARM software completes the encoding control processing such as quantization table configuration and interrupt processing.

Before the JPGE starts encoding, the software allocates two types of buffers mainly in the external DDR SDRAM:

- **Input picture buffer**

The JPGE reads the source pictures to be encoded from this buffer during encoding. This buffer is generally written by the Video Input Port module.

- **Stream buffer**

This buffer stores encoded streams. The JPGE writes streams to this buffer during encoding. This buffer is read by software.

# Contents

5	Video Output Interfaces.....	370
5.1	TCON LCD .....	370
5.1.1	Overview .....	370
5.1.2	Block Diagram .....	370
5.1.3	Functional Description .....	370
5.1.4	Programming Guidelines.....	376
5.1.5	Register List.....	380
5.1.6	Register Description .....	381
5.2	TCON TV .....	413
5.2.1	Overview .....	413
5.2.2	Functional Description .....	413
5.2.3	Register List.....	413
5.2.4	Register Description .....	414
5.3	MIPI DSI.....	426
5.3.1	Overview .....	426
5.4	HDMI .....	427
5.4.1	Overview .....	427

## Figures

Figure 5-1 TCON_LCD Block Diagram.....	370
Figure 5-2 LVDS Single Link JEDIA Mode Interface Timing.....	373
Figure 5-3 LVDS Single Link NS Mode Interface Timing .....	373
Figure 5-4 CMAP Module .....	375
Figure 5-5 FRM Module .....	375
Figure 5-6 The Data Timing of MIPI DSI Video Mode.....	376
Figure 5-8 LVDS Mode Configuration Process .....	376

## Tables

Table 5-1 LCD External Signals .....	371
Table 5-2 The Correspondence between LCD and LVDS.....	371
Table 5-3 LVDS Panel Signals .....	372
Table 5-4 TCON_LCD Clock Sources .....	374

## 5 Video Output Interfaces

### 5.1 TCON LCD

#### 5.1.1 Overview

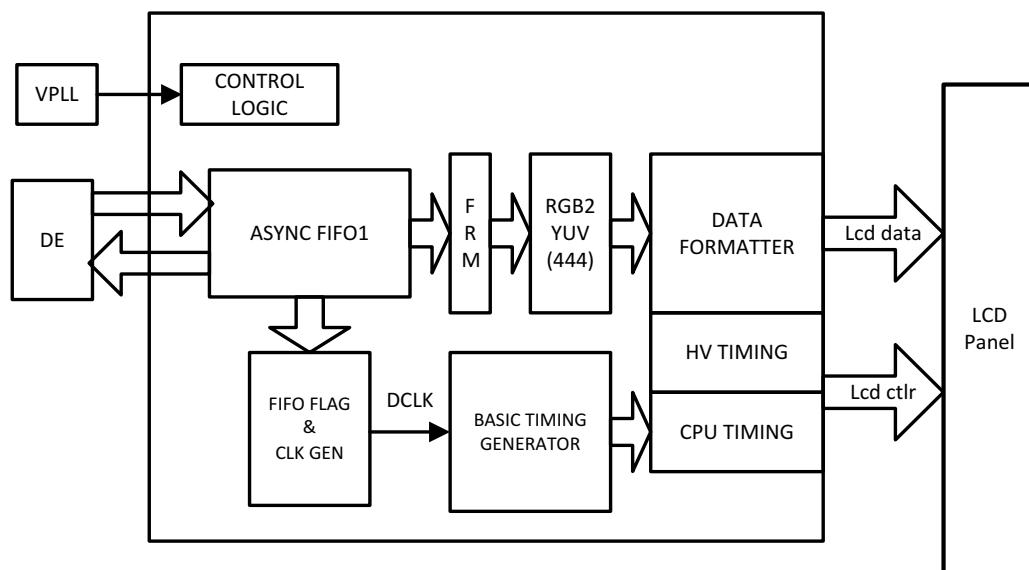
The Timing Controller\_LCD (TCON\_LCD) is a module that processes video signals received from system through a complicated arithmetic and then generates control signals and transmits them to the LCD panel driver IC.

The TCON\_LCD includes the following features:

- Supports LVDS interface with single link, up to 1366x768@60fps
- Supports Gamma correction with R/G/B channel independence

#### 5.1.2 Block Diagram

Figure 5-1 TCON\_LCD Block Diagram



#### 5.1.3 Functional Description

##### 5.1.3.1 External Signals

The LCD external signals are used to connect to panel interface. The panel interface has various types.

**Table 5-1 LCD External Signals**

Signal Name	Description	Type
LCD0-D[13:10], LCD0-D[7:2]	LCD Data Output	O
LCD0-CLK	LCD Clock The pixel data are synchronized by this clock	O
LCD0-VSYNC	LCD Vertical Sync It indicates one new frame	O
LCD0-HSYNC	LCD Horizontal Sync It indicates one new scan line	O
LCD0-DE	LCD Data Output Enable	O
TCON-TRIG	LCD Sync (TCON outputs to LCD for sync)	O
LVDS0-CKP	LVDS0 Positive Port of Clock	O
LVDS0-CKN	LVDS0 Negative Port of Clock	O
LVDS0-V[3:0]P	LVDS0 Positive Port of Data Channel [3:0]	O
LVDS0-V[3:0]N	LVDS0 Negative Port of Data Channel [3:0]	O

The multiplex relationship between LCD and LVDS is shown as follows.

**Table 5-2 The Correspondence between LCD and LVDS**

LCD I/O	LVDS I/O
LCD0-D2	LVDS0-V0P
LCD0-D3	LVDS0-V0N
LCD0-D4	LVDS0-V1P
LCD0-D5	LVDS0-V1N
LCD0-D6	LVDS0-V2P
LCD0-D7	LVDS0-V2N
LCD0-D10	LVDS0-CKP
LCD0-D11	LVDS0-CKN
LCD0-D12	LVDS0-V3P
LCD0-D13	LVDS0-V3N

### 5.1.3.2 Control Signal and Data Port Mapping

External I/O	Internal pin	LVDS
LCD0_VSYNC	IO0	
LCD0_HSYNC	IO1	
LCD0_CLK	IO2	
LCD0_DE	IO3	
LCD0_D23	D23	
LCD0_D22	D22	
LCD0_D21	D21	
LCD0_D20	D20	
LCD0_D19	D19	
LCD0_D18	D18	
LCD0_D17	D17	
LCD0_D16	D16	
LCD0_D15	D15	
LCD0_D14	D14	
LCD0_D13	D13	D3N1
LCD0_D12	D12	D3P1
LCD0_D11	D11	CKN1
LCD0_D10	D10	CKP1
LCD0_D9	D9	
LCD0_D8	D8	
LCD0_D7	D7	D2N1
LCD0_D6	D6	D2P1
LCD0_D5	D5	D1N1
LCD0_D4	D4	D1P1
LCD0_D3	D3	D0N1
LCD0_D2	D2	D0P1
LCD0_D1	D1	
LCD0_D0	D0	

### 5.1.3.3 LVDS Interface

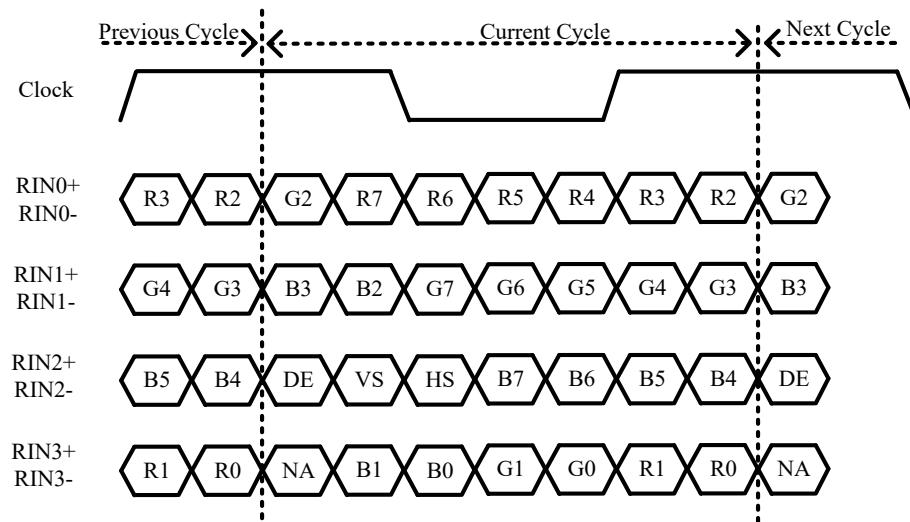
Table 5-3 LVDS Panel Signals

Signal Name	Description	Type
CKP	The positive port of clock	O
CKN	The negative port of clock	O
DOP	The positive port of data channel 0	O
DON	The negative port of data channel 0	O
D1P	The positive port of data channel 1	O
D1N	The negative port of data channel 1	O
D2P	The positive port of data channel 2	O

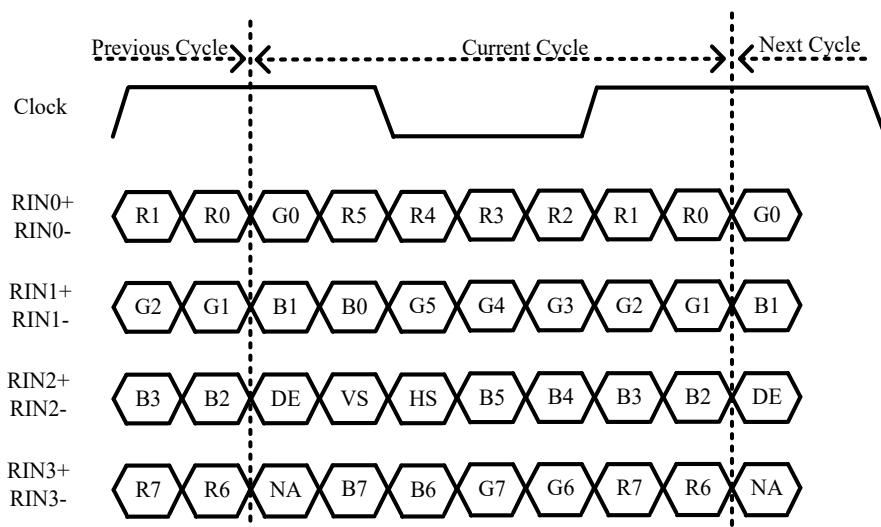
Signal Name	Description	Type
D2N	The negative port of data channel 2	O
D3P	The positive port of data channel 3	O
D3N	The negative port of data channel 3	O

The following figures show the timing of LVDS interface.

**Figure 5-2 LVDS Single Link JEDIA Mode Interface Timing**



**Figure 5-3 LVDS Single Link NS Mode Interface Timing**



### 5.1.3.4 Clock Sources

The following table describes the clock sources of TCON\_LCD. Table 5-9 describes the clock sources of TCON\_LCD.

**Table 5-4 TCON\_LCD Clock Sources**

Clock Sources	Description
PLL_VIDEO0(1X)	By default, PLL_VIDEO0(4X) is 1188 MHz, PLL_VIDEO0(1X) is 297 MHz
PLL_VIDEO0(4X)	
PLL_VIDEO1(1X)	By default, PLL_VIDEO1(4X) is 1188 MHz, PLL_VIDEO1(1X) is 297 MHz
PLL_VIDEO1(4X)	
PLL_PERI(2X)	By default, PLL_PERI(2X) is 1.2 GHz
PLL_AUDIO1(DIV2)	By default, PLL_AUDIO1 is 3072 MHz, PLL_AUDIO1(DIV2) is 1536 MHz

### 5.1.3.5 CEU Module

This module enhances color data from DE .

$$R' = Rr*R + Rg*G + Rb*B + Rc$$

$$G' = Gr*R + Gg*G + Gb*B + Gc$$

$$B' = Br*R + Bg*G + Bb*B + Bc$$

Rr, Rg, Rb, Gr, Gg, Gb, Br, Bg, Bb s13 (-16, 16)

Rc, Gc, Bc s19 (-16384, 16384)

R, G, B u8 [0-255]

R' has the range of [Rmin ,Rmax]

G' has the range of [Rmin ,Rmax]

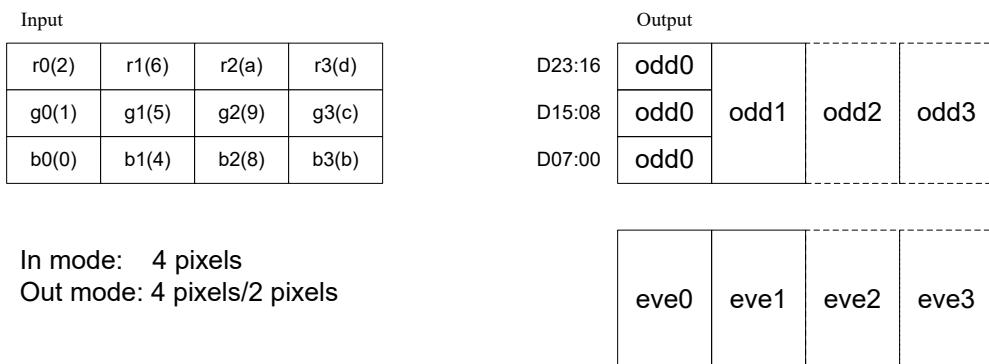
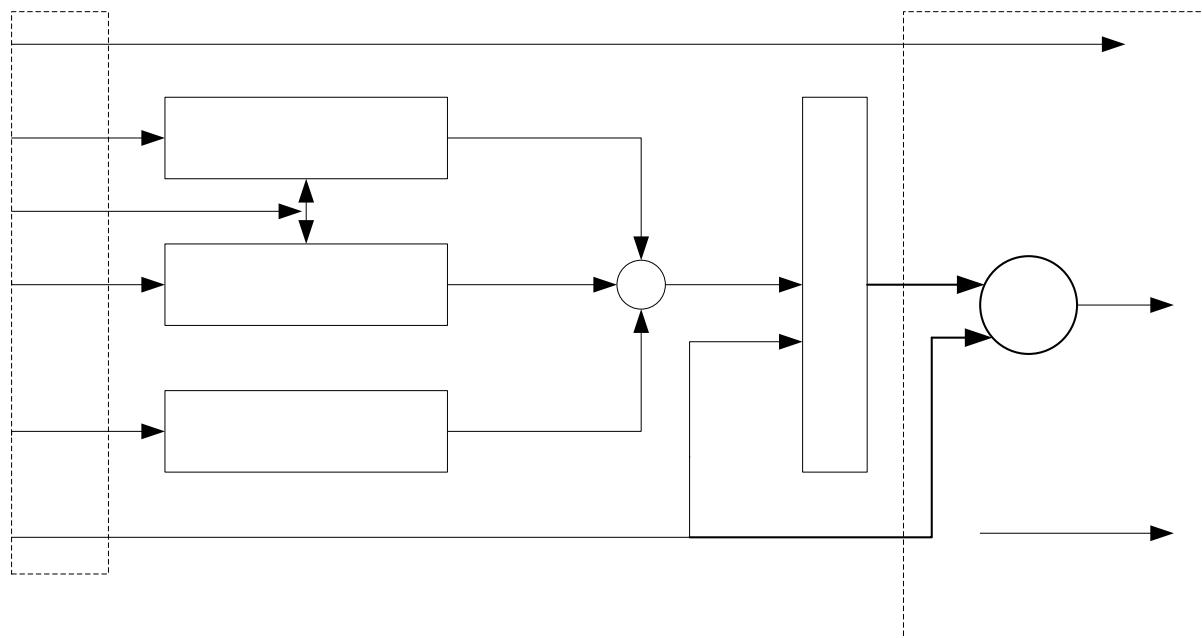
B' has the range of [Rmin ,Rmax]

### 5.1.3.6 CMAP Module

Function: This module maps color data from DE.

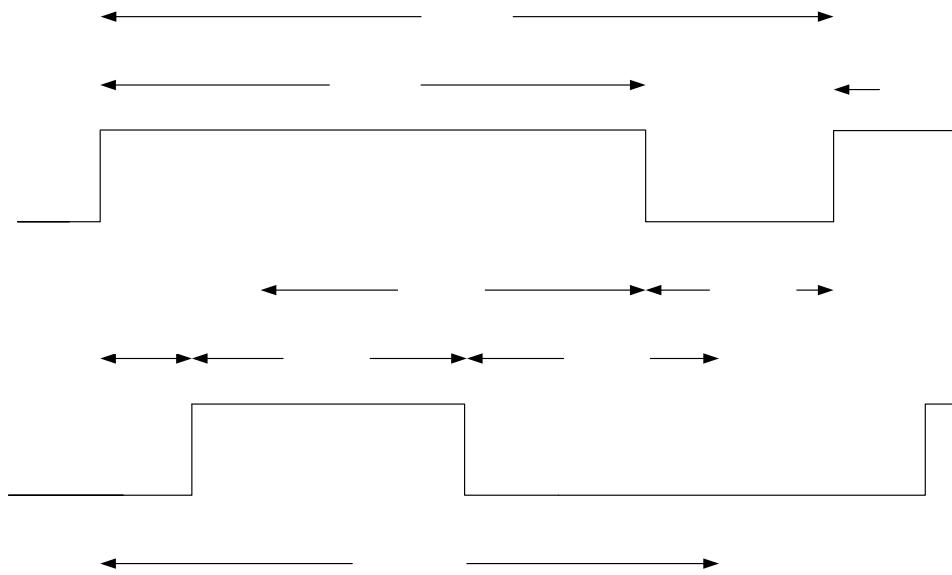
Every 4 input pixels are a unit. A unit is divided into 12 bytes. Output byte can select one of those 12 bytes.

Note that even line and odd line can be different, and output can be 12 bytes (4 pixels) or reduce to 6 bytes (2 pixels).

**Figure 5-4 CMAP Module****5.1.3.7 FRM Module****Figure 5-5 FRM Module****5.1.3.8 MIPI DSI**

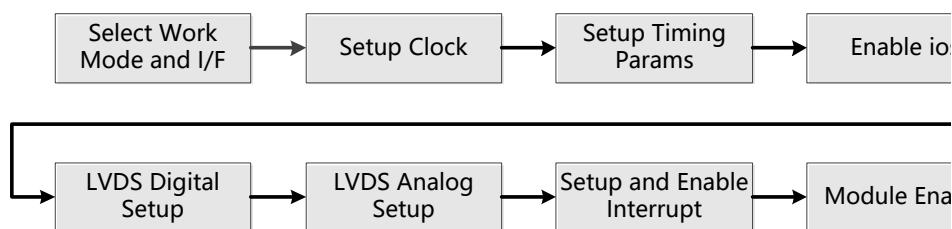
The requirements on MIPI DSI mode are as follows.

- (1). When using MIPI DSI as display interface, the data clk of TCON needs to be started firstly.
- (2). When it is used with DSI video mode, the setting of block space needs to meet the following relationship.

**Figure 5-6 The Data Timing of MIPI DSI Video Mode**

#### 5.1.4 Programming Guidelines

##### 5.1.4.1 LVDS Mode Configuration Process

**Figure 5-7 LVDS Mode Configuration Process****Step 1 Select HV interface type**

Configure LCD\_CTL\_REG[LCD\_IF] (reg0x40) to 0 to select HV (Sync+DE) mode, and configure LCD\_HV\_IF\_REG[HV\_MODE] (reg0x58) to 0 to select 24bit/1cycle parallel mode.

```
lcd_dev[sel]->lcd_ctl.lcd_if = HV(Sync+DE);
```

```
lcd_dev[sel]->lcd_ctl.src_sel = src; //src = DE/color/grayscale/...
```

```
lcd_dev[sel]->lcd_hv_ctl.hv_mode = 24bit/1cycle parallel mode;
```

**Step 2 Clock configuration**

**NOTE**

In parallel mode, the displayed pixel clock (pixel\_CLK) is required to be consistent with the DCLK,  $\text{pixel\_clk} = \text{Ht} * \text{Vt} * \text{frame rate}$ .

- Configure LCD\_DCLK\_REG.LCD\_DCLK\_DIV (reg0x44) to 7 after DCLK is determined;
- Configure the PLL clock in CCU based on proportional relationship;
- Release the LVDS reset of TCON LCD BUS GATING RESET register;
- Other configurations remain unchanged.

```
lcd_dev[sel]->lcd_dclk.dclk_en = en;
```

```
lcd_dev[sel]->lcd_dclk.dclk_div = 7;
```

**Step 3 Set sequence parameters**

The sequence parameters include x,ht,hbp,hspw,y,vt,vbp,vspw, and correspond to LCD\_BASE\_REG from reg0x48 to reg 0x54. Note that hbp includes hspw, and vbp includes vspw. And LCD\_BASE2\_REG.VT needs be set to the twice of the actual value.

```
lcd_dev[sel]->lcd_basic0.x = x-1;
```

```
lcd_dev[sel]->lcd_basic0.y = y-1;
```

```
lcd_dev[sel]->lcd_basic1.ht = ht-1;
```

```
lcd_dev[sel]->lcd_basic1.hbp = hbp-1;
```

```
lcd_dev[sel]->lcd_basic2.vt = vt*2;
```

```
lcd_dev[sel]->lcd_basic2.vbp = vbp-1;
```

```
lcd_dev[sel]->lcd_basic3.hspw = hspw-1;
```

```
lcd_dev[sel]->lcd_basic3.vspw = vspw-1;
```

**Step 4 Open IO output**

Set the corresponding data IO enable and control signal IO enable of LCD\_IO\_TRI\_REG (reg0x8C) to 0 to start enable. Note that except the internal IO of TCON\_LCD, the external GPIO mapping needs to be set to LCD mode.

When some control signals require polarity reversal, it can realize by setting LCD\_IO\_POL\_REG.IO0~3\_INV (reg0x88).

**Step 5 LVDS digital logic configuration**

Includes clock source select of module, LVDS link number, data mode and bit width configuration.

- Configure LCD\_LVDS\_IF\_REG.LCD\_LVDS\_CLK\_SEL (reg0x84) to set LCD CLK;

- Configure LCD\_LVDS\_IF\_REG.LCD\_LVDS\_LINK to set the required LVDS port number;
- Configure LCD\_LVDS\_IF\_REG.LCD\_LVDS\_MODE to set JEDIA and NS mode;
- Configure LCD\_LVDS\_IF\_REG.LCD\_LVDS\_BITWIDTH to select 24-bit or 18-bit width;
- Lastly configure LCD\_LVDS\_IF\_REG.LCD\_LVDS\_EN to start LVDS mode.

```
lcd_dev[sel]->lcd_lvds_ctl.lvds_link = link_num-1;  
lcd_dev[sel]->lcd_lvds_ctl.lvds_mode = mode;  
lcd_dev[sel]->lcd_lvds_ctl.lvds_bitwidth = bitwidth;  
lcd_dev[sel]->lcd_lvds_ctl.lvds_clk_sel = clk_src;  
lcd_dev[sel]->lcd_lvds_ctl.lvds_en = 1;
```



#### NOTE

If configuring the same source data output mode of dual link, except the reg0x84 register of TCON\_LCD0 needs be configured, the LCD\_LVDS\_IF\_REG.LCD\_LVDS\_CLK\_SEL, LCD\_LVDS\_IF\_REG.LCD\_LVDS\_LINK, LCD\_LVDS\_IF\_REG.LCD\_LVDS\_MODE, and LCD\_LVDS\_IF\_REG.LCD\_LVDS\_BITWIDTH of the reg0x244 register need be configured.

---

## Step 6 LVDS controller configuration

---



#### NOTE

The TCON LCD0 PHY0 is controlled by COMBO\_PHY\_REG (reg0x1110, reg0x1114). The TCON LCD0 PHY1 is controlled by LCD\_LVDS0\_ANA\_REG (reg0x220).

---

#### For PHY0:

- Configure the reg\_verf1p6 (differential mode voltage) in reg0x1114 to 4;
- Configure the reg\_vref0p8 reg0x1114 (common mode voltage) in reg0x1114 to 3;
- Start en\_cp, en\_mipi, en\_lvds, and en\_combaldo in reg0x1110, in turn.

#### For PHY1:

The LVDS analog configuration process is to start clock and data channel, and set the common mode and differential mode voltage, and start module power.

- Configure LVDS\_HPREN\_DRVC and LVDS\_HPREN\_DRV. When LVDS signal is 18-bit, LVDS\_HPREN\_DRV=0x7; when LVDS signal is 24-bit, LVDS\_HPREN\_DRV=0xF;
- Configure LVDS0\_REG\_C (differential mode voltage) to 4;
- Configure LVDS0\_REG\_V (common mode voltage) to 3;
- Lastly, start module voltage, and enable EN\_LVDS and EN\_24M.

```
lcd_dev[sel]->lcd_lvds_ana_ctl[lvds_num].bits.en_drvd = 0x7; //18-bit=0x7, 24-bit=0xf  
lcd_dev[sel]->lcd_lvds_ana_ctl[lvds_num].bits.en_drvc = 1;  
lcd_dev[sel]->lcd_lvds_ana_ctl[lvds_num].bits.diff_level = diff;  
lcd_dev[sel]->lcd_lvds_ana_ctl[lvds_num].bits.com_level = com;  
lcd_dev[sel]->lcd_lvds_ana_ctl[lvds_num].bits.dual_src = link_src;  
lcd_dev[sel]->lcd_lvds_ana_ctl[lvds_num].bits.en_24M = 1;  
lcd_dev[sel]->lcd_lvds_ana_ctl[lvds_num].bits.en_lvds = 1;  
lcd_dev[sel]->lcd_lvds_ana_ctl[lvds_num].bits.en_mb = 1;
```

### Step 7 Set and open interrupt function

The LCD\_GINT0\_REG (reg0x4) controls interrupt mode and flag, and the LCD\_GINT1\_REG (reg0x8) sets the interrupt line position of Line interrupt mode.

V interrupt:

```
lcd_dev[sel]->lcd_gint0.vb_en = 1;
```

Line interrupt:

```
lcd_dev[sel]->lcd_gint1.lcd_line_int_num = line;  
lcd_dev[sel]->lcd_gint0.line_en = 1;
```

### Step 8 Open module enable

Enable LCD\_CTL\_REG.LCD\_EN (reg0x40) and LCD\_GCTL\_REG.LCD\_EN (reg0x00).

```
lcd_dev[sel]->lcd_ctl.lcd_en = 1;  
lcd_dev[sel]->lcd_gctl.lcd_en = 1;
```

### 5.1.5 Register List

Module Name	Base Address
TCON_LCD0	0x05461000

Register Name	Offset	Description
LCD_GCTL_REG	0x0000	LCD Global Control Register
LCD_GINT0_REG	0x0004	LCD Global Interrupt Register0
LCD_GINT1_REG	0x0008	LCD Global Interrupt Register1
LCD_FRM_CTL_REG	0x0010	LCD FRM Control Register
LCD_FRM_SEED_REG	0x0014+N*0x04	LCD FRM Seed Register (N=0,1,2,3,4,5)
LCD_FRM_TAB_REG	0x002C+N*0x04	LCD FRM Table Register (N=0,1,2,3)
LCD_3D_FIFO_REG	0x003C	LCD 3D FIFO Register
LCD_CTL_REG	0x0040	LCD Control Register
LCD_DCLK_REG	0x0044	LCD Data Clock Register
LCD_BASIC0_REG	0x0048	LCD Basic Timing Register0
LCD_BASIC1_REG	0x004C	LCD Basic Timing Register1
LCD_BASIC2_REG	0x0050	LCD Basic Timing Register2
LCD_BASIC3_REG	0x0054	LCD Basic Timing Register3
LCD_HV_IF_REG	0x0058	LCD HV Panel Interface Register
LCD_CPU_IF_REG	0x0060	LCD CPU Panel Interface Register
LCD_CPU_WR_REG	0x0064	LCD CPU Panel Write Data Register
LCD_CPU_RDO_REG	0x0068	LCD CPU Panel Read Data Register0
LCD_CPU_RD1_REG	0x006C	LCD CPU Panel Read Data Register1
LCD_LVDS_IF_REG	0x0084	LCD LVDS Configure Register
LCD_IO_POL_REG	0x0088	LCD IO Polarity Register
LCD_IO_TRI_REG	0x008C	LCD IO Control Register
LCD_DEBUG_REG	0x00FC	LCD Debug Register
LCD_CEU_CTL_REG	0x0100	LCD CEU Control Register
LCD_CEU_COEF_MUL_REG	0x0110+N*0x04	LCD CEU Coefficient Register0 (N=0~10)
LCD_CEU_COEF_ADD_REG	0x011C+N*0x10	LCD CEU Coefficient Register1 (N=0,1,2)
LCD_CEU_COEF_RANG_REG	0x0140+N*0x04	LCD CEU Coefficient Register2 (N=0,1,2)
LCD_CPU_TRI0_REG	0x0160	LCD CPU Panel Trigger Register0
LCD_CPU_TRI1_REG	0x0164	LCD CPU Panel Trigger Register1
LCD_CPU_TRI2_REG	0x0168	LCD CPU Panel Trigger Register2
LCD_CPU_TRI3_REG	0x016C	LCD CPU Panel Trigger Register3
LCD_CPU_TRI4_REG	0x0170	LCD CPU Panel Trigger Register4
LCD_CPU_TRI5_REG	0x0174	LCD CPU Panel Trigger Register5
LCD_CMAP_CTL_REG	0x0180	LCD Color Map Control Register
LCD_CMAP_ODD0_REG	0x0190	LCD Color Map Odd Line Register0
LCD_CMAP_ODD1_REG	0x0194	LCD Color Map Odd Line Register1
LCD_CMAP_EVEN0_REG	0x0198	LCD Color Map Even Line Register0
LCD_CMAP_EVEN1_REG	0x019C	LCD Color Map Even Line Register1

Register Name	Offset	Description
LCD_SAFE_PERIOD_REG	0x01F0	LCD Safe Period Register
LCD_LVDS0_ANA_REG	0x0220	LCD LVDS Analog Register 0
LCD_SYNC_CTL_REG	0x0230	LCD Sync Control Register
LCD_SYNC_POS_REG	0x0234	LCD Sync Position Register
LCD_SLAVE_STOP_POS_REG	0x0238	LCD Slave Stop Position Register
LCD_GAMMA_TABLE_REG	0x0400-0x07FF	LCD Gamma Table Register

### 5.1.6 Register Description

#### 5.1.6.1 0x0000 LCD Global Control Register (Default Value: 0x0000\_0000)

Offset: 0x0000			Register Name: LCD_GCTL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<p>LCD_EN</p> <p>When it is disabled, the module will be reset to idle state.</p> <p>0: Disable 1: Enable</p>
30	R/W	0x0	<p>LCD_GAMMA_EN</p> <p>Enable the Gamma correction function.</p> <p>0: Disable 1: Enable</p>
29:0	/	/	/

#### 5.1.6.2 0x0004 LCD Global Interrupt Register0 (Default Value: 0x0000\_0000)

Offset: 0x0004			Register Name: LCD_GINT0_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<p>LCD_VB_INT_EN</p> <p>Enable the Vb interrupt</p> <p>0: Disable 1: Enable</p>
30	/	/	/
29	R/W	0x0	<p>LCD_LINE_INT_EN</p> <p>Enable the line interrupt</p> <p>0: Disable 1: Enable</p>
28	/	/	/

Offset: 0x0004			Register Name: LCD_GINT0_REG
Bit	Read/Write	Default/Hex	Description
27	R/W	0x0	LCD_TRI_FINISH_INT_EN Enable the trigger finish interrupt 0: Disable 1: Enable
26	R/W	0x0	LCD_TRI_COUNTER_INT_EN Enable the trigger counter interrupt 0: Disable 1: Enable
25:16	/	/	/
15	R/WOC	0x0	LCD_VB_INT_FLAG Asserted during vertical no-display period every frame Write 0 to clear it.
14	/	/	/
13	R/WOC	0x0	LCD_LINE_INT_FLAG Triggered when SY0 matches the current LCD scan line Write 0 to clear it.
12	/	/	/
11			
10	R/WOC	0x0	LCD_TRI_COUNTER_INT_FLAG Triggered when tri-counter reaches this value Write 0 to clear it.
9	R/WOC	0x0	LCD_TRI_UNDERFLOW_FLAG Only used in DSI video mode, trigger this bit when sync of DIS is not finished Write 0 to clear it.
8:3	/	/	/
2	R/WOC	0x0	FSYNC_INT_INV Enable the fsync interrupt to set signal inverse polarity. When FSYNC is positive, this bit must be 1. Vice versa.
1	R/WOC	0x0	DE_INT_FLAG Asserted at the first valid line in every frame Write 0 to clear it.
0	R/WOC	0x0	FSYNC_INT_FLAG Asserted at the fsync signal in every frame Write 0 to clear it.

### 5.1.6.3 0x0008 LCD Global Interrupt Register1 (Default Value: 0x0000\_0000)

Offset: 0x0008			Register Name: LCD_GINT1_REG
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27:16	R/W	0x0	LCD_LINE_INT_NUM Scan line for LCD line trigger (including inactive lines). Set it for the specified line for trigger0. <b>Note: SY0 is writable only when LINE_TRG0 is disabled.</b>
15:0	/	/	/

### 5.1.6.4 0x0010 LCD FRM Control Register (Default Value: 0x0000\_0000)

Offset: 0x0010			Register Name: LCD_FRM_CTL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	LCD_FRM_EN Enable the dither function 0: Disable 1: Enable
30:7	/	/	/
6	R/W	0x0	LCD_FRM_MODE_R The R component output bits in dither function 0: 6-bit FRM output 1: 5-bit FRM output
5	R/W	0x0	LCD_FRM_MODE_G The G component output bits in dither function 0: 6-bit frm output 1: 5-bit frm output
4	R/W	0x0	LCD_Frm_MODE_B The B component output bits in dither function 0: 6-bit frm output 1: 5-bit frm output
3:2	/	/	/
1:0	R/W	0x0	LCD_FRM_TEST Set the test mode of dither function 00: FRM 01: Half 5-/6-bit, half FRM 10: Half 8-bit, half FRM 11: Half 8-bit, half 5-/6-bit

### 5.1.6.5 0x0014+ N\*0x04 (N=0~5) LCD FRM Seed Register (Default Value: 0x0000\_0000)

Offset: 0x0014+N*0x04 (N=0~5)			Register Name:
Bit	Read/Write	Default/Hex	Description
31:25	/	/	/
24:0	R/W	0x0	<p>SEED_VALUE                      Set the seed used in dither function                      N=0: Pixel_Seed_R                      N=1: Pixel_Seed_G                      N=2: Pixel_Seed_B                      N=3: Line_Seed_R                      N=4: Line_Seed_G                      N=5: Line_Seed_B  <b>Note: Avoid setting it to 0.</b></p>

### 5.1.6.6 0x002C+ N\*0x04 (N=0~3) LCD FRM Table Register (Default Value: 0x0000\_0000)

Offset: 0x002C+N*0x04 (N=0~3)			Register Name: LCD_FRM_TAB_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<p>FRM_TABLE_VALUE                      Set the data used in dither function                      Usually set as follows:                      Table0 = 0x01010000                      Table1 = 0x15151111                      Table2 = 0x57575555                      Table3 = 0x7f7f7777</p>

### 5.1.6.7 0x003C LCD 3D FIFO Register (Default Value: 0x0000\_0000)

Offset: 0x003C			Register Name: LCD_3D_FIFO_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<p>3D_FIFO_BIST_EN                      Enable the 3D fifo bist test function                      0: Disable                      1: Enable</p>
30:14	/	/	/

Offset: 0x003C			Register Name: LCD_3D_FIFO_REG
Bit	Read/Write	Default/Hex	Description
13:4	R/W	0x0	3D_FIFO_HALF_LINE_SIZE The number of data in half line=3D_FIFO_HALF_LINE_SIZE+1, only valid when 3D_FIFO_SETTING is set as 2.
3:2	/	/	/
1:0	R/W	0x0	3D_FIFO_SETTING Set the work mode of 3D FIFO 00: Bypass 01: Used as normal FIFO 10: Used as 3D interlace FIFO 11: Reserved

#### 5.1.6.8 0x0040 LCD Control Register (Default Value: 0x0000\_0000)

Offset: 0x0040			Register Name: LCD_CTL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	LCD_EN It executes at the beginning of the first blank line of LCD timing. 0: Disable 1: Enable
30:26	/	/	/
25:24	R/W	0x0	LCD_IF Set the interface type of LCD controller. 00: HV(Sync+DE) 01: 8080 I/F 1x: Reserved
23	R/W	0x0	LCD_RB_SWAP Enable the function to swap red data and blue data in FIFO 1. 0: Default 1: Swap RED and BLUE data at FIFO1
22	/	/	/
21	R/W	0x0	LCD_FIFO1_RST Writing 1 and then 0 to this bit will reset FIFO 1 <b>Note: The holding time of writing 1 must more than 1 DCLK</b>
20	R/W	0x0	LCD_INTERLACE_EN This flag is valid only when LCD_EN == 1 0: Disable 1: Enable
19:9	/	/	/

Offset: 0x0040			Register Name: LCD_CTL_REG
Bit	Read/Write	Default/Hex	Description
8:4	R/W	0x0	LCD_START_DLY The unit of delay is $T_{line}$ . This parameter is valid only when LCD_EN == 1
3	/	/	/
2:0	R/W	0x0	LCD_SRC_SEL LCD Source Select 000: DE 001: Color Check 010: Grayscale Check 011: Black by White Check 100: Test Data all 0 101: Test Data all 1 110: Reversed 111: Gridding Check

#### 5.1.6.9 0x0044 LCD Data Clock Register (Default Value: 0x0000\_0000)

Offset: 0x0044			Register Name: LCD_DCLK_REG
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	LCD_DCLK_EN LCD clock enable 0000: dclk_en = 0; dclk1_en = 0; dclk2_en = 0; dclkm2_en = 0; 0001: dclk_en = 1; dclk1_en = 0; dclk2_en = 0; dclkm2_en = 0; 0010: dclk_en = 1; dclk1_en = 0; dclk2_en = 0; dclkm2_en = 1; 0011: dclk_en = 1; dclk1_en = 1; dclk2_en = 0; dclkm2_en = 0; 0101: dclk_en = 1; dclk1_en = 0; dclk2_en = 1; dclkm2_en = 0; 1111: dclk_en = 1; dclk1_en = 1; dclk2_en = 1; dclkm2_en = 1; Others:Reversed
27:7	/	/	/
6:0	R/W	0x0	LCD_DCLK_DIV Tdclk = Tsclk/DCLKDIV <b>Note:</b> <b>1.If dclk1&amp;dclk2 are used, DCLKDIV &gt;=6</b> <b>2.If only dclk is used, DCLKDIV &gt;=1</b>

### 5.1.6.10 0x0048 LCD Basic Timing Register0 (Default Value: 0x0000\_0000)

Offset: 0x0048			Register Name: LCD_BASIC0_REG
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27:16	R/W	0x0	WIDTH_X Panel width is X+1
15:12	/	/	/
11:0	R/W	0x0	HEIGHT_Y Panel height is Y+1

### 5.1.6.11 0x004C LCD Basic Timing Register1 (Default Value: 0x0000\_0000)

Offset: 0x004C			Register Name: LCD_BASIC1_REG
Bit	Read/Write	Default/Hex	Description
31:29	/	/	/
28:16	R/W	0x0	HT Thcycle = (HT+1) * Tdclk Computation: 1) parallel: HT = X + BLANK Limitation: 1) parallel: HT >= (HBP +1) + (X+1) +2 2) serial 1: HT >= (HBP +1) + (X+1) *3+2 3) serial 2: HT >= (HBP +1) + (X+1) *3/2+2
15:12	/	/	/
11:0	R/W	0x0	HBP Horizontal back porch (in dclk) Thbp = (HBP +1) * Tdclk

### 5.1.6.12 0x0050 LCD Basic Timing Register2 (Default Value: 0x0000\_0000)

Offset: 0x0050			Register Name: LCD_BASIC2_REG
Bit	Read/Write	Default/Hex	Description
31:29	/	/	/
28:16	R/W	0x0	VT Tvt = (VT)/2 * Thsync VT/2 >= (VBP+1 ) + (Y+1) +2
15:12	/	/	/

Offset: 0x0050			Register Name: LCD_BASIC2_REG
Bit	Read/Write	Default/Hex	Description
11:0	R/W	0x0	VBP Tvbp = (VBP +1) * Thsync

#### 5.1.6.13 0x0054 LCD Basic Timing Register3 (Default Value: 0x0000\_0000)

Offset: 0x0054			Register Name: LCD_BASIC3_REG
Bit	Read/Write	Default/Hex	Description
31:26	/	/	/
25:16	R/W	0x0	HSPW Thspw = (HSPW+1) * Tdclk HT > (HSPW+1)
15:10	/	/	/
9:0	R/W	0x0	VSPW Tvspw = (VSPW+1) * Thsync VT/2 > (VSPW+1)

#### 5.1.6.14 0x0058 LCD HV Panel Interface Register (Default Value: 0x0000\_0000)

Offset: 0x0058			Register Name: LCD_HV_IF_REG
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	HV_MODE Set the HV mode of LCD controller 0000: 24-bit/1-cycle parallel mode 1000: 8-bit/3-cycle RGB serial mode (RGB888) 1010: 8-bit/4-cycle Dummy RGB (DRGB) 1011: 8-bit/4-cycle RGB Dummy (RGBD) 1100: 8-bit/2-cycle YUV serial mode (CCIR656)
27:26	R/W	0x0	RGB888_ODD_ORDER Serial RGB888 mode Output sequence at odd lines of the panel (line 1, 3, 5, 7...). 00: R→G→B 01: B→R→G 10: G→B→R 11: R→G→B

Offset: 0x0058			Register Name: LCD_HV_IF_REG
Bit	Read/Write	Default/Hex	Description
25:24	R/W	0x0	<p>RGB888_EVEN_ORDER Serial RGB888 mode Output sequence at even lines of the panel (line 2, 4, 6, 8...).</p> <p>00: R→G→B 01: B→R→G 10: G→B→R 11: R→G→B</p>
23:22	R/W	0x0	<p>YUV_SM Serial YUV mode Output sequence 2-pixel-pair of every scan line.</p> <p>00: YUYV 01: YVYU 10: UYVY 11: VYUY</p>
21:20	R/W	0x0	<p>YUV_EAV_SAV_F_LINE_DLY Set the delay line mode.</p> <p>00: F toggle right after active video line 01: delay 2 line (CCIR PAL) 10: delay 3 line (CCIR NTSC) 11: reserved</p>
19	R/W	0x0	<p>CCIR_CSC_DIS LCD convert source from RGB to YUV.</p> <p>0: Enable 1: Disable Only valid when HV mode is "1100".</p>
18:0	/	/	/

#### 5.1.6.15 0x0060 LCD CPU Panel Interface Register (Default Value: 0x0000\_0000)

Offset: 0x0060			Register Name: LCD_CPU_IF_REG
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	Reserved
27	/	/	/
26	R/W	0x0	Reserved
25	R/W	0x0	Reserved
24	/	/	/
23	R	0x0	<p>WR_FLAG The status of write operation.</p> <p>0: Write operation is finished 1: Write operation is pending</p>

Offset: 0x0060			Register Name: LCD_CPU_IF_REG
Bit	Read/Write	Default/Hex	Description
22	R	0x0	RD_FLAG The status of read operation. 0: Read operation is finished 1: Read operation is pending
21:18	/	/	/
17	R/W	0x0	AUTO Auto transfer mode If it is 1, all the valid data during this frame are written to panel. <b>Note: This bit is sampled by Vsync.</b>
16	R/W	0x0	FLUSH Direct transfer mode If it is enabled, FIFO1 is regardless of the HV timing, the pixels data keep being transferred unless the input FIFO was empty. Data output rate is controlled by DCLK.
15:4	/	/	/
3	R/W	0x0	TRI_FIFO_BIST_EN Entry address is 0xFF8 0: Disable 1: Enable
2	R/W	0x0	TRI_FIFO_EN Enable the trigger FIFO 0: Disable 1: Enable
1	R/W1S	0x0	TRI_START Software must make sure that write '1' only when this flag is '0'. Writing '1' starts a frame flush and writing '0' has no effect. This flag indicates the frame flush is running.
0	R/W	0x0	TRI_EN Enable trigger mode 0: Trigger mode disable 1: Trigger mode enable

## 5.1.6.16 0x0084 LCD LVDS Interface Register (Default Value: 0x0000\_0000)

Offset: 0x0084			Register Name: LCD_LVDS_IF_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0	LCD_LVDS_EN Enable LVDS interface 0: Disable 1: Enable
30	R/W	0	LCD_LVDS_LINK Select work in single link mode or dual link mode 0: Single link 1: Reserved
29	R/W	0	LCD_LVDS_EVEN_ODD_DIR Set the order of even field and odd field 0: normal 1: reverse
28	R/W	0	LCD_LVDS_DIR Set the LVDS direction 0: Normal 1: Reverse
27	R/W	0	LCD_LVDS_MODE Set the LVDS data mode 0: NS mode 1: JEIDA mode
26	R/W	0	LCD_LVDS_BITWIDTH Set the bit width of data 0: 24-bit 1: 18-bit
25	R/W	0	LCD_LVDS_DEBUG_EN Enable LVDS debug function 0: Disable 1: Enable
24	R/W	0	LCD_LVDS_DEBUG_MODE Set the output signal in debug mode 0: Mode0—Random data 1: Mode1—Output CLK period=7/2 LVDS CLK period
23	R/W	0	LCD_LVDS_CORRECT_MODE Set the LVDS correct mode 0: Mode0 1: Mode1
22:21	/	/	/

Offset: 0x0084			Register Name: LCD_LVDS_IF_REG
Bit	Read/Write	Default/Hex	Description
20	R/W	0	LCD_LVDS_CLK_SEL Select the clock source of LVDS 0: Reserved 1: LCD CLK
19:5	/	/	/
4	R/W	0	LCD_LVDS_CLK_POL Set the clock polarity of LVDS 0: Reverse 1: Normal
3:0	R/W	0	LCD_LVDS_DATA_POL Set the data polarity of LVDS 0: Reverse 1: Normal

#### 5.1.6.17 0x0088 LCD IO Polarity Register (Default Value: 0x0000\_0000)

Offset: 0x0088			Register Name: LCD_IO_POL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	IO_OUTPUT_SEL When it is set as '1', the D[23:0], io0, io1, io3 are sync to dclk. 0: Normal output 1: Register output
30:28	R/W	0x0	DCLK_SEL Set the phase offset of clock and data in hv mode. 000: Used DCLK0 (normal phase offset) 001: Used DCLK1 (1/3 phase offset) 010: Used DCLK2 (2/3 phase offset) 100: DCLK0/2 phase 0 101: DCLK0/2 phase 90 Others: Reserved
27	R/W	0x0	IO3_INV Enable invert function of IO3 0: Not invert 1: Invert
26	R/W	0x0	IO2_INV Enable invert function of IO2 0: Not invert 1: Invert

Offset: 0x0088			Register Name: LCD_IO_POL_REG
Bit	Read/Write	Default/Hex	Description
25	R/W	0x0	IO1_INV Enable invert function of IO1 0: Not invert 1: Invert
24	R/W	0x0	IO0_INV Enable invert function of IO0 0: Not invert 1: Invert
23:0	R/W	0x0	Data_INV LCD output port D[23:0] polarity control, with independent bit control. 0: Normal polarity 1: Invert the specify output

#### 5.1.6.18 0x008C LCD IO Control Register (Default Value: 0xFFFF\_FFFF)

Offset: 0x008C			Register Name: LCD_IO_TRI_REG
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27	R/W	0x1	IO3_OUTPUT_TRI_EN Enable the output of IO3 1: Disable 0: Enable
26	R/W	0x1	IO2_OUTPUT_TRI_EN Enable the output of IO2 1: Disable 0: Enable
25	R/W	0x1	IO1_OUTPUT_TRI_EN Enable the output of IO1 1: Disable 0: Enable
24	R/W	0x1	IO0_OUTPUT_TRI_EN Enable the output of IO0 1: Disable 0: Enable
23:0	R/W	0xFFFFFFFF	DATA_OUTPUT_TRI_EN LCD output port D[23:0] output enable, with independent bit control. 1: Disable 0: Enable

**5.1.6.19 0x00FC LCD Debug Register (Default Value: 0x0000\_0000)**

Offset: 0x00FC			Register Name: LCD_DEBUG_REG
Bit	Read/Write	Default/Hex	Description
31	R	0x0	LCD_FIFO_UNDERFLOW The flag shows whether the fifos in underflow status 0: Not underflow 1: Underflow
30	/	/	/
29	R	0x0	LCD_FIELD_POL The flag indicates the current field polarity 0: Second field 1: First field
28	/	/	/
27:16	R	0x0	LCD_CURRENT_LINE The current scan line
15:0	/	/	/

**5.1.6.20 0x0100 LCD CEU Control Register (Default Value: 0x0000\_0000)**

Offset: 0x0100			Register Name: LCD_CEU_CTL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	CEU_EN Enable CEU function 0: Bypass 1: Enable
30	R/W	0x0	BT656_F_MASK BT656 F Mask 0: Disable 1: Enable
29	R/W	0x0	BT656_F_MASK_VALUE BT656 F Mask Value
28:0	/	/	/

### 5.1.6.21 0x0110+N\*0x04 (N=0–10) LCD CEU Coefficient Register0 (Default Value: 0x0000\_0000)

Offset: 0x0110+N*0x04 (N=0–10)			Register Name: LCD_CEU_COEF_MUL_REG
Bit	Read/Write	Default/Hex	Description
31:13	/	/	/
12:0	R/W	0x0	CEU_COEF_MUL_VALUE Signed 13-bit value, range of (-16,16). N=0: Rr N=1: Rg N=2: Rb N=4: Gr N=5: Gg N=6: Gb N=8: Br N=9: Bg N=10: Bb

### 5.1.6.22 0x011C+N\*0x10 (N=0–2) LCD CEU Coefficient Register1 (Default Value: 0x0000\_0000)

Offset: 0x011C+N*0x10 (N=0–2)			Register Name: LCD_CEU_COEF_ADD_REG
Bit	Read/Write	Default/Hex	Description
31:19	/	/	/
18:0	R/W	0x0	CEU_COEF_ADD_VALUE Signed 19-bit value, range of (-16384, 16384). N=0: Rc N=1: Gc N=2: Bc

### 5.1.6.23 0x0140+N\*0x04 (N=0–2) LCD CEU Coefficient Register2 (Default Value: 0x0000\_0000)

Offset: 0x0140+N*0x04 (N=0–2)			Register Name: LCD_CEU_COEF_RANG_REG
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23:16	R/W	0x0	CEU_COEF_RANGE_MIN Unsigned 8-bit value, range of [0, 255].
15:8	/	/	/
7:0	R/W	0x0	CEU_COEF_RANGE_MAX Unsigned 8-bit value, range of [0, 255].

**5.1.6.24 0x0160 LCD CPU Panel Trigger Register0 (Default Value: 0x0000\_0000)**

Offset: 0x0160			Register Name: LCD_CPU_TRI0_REG
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27:16	R/W	0x0	BLOCK_SPACE The spaces between data blocks. It should be set >20*pixel.
15:12	/	/	/
11:0	R/W	0x0	BLOCK_SIZE The size of data block. It is usually set as X.

**5.1.6.25 0x0164 LCD CPU Panel Trigger Register1 (Default Value: 0x0000\_0000)**

Offset: 0x0164			Register Name: LCD_CPU_TRI1_REG
Bit	Read/Write	Default/Hex	Description
31:16	R	0x0	BLOCK_CURRENT_NUM Shows the current data block transmitting to panel.
15:0	R/W	0x0	BLOCK_NUM The number of data blocks. It is usually set as Y.

**5.1.6.26 0x0168 LCD CPU Panel Trigger Register2 (Default Value: 0x0020\_0000)**

Offset: 0x0168			Register Name: LCD_CPU_TRI2_REG
Bit	Read/Write	Default/Hex	Description
31:16	R/W	0x20	START_DLY $T_{dly} = (\text{Start\_Delay} + 1) * \text{be\_clk} * 8.$
15	R/W	0x0	TRANS_START_MODE Select the FIFOs used in CPU mode. 0: ECC_FIFO+TRI_FIFO 1: TRI_FIFO
14:13	R/W	0x0	SYNC_MODE Set the sync mode in CPU interface. 2'b0x: Auto 10: 0 11: 1
12:0	R/W	0x0	TRANS_START_SET Usual set as the length of a line.

**5.1.6.27 0x016C LCD CPU Panel Trigger Register3 (Default Value: 0x0000\_0000)**

Offset: 0x016C			Register Name: LCD_CPU_TRI3_REG
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29:28	R/W	0x0	TRI_INT_MODE When set as 01, the Tri_Counter_Int occurs in cycle of (Count_N+1)×(Count_M+1)×4 dclk. When set as 10 or 11, the io0 is mapped as TE input. 00: Disable 01: Counter mode 10: Te rising mode 11: Te falling mode
27:24	/	/	/
23:8	R/W	0x0	COUNTER_N The value of counter factor
7:0	R/W	0x0	COUNTER_M The value of counter factor

**5.1.6.28 0x0170 LCD CPU Panel Trigger Register4 (Default Value: 0x0000\_0000)**

Offset: 0x0170			Register Name: LCD_CPU_TRI4_REG
Bit	Read/Write	Default/Hex	Description
31:29	/	/	/
28	R/W	0x0	PLUG_MODE_EN Enable the plug mode used in DSI command mode. 0: Disable 1: Enable
27:25	/	/	/
24	R/W	0x0	A1_First_Valid Valid in first Block.
23:0	R/W	0x0	D23_TO_D0_First_Valid Valid in first Block.

**5.1.6.29 0x0174 LCD CPU Panel Trigger Register5 (Default Value: 0x0000\_0000)**

Offset: 0x0174			Register Name: LCD_CPU_TRI5_REG
Bit	Read/Write	Default/Hex	Description
31:25	/	/	/

Offset: 0x0174			Register Name: LCD_CPU_TRI5_REG
Bit	Read/Write	Default/Hex	Description
24	R/W	0x0	A1_NON_First_Valid Valid in Block except first.
23:0	R/W	0x0	D23_TO_D0_NON_First_Valid Valid in Block except first.

#### 5.1.6.30 0x0180 LCD Color Map Control Register (Default Value: 0x0000\_0000)

Offset: 0x0180			Register Name: LCD_CMAP_CTL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	COLOR_MAP_EN Enable the color map function. This module only works when X is divided by 4. 0: Bypass 1: Enable
30:1	/	/	/
0	R/W	0x0	OUT_FORMAT Set the pixel output format in color map function. 0: 4 pixel output mode: Out0 -> Out1 -> Out2 -> Out3 1: 2 pixel output mode: Out0 -> Out1

**5.1.6.31 0x0190 LCD Color Map Odd Line Register0 (Default Value: 0x0000\_0000)**

Offset: 0x0190			Register Name: LCD_CMAP_ODD0_REG
Bit	Read/Write	Default/Hex	Description
31:16	R/W	0x0	<p>OUT_ODD1 Indicates the output order of components. bit15-12: Reserved bit11-08: Out_Odd0[23:16] bit07-04: Out_Odd0[15:8] bit03-00: Out_Odd0[7:0]</p> <p>0000: in_b0 0001: in_g0 0010: in_r0 0011: Reserved 0100: in_b1 0101: in_g1 0110: in_r1 0111: Reserved 1000: in_b2 1001: in_g2 1010: in_r2 1011: Reserved 1100: in_b3 1101: in_g3 1110: in_r3 1111: Reserved</p>

Offset: 0x0190			Register Name: LCD_CMAP_ODD0_REG
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0x0	<p>OUT_ODD0 Indicates the output order of components. bit15-12: Reserved bit11-08: Out_Odd0[23:16] bit07-04: Out_Odd0[15:8] bit03-00: Out_Odd0[7:0]</p> <p>0000: in_b0 0001: in_g0 0010: in_r0 0011: Reserved 0100: in_b1 0101: in_g1 0110: in_r1 0111: Reserved 1000: in_b2 1001: in_g2 1010: in_r2 1011: Reserved 1100: in_b3 1101: in_g3 1110: in_r3 1111: Reserved</p>

**5.1.6.32 0x0194 LCD Color Map Odd Line Register1 (Default Value: 0x0000\_0000)**

Offset: 0x0194			Register Name: LCD_CMAP_ODD1_REG
Bit	Read/Write	Default/Hex	Description
31:16	R/W	0x0	<p>OUT_ODD3                      Indicates the output order of components.                      bit15-12: Reserved                      bit11-08: Out_Odd0[23:16]                      bit07-04: Out_Odd0[15:8]                      bit03-00: Out_Odd0[7:0]</p> <p>0000: in_b0                      0001: in_g0                      0010: in_r0                      0011: Reserved                      0100: in_b1                      0101: in_g1                      0110: in_r1                      0111: Reserved                      1000: in_b2                      1001: in_g2                      1010: in_r2                      1011: Reserved                      1100: in_b3                      1101: in_g3                      1110: in_r3                      1111: Reserved</p>

Offset: 0x0194			Register Name: LCD_CMAP_ODD1_REG
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0x0	<p>OUT_ODD2 Indicates the output order of components. bit15-12: Reserved bit11-08: Out_Odd0[23:16] bit07-04: Out_Odd0[15:8] bit03-00: Out_Odd0[7:0]</p> <p>0000: in_b0 0001: in_g0 0010: in_r0 0011: Reserved 0100: in_b1 0101: in_g1 0110: in_r1 0111: Reserved 1000: in_b2 1001: in_g2 1010: in_r2 1011: Reserved 1100: in_b3 1101: in_g3 1110: in_r3 1111: Reserved</p>

**5.1.6.33 0x0198 LCD Color Map Even Line Register0 (Default Value: 0x0000\_0000)**

Offset: 0x0198			Register Name: LCD_CMAP_EVEN0_REG
Bit	Read/Write	Default/Hex	Description
31:16	R/W	0x0	<p>OUT_EVEN1 Indicates the output order of components. bit15-12: Reserved bit11-08: Out_Odd0[23:16] bit07-04: Out_Odd0[15:8] bit03-00: Out_Odd0[7:0]</p> <p>0000: in_b0 0001: in_g0 0010: in_r0 0011: Reserved 0100: in_b1 0101: in_g1 0110: in_r1 0111: Reserved 1000: in_b2 1001: in_g2 1010: in_r2 1011: Reserved 1100: in_b3 1101: in_g3 1110: in_r3 1111: Reserved</p>

Offset: 0x0198			Register Name: LCD_CMAP_EVEN0_REG
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0x0	<p>OUT_EVEN0                      Indicates the output order of components.                      bit15-12: Reserved                      bit11-08: Out_Odd0[23:16]                      bit07-04: Out_Odd0[15:8]                      bit03-00: Out_Odd0[7:0]</p> <p>0000: in_b0                      0001: in_g0                      0010: in_r0                      0011: Reserved                      0100: in_b1                      0101: in_g1                      0110: in_r1                      0111: Reserved                      1000: in_b2                      1001: in_g2                      1010: in_r2                      1011: Reserved                      1100: in_b3                      1101: in_g3                      1110: in_r3                      1111: Reserved</p>

**5.1.6.34 0x019C LCD Color Map Even Line Register1 (Default Value: 0x0000\_0000)**

Offset: 0x019C			Register Name: LCD_CMAP_EVEN1_REG
Bit	Read/Write	Default/Hex	Description
31:16	R/W	0x0	<p>OUT_EVEN3 Indicates the output order of components. bit15-12: Reserved bit11-08: Out_Odd0[23:16] bit07-04: Out_Odd0[15:8] bit03-00: Out_Odd0[7:0]</p> <p>0000: in_b0 0001: in_g0 0010: in_r0 0011: Reserved 0100: in_b1 0101: in_g1 0110: in_r1 0111: Reserved 1000: in_b2 1001: in_g2 1010: in_r2 1011: Reserved 1100: in_b3 1101: in_g3 1110: in_r3 1111: Reserved</p>

Offset: 0x019C			Register Name: LCD_CMAP_EVEN1_REG
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0x0	<p>OUT_EVEN2 Indicates the output order of components.</p> <p>bit15-12: Reserved</p> <p>bit11-08: Out_Odd0[23:16]</p> <p>bit07-04: Out_Odd0[15:8]</p> <p>bit03-00: Out_Odd0[7:0]</p> <p>0000: in_b0 0001: in_g0 0010: in_r0 0011: Reserved</p> <p>0100: in_b1 0101: in_g1 0110: in_r1 0111: Reserved</p> <p>1000: in_b2 1001: in_g2 1010: in_r2 1011: Reserved</p> <p>1100: in_b3 1101: in_g3 1110: in_r3 1111: Reserved</p>

#### 5.1.6.35 0x01F0 LCD Safe Period Register (Default Value: 0x0000\_0000)

Offset: 0x01F0			Register Name: LCD_SAFE_PERIOD_REG
Bit	Read/Write	Default/Hex	Description
31:29	/	/	/
28:16	R/W	0x0	<p>SAFE_PERIOD_FIFO_NUM When the data length in line buffer is more than SAFE_PERIOD_FIFO_NUM, the LCD controller will allow dram controller to stop working to change frequency.</p>
15:4	R/W	0x0	<p>SAFE_PERIOD_LINE Set a fixed line and during the line time, the LCD controller allow dram controller to change frequency. The fixed line should be set in the blanking area.</p>
3	/	/	/

Offset: 0x01F0			Register Name: LCD_SAFE_PERIOD_REG
Bit	Read/Write	Default/Hex	Description
2:0	R/W	0x0	<p>SAFE_PERIOD_MODE Select the save mode</p> <p>000: unsafe 001: safe 010: safe when FIFO_CURR_NUM &gt; SAFE_PERIOD_FIFO_NUM 011: safe when FIFO_CURR_NUM &gt; SAFE_PERIOD_FIFO_NUM and sync is activated 100: safe at line</p>

#### 5.1.6.36 0x0220 LCD LVDS Analog Register0 (Default Value: 0x0000\_0000)

Offset: 0x0220			Register Name: LCD_LVDS_ANA0_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<p>LVDS_EN_MB Enable the bias circuit of the LVDS_Ana module.</p> <p>0: Disable 1: Enable</p>
30	R/W	0x0	Reserved
29	R/W	0x0	<p>EN_LVDS Enable LVDS</p>
28	R/W	0x0	<p>EN_24M Enable the 24M clock</p>
27:25	/	/	/
24	R/W	0x0	<p>LVDS_HPREN_DRVC Enable clock channel drive</p> <p>0: Disable 1: Enable</p>
23:20	R/W	0x0	<p>LVDS_HPREN_DRV Enable data channel[3:0] drive</p> <p>0: Disable 1: Enable</p>

Offset: 0x0220			Register Name: LCD_LVDS_ANA0_REG
Bit	Read/Write	Default/Hex	Description
19:17	R/W	0x0	<p>LVDS_REG_C Adjust current flowing through R<sub>load</sub> of Rx to change the differential signals amplitude.</p> <p>000: 216 mV 001: 252 mV 010: 276 mV 011: 312 mV 100: 336 mV 101: 372 mV 110: 395 mV 111: 432 mV</p>
16	R/W	0x0	<p>LVDS_REG_DENC Choose data output or PLL test clock output in LVDS_tx.</p>
15:12	R/W	0x0	<p>LVDS_REG_DEN Choose data output or PLL test clock output in LVDS_tx.</p>
11	/	/	/
10:8	R/W	0x0	<p>LVDS_REG_R Adjust current flowing through R<sub>load</sub> of R<sub>x</sub> to change the common signals amplitude.</p> <p>000: 0.925 V 001: 0.950 V 010: 0.975 V 011: 1.000 V 100: 1.025 V 101: 1.050 V 110: 1.075 V 111: 1.100 V</p>
7:5	/	/	/
4	R/W	0x0	<p>LVDS_REG_PLRC LVDS clock channel direction. 0: Normal 1: Reverse</p>
3:0	R/W	0x0	<p>LVDS_REG_PLR LVDS data channel [3:0] direction. 0: Normal 1: Reverse</p>

**5.1.6.37 0x0228 LCD FSYNC Generate Control Register (Default Value: 0x0000\_0000)**

Offset: 0x0228			Register Name: FSYNC_GEN_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:19	/	/	/
18:8	R/W	0x0	SENSOR_DIS_TIME Delay 0–2047 Hsync Period When hsync_pol_sel is 0, the actual delay is sensor_dis_time-1. When hsync_pol_sel is 1, the actual delay is sensor_dis_time.
7	/	/	/
6	R/W	0x0	SENSOR_ACT1_VALUE Sensor Active1 Value 0: Fsync outputs 0 at active_1 period 1: Fsync outputs 1 active_1 period
5	R/W	0x0	SENSOR_ACT0_VALUE Sensor Active0 Value 0: Fsync outputs 0 at active_0 period 1: Fsync outputs 1 at active_0 period
4	R/W	0x0	SENSOR_DIS_VALUE Sensor Disable Value 0: Fsync outputs 0 at disable period 1: Fsync disable period outputs 1 at disable period
3	/	/	/
2	R/W	0x0	H SYNC POL SEL Hsync Polarity Select 0: Normal 1: Opposite hsync to hysnc counter
1	R/W	0x0	SEL_VSYNC_EN Select Vsync Enable 0: Select vsync falling edge to start state machine 1: Select vsync rising edge to start state machine
0	R/W	0x0	FSYNC_GEN_EN Fsync Generate Enable 0: Disable 1: Enable

**5.1.6.38 0x022C LCD FSYNC Generate Delay Register (Default Value: 0x0000\_0000)**

Offset: 0x022C			Register Name: FSYNC_GEN_DLY_REG
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/

Offset: 0x022C			Register Name: FSYNC_GEN_DLY_REG
Bit	Read/Write	Default/Hex	Description
27:16	R/W	0x0	SENSOR_ACT0_TIME Delay 0~4095 Pixel clk Period The actual delay is sensor_act0_time+1.
15:12	/	/	/
11:0	R/W	0x0	SENSOR_ACT1_TIME Delay 0~4095 Pixel clk Period The actual delay is sensor_act1_time+1.

#### 5.1.6.39 0x0230 LCD Sync Control Register (Default Value: 0x0000\_0000)

Offset: 0x0230			Register Name: LCD_SYNC_CTL_REG
Bit	Read/Write	Default/Hex	Description
31:9	/	/	/
8	R/W	0x0	LCD_CTRL_WORK_MODE LCD Controller Work mode 0: Single DSI mode 1: Dual DSI mode
7:5	/	/	/
4	R/W	0x0	LCD_CYRL_SYNC_MASTER_SLAVE LCD Controller Sync Master Slave 0: Master 1: Slave <b>Note: Only use in Single DSI mode.</b>
3:1	/	/	/
0	R/W	0x0	LCD_CTRL_SYNC_MODE LCD Controller Sync Mode 0: Sync at the first time 1: Sync every frame <b>Note: Only use in Single DSI mode.</b>

#### 5.1.6.40 0x0234 LCD Sync Position Register (Default Value: 0x0000\_0000)

Offset: 0x0234			Register Name: LCD_SYNC_POS_REG
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/

Offset: 0x0234			Register Name: LCD_SYNC_POS_REG
Bit	Read/Write	Default/Hex	Description
27:16	R/W	0x0	<p>LCD_Sync_Pixel_Num Set the pixel number of master LCD controller which is used to trigger the slave LCD controller to start working. This value is the number of pixels between the trigger point and the end of the line.</p> <p>Tri pos = <math>Tline * LCD\_Sync\_Line\_Num + Tpixel * (HT - LCD\_Sync\_Pixel\_Num)</math></p> <p><b>Note: Only use in Single DSI mode.</b></p>
15:12	/	/	/
11:0	R/W	0x0	<p>LCD_Sync_Line_Num Set the line number of master LCD controller which is used to trigger the slave LCD controller to start working.</p> <p><b>Note: It is only set in master LCD controller. It is not necessarily to set in slave LCD controller.</b></p> <p>Tri pos = <math>Tline * LCD\_Sync\_Line\_Num + Tpixel * (HT - LCD\_Sync\_Pixel\_Num)</math></p> <p><b>Note: Only use in Single DSI mode.</b></p>

#### 5.1.6.41 0x0238 LCD Slave Stop Position Register (Default Value: 0x0000\_0000)

Offset: 0x0238			Register Name: LCD_SLAVE_STOP_POS_REG
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R/W	0x0	<p>STOP_VAL Set the stop position of the slave LCD. This value is the number of pixels between the stop position and the end of the HFP.</p> <p>Stop_pos = HFP - Stop_val. <math>0 &lt; Stop\_pos &lt; HFP - 2</math></p> <p><b>Note: Only use in Single DSI mode.</b></p>

#### 5.1.6.42 0x0400-0x07FF LCD Gamma Table Registers (Default Value: 0x0000\_0000)

Offset: 0x0400–0x07FF			Register Name: LCD_GAMMA_TABLE_REG
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23:16	R/W	0x0	RED_COMP Red Component

Offset: 0x0400–0x07FF			Register Name: LCD_GAMMA_TABLE_REG
Bit	Read/Write	Default/Hex	Description
15:8	R/W	0x0	GREEN_COMP Green Component
7:0	R/W	0x0	BLUE_COMP Blue Component

## 5.2 TCON TV

### 5.2.1 Overview

The Timing Controller\_TV (TCON\_TV) is a module that processes video signals received from systems using a complicated arithmetic and then generates control signals and transmits them to the TV panel driver IC.

The TCON\_TV includes the following features:

- Supports 10-bit pixel depth YUV444, and HV format output up to 4K@60Hz
  - Supports 8-bit pixel depth YUV444, and HV format output up to 4K@60Hz

### **5.2.2 Functional Description**

### **5.2.2.1 CEU Module**

Function: This module enhance color data from DE .

$$R' = Rr^*R + Rg^*G + Rb^*B$$

$$G' = Gr^*R + Gg^*G + Gb^*B$$

$$B' = Br^*R + Bg^*G + Bb^*B$$

**Note:**

Rr, Rg, Rb, ,Gr, Gg, Gb, Br, Bg, Bb      bool      0,1

R. G. B u10 [0-1023]

R' have the range of [Rmin, Rmax]

$G'$  have the range of  $[R_{\min}, R_{\max}]$

B' have the range of [Rmin, Rmax]

### 5.2.3 Register List

<b>Module Name</b>	<b>Base Address</b>
TC0N_TV0	0x05470000

<b>Register Name</b>	<b>Offset</b>	<b>Description</b>
TV_GCTL_REG	0x0000	TV Global Control Register
TV_GINTO_REG	0x0004	TV Global Interrupt Register0

Register Name	Offset	Description
TV_GINT1_REG	0x0008	TV Global Interrupt Register1
TV_SRC_CTL_REG	0x0040	TV Source Control Register
TV_CTL_REG	0x0090	TV Control Register
TV_BASIC0_REG	0x0094	TV Basic Timing Register0
TV_BASIC1_REG	0x0098	TV Basic Timing Register1
TV_BASIC2_REG	0x009C	TV Basic Timing Register2
TV_BASIC3_REG	0x00A0	TV Basic Timing Register3
TV_BASIC4_REG	0x00A4	TV Basic Timing Register4
TV_BASIC5_REG	0x00A8	TV Basic Timing Register5
TV_IO_POL_REG	0x0088	TV SYNC Signal Polarity Register
TV_IO_TRI_REG	0x008C	TV SYNC Signal IO Control Register
TV_DEBUG_REG	0x00FC	TV Debug Register
TV_CEU_CTL_REG	0x0100	TV CEU Control Register
TV_CEU_COEF_MUL_REG	0x0110+N*0x04(N=0~10)	TV CEU Coefficient Register0
TV_CEU_COEF_RANG_REG	0x0140+N*0x04(N=0~2)	TV CEU Coefficient Register2
TV_SAFE_PERIOD_REG	0x01F0	TV Safe Period Register
TV_FILL_CTL_REG	0x0300	TV Fill Data Control Register
TV_FILL_BEGIN_REG	0x0304+N*0x0C(N=0~2)	TV Fill Data Begin Register
TV_FILL_END_REG	0x0308+N*0x0C(N=0~2)	TV Fill Data End Register
TV_FILL_DATA_REG	0x030C+N*0x0C(N=0~2)	TV Fill Data Value Register
TV_DATA_IO_PLO_REG	0x0330	TCON Data IO Polarity Control0
TV_DATA_IO_POL1_REG	0x0334	TCON Data IO Polarity Control1
TV_DATA_IO_TRI0_REG	0x0338	TCON Data IO Enable Control0
TV_DATA_IO_TRI1_REG	0x033C	TCON Data IO Enable Control1
TV_PIXELDEPTH_MODE_REG	0x0340	TV Pixeldepth Mode Control Register

## 5.2.4 Register Description

### 5.2.4.1 0x0000 TV Global Control Register (Default Value: 0x0000\_0000)

Offset: 0x0000			Register Name: TV_GCTL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	TV_EN When it is disabled, the module will be reset to idle state. 0: Disable 1: Enable
30:2	/	/	/

Offset: 0x0000			Register Name: TV_GCTL_REG
Bit	Read/Write	Default/Hex	Description
1	R/W	0x0	CEC_DDC_PAD_SEL CEC DDC PAD Select 1: TCON_TV internal pad for cec scl sal 0: GPIO pad for cec scl sal
0	/	/	/

#### 5.2.4.2 0x0004 TV Global Interrupt Register0 (Default Value: 0x0000\_0000)

Offset: 0x0004			Register Name: TV_GINT0_REG
Bit	Read/Write	Default/Hex	Description
31	/	/	/
30	R/W	0x0	TV_VB_INT_EN TV Vb Interrupt Enable 0: Disable 1: Enable
29	/	/	/
28	R/W	0x0	TV_Line_Int_En TV Line Interrupt Enable 0: Disable 1: Enable
27:15	/	/	/
14	R/W	0x0	TV_VB_INT_FLAG TV Vb Interrupt Flag Asserted during vertical no-display period every frame. Write 0 to clear it.
13	/	/	/
12	R/W	0x0	TV_Line_Int_Flag TV Line Interrupt Flag Trigger when SY1 matches the current TV scan line Write 0 to clear it.
11:0	/	/	/

#### 5.2.4.3 0x0008 TV Global Interrupt Register1 (Default Value: 0x0000\_0000)

Offset: 0x0008			Register Name: TV_GINT1_REG
Bit	Read/Write	Default/Hex	Description
31:12	/	/	/

Offset: 0x0008			Register Name: TV_GINT1_REG
Bit	Read/Write	Default/Hex	Description
11:0	R/W	0	<p>TV_Line_Int_Num Scan line for TV line trigger(including inactive lines) Setit for the specified line for trigger 1. Note: SY1 is writable only when LINE_TRG1 is disabled.</p>

#### 5.2.4.4 0x0040 TV Source Control Register (Default Value: 0x0000\_0000)

Offset: 0x0040			Register Name: TV_SRC_CTL_REG
Bit	Read/Write	Default/Hex	Description
31:3	/	/	/
2:0	R/W	0	<p>TV_SRC_SEL TV Source Select 000: DE 001: Color Check 010: Grayscale Check 011: Black by White Check 100: Reserved 101: Reserved 111: Gridding Check</p>

#### 5.2.4.5 0x0090 TV Control Register (Default Value: 0x0000\_0000)

Offset: 0x0090			Register Name: TV_CTL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<p>TV_EN When enabling TCON_TV, this bit and the 0x0000[bit31] need to be enabled. 0: Disable 1: Enable</p>
30:9	/	/	/
8:4	R/W	0x0	<p>START_DELAY This is for DE0 and DE1.</p>
3:2	/	/	/

Offset: 0x0090			Register Name: TV_CTL_REG
Bit	Read/Write	Default/Hex	Description
1	R/W	0x0	<p>TV_SRC_SEL TV Source Select 0: Reserved 1: BLUE data</p> <p><b>Note:</b> The priority of this bit is higher than TV_SRC_SEL(bit[2:0]) in TV_SRC_CTL_REG.</p>
0	/	/	/

#### 5.2.4.6 0x0094 TV Basic Timing Register0 (Default Value: 0x0000\_0000)

Offset: 0x0094			Register Name: TV_BASIC0_REG
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27:16	R/W	0x0	WIDTH_XI Source Width Is X+1
15:12	/	/	/
11:0	R/W	0x0	HEIGHT_YI Source Height Is Y+1

#### 5.2.4.7 0x0098 TV Basic Timing Register1 (Default Value: 0x0000\_0000)

Offset: 0x0098			Register Name: TV_BASIC1_REG
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27:16	R/W	0x0	LS_XO Width Is LS_XO+1
15:12	/	/	/
11:0	R/W	0x0	LS_YO Width Is LS_YO+1 <b>Note: LS_YO = TV_YI in this version</b>

#### 5.2.4.8 0x009C TV Basic Timing Register2 (Default Value: 0x0000\_0000)

Offset: 0x009C			Register Name: TV_BASIC2_REG
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27:16	R/W	0x0	TV_XO Width is TV_XO+1
15:12	/	/	/
11:0	R/W	0x0	TV_YO Height is TV_YO+1

#### 5.2.4.9 0x00A0 TV Basic Timing Register3 (Default Value: 0x0000\_0000)

Offset: 0x00A0			Register Name: TV_BASIC3_REG
Bit	Read/Write	Default/Hex	Description
31:29	/	/	/
28:16	R/W	0x0	H_T Horizontal total time Thcycle = (HT+1) * Thdclk
15:12	/	/	/
11:0	R/W	0x0	H_BP Horizontal back porch Thbp = (HBP +1) * Thdclk

#### 5.2.4.10 0x00A4 TV Basic Timing Register4 (Default Value: 0x0000\_0000)

Offset: 0x00A4			Register Name: TV_BASIC4_REG
Bit	Read/Write	Default/Hex	Description
31:29	/	/	/
28:16	R/W	0x0	V_T Horizontal total time (in HD line) Tvt = VT/2 * Th
15:12	/	/	/
11:0	R/W	0x0	V_BP Horizontal back porch (in HD line) Tvbp = (VBP +1) * Th

**5.2.4.11 0x00A8 TV Basic Timing Register5 (Default Value: 0x0000\_0000)**

Offset: 0x00A8			Register Name: TV_BASIC5_REG
Bit	Read/Write	Default/Hex	Description
31:26	/	/	/
25:16	R/W	0x0	H_SPW Horizontal Sync Pulse Width (in dclk) Thspw = (HSPW+1) * Tdclk <b>Note: HT &gt; (HSPW+1)</b>
15:10	/	/	/
9:0	R/W	0x0	V_SPW Vertical Sync Pulse Width (in lines) Tvspw = (VSPW+1) * Th <b>Note: VT/2 &gt; (VSPW+1)</b>

**5.2.4.12 0x0088 TV SYNC Signal Polarity Register (Default Value: 0x0000\_0000)**

Offset: 0x0088			Register Name: TV_IO_POL_REG
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27	R/W	0x0	IO3_INV IO3 Invert 0: Not invert 1: Invert
26	R/W	0x0	IO2_INV IO2 Invert 0: Not invert 1: Invert
25	R/W	0x0	IO1_INV IO1 Invert 0: Not invert 1: Invert
24	R/W	0x0	IO0_INV IO0 Invert 0: Not invert 1: Invert
23:0	/	/	/

**5.2.4.13 0x008C TV SYNC Signal IO Control Register (Default Value: 0x0F00\_0000)**

Offset: 0x008C			Register Name: TV_IO_TRI_REG
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27	R/W	0x1	IO3_OUTPUT_TRI_EN IO3 Output Trigger Enable 1: Disable 0: Enable
26	R/W	0x1	IO2_OUTPUT_TRI_EN IO2 Output Trigger Enable 1: Disable 0: Enable
25	R/W	0x1	IO1_OUTPUT_TRI_EN IO1 Output Trigger Enable 1: Disable 0: Enable
24	R/W	0x1	IO0_OUTPUT_TRI_EN IO0 Output Trigger Enable 1: Disable 0: Enable
23:0	/	/	/

**5.2.4.14 0x00FC TV Debug Register (Default Value: 0x0000\_0000)**

Offset: 0x00FC			Register Name: TV_DEBUG_REG
Bit	Read/Write	Default/Hex	Description
31	/	/	/
30	R/W	0x0	TV_FIFO_U TV FIFO Underflow
29	/	/	/
28	R	0x0	TV_FIELD_POL TV Field Polarity 0: Second field 1: First field
27:12	/	/	/
13	R/W	0x0	LINE_BUF_BYPASS Line Buffer Bypass 0: Used 1: Bypass
12	/	/	/

Offset: 0x00FC			Register Name: TV_DEBUG_REG
Bit	Read/Write	Default/Hex	Description
11:0	R	0x0	TV_CURRENT_LINE TV Current Line

#### 5.2.4.15 0x0100 TV CEU Control Register (Default Value: 0x0000\_0000)

Offset: 0x0100			Register Name: TV_CEU_CTL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	CEU_EN CEU Enable 0: Bypass 1: Enable
30:0	/	/	/

#### 5.2.4.16 0x110+N\*0x04 (N=0~10) TV CEU Coefficient Register0 (Default Value: 0x0000\_0000)

Offset: 0x110+N*0x04 (N=0~10)			Register Name: TV_CEU_COEF_MUL_REG
Bit	Read/Write	Default/Hex	Description
31:9	/	/	/
8	R/W	0x0	CEU_COEF_MUL_VALUE Note: 1.CEU_Coef_Mul_Value only can be 0 or 1. 2. REG Map: N=0: Rr N=1: Rg N=2: Rb N=4: Gr N=5: Gg N=6: Gb N=8: Br N=9: Bg N=10: Bb
7:0	/	/	/

#### 5.2.4.17 0x0140+N\*0x04 (N=0~2) TV CEU Coefficient Register2 (Default Value: 0x0000\_0000)

Offset: 0x0140+N*0x04 (N=0~2)			Register Name: TV_CEU_COEF_RANG_REG
Bit	Read/Write	Default/Hex	Description
31:26	/	/	/
25:16	R/W	0	CEU_COEF_RANGE_MIN Unsigned 10-bit Value, range of [0, 1023]
15:10	/	/	/
9:0	R/W	0	CEU_COEF_RANGE_MAX Unsigned 10-bit Value, range of [0, 1023]

#### 5.2.4.18 0x01F0 TV Safe Period Register (Default Value: 0x0000\_0000)

Offset: 0x01F0			Register Name: TV_SAFE_PERIOD_REG
Bit	Read/Write	Default/Hex	Description
31:29	/	/	/
28:16	R/W	0x0	SAFE_PERIOD_FIFO_NUM Safe Period FIFO Number
15:4	R/W	0x0	Safe_Period_Line Safe Period Line
3	/	/	/
2:0	R/W	0x0	Safe_Period_Mode Safe Period Mode 000: unsafe 001: safe 010: safe when line_buf_curr_num > safe_period_fifo_num 011: safe when FIFO_CURR_NUM > SAFE_PERIOD_FIFO_NUM and sync is activated 100: safe at line

#### 5.2.4.19 0x0300 TV Fill Data Control Register (Default Value: 0x0000\_0000)

Offset: 0x0300			Register Name: TV_FILL_CTL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	TV_Fill_En TV Fill Enable 0: Bypass 1: Enable
30:0	/	/	/

#### 5.2.4.20 0x0304+N\*0x0C (N=0–2) TV Fill Data Begin Register (Default Value: 0x0000\_0000)

Offset: 0x0304+N*0x0C (N=0–2)			Register Name: TV_FILL_BEGIN_REG
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23:0	R/W	0	FILL_BEGIN Fill Begin

#### 5.2.4.21 0x0308+N\*0x0C (N=0–2) TV Fill Data End Register (Default Value: 0x0000\_0000)

Offset: 0x0308+N*0x0C (N=0–2)			Register Name: TV_FILL_END_REG
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23:0	R/W	0x0	FILL_END Fill End

#### 5.2.4.22 0x030C+N\*0x0C (N=0–2) TV Fill Data Value Register (Default Value: 0x0000\_0000)

Offset: 0x030C+N*0x0C (N=0–2)			Register Name: TV_FILL_DATA_REG
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29:0	R/W	0x0	FILL_VALUE Fill Value

#### 5.2.4.23 0x0330 TCON Data IO Polarity Control0 (Default Value: 0x0000\_0000)

Offset: 0x0330			Register Name: TV_DATA_IO_POL0_REG
Bit	Read/Write	Default/Hex	Description
31:26	/	/	/
25:16	R/W	0x0	R_CB_CH_DATA_INV R CB Channel Data Inv 0: Normal polarity 1: Invert the specify output
15:10	/	/	/

Offset: 0x0330			Register Name: TV_DATA_IO_POL0_REG
Bit	Read/Write	Default/Hex	Description
9:0	R/W	0x0	G_Y_CH_DATA_INV G Y Channel Data Inv 0: normal polarity 1: invert the specify output

#### 5.2.4.24 0x0334 TCON Data IO Polarity Control1 (Default Value: 0x0000\_0000)

Offset: 0x0334			Register Name: TV_DATA_IO_POL1_REG
Bit	Read/Write	Default/Hex	Description
31:26	/	/	/
25:16	R/W	0x0	B_CR_CH_DATA_INV B CR CHANNEL DATA INV 0: Normal polarity 1: Invert the specify output
15:0	/	/	/

#### 5.2.4.25 0x0338 TCON Data IO Enable Control0 (Default Value: 0x03FF\_03FF)

Offset: 0x0338			Register Name: TV_DATA_IO_TRI0_REG
Bit	Read/Write	Default/Hex	Description
31:26	/	/	/
25:16	R/W	0x3ff	R_CB_CH_DATA_OUT_TRI_EN R CB Channel Data Output Trigger Enable 1: Disable 0: Enable
15:10	/	/	/
9:0	R/W	0x3ff	G_Y_CH_DATA_OUT_TRI_EN G Y Channel Data Output Trigger Enable 1: Disable 0: Enable

**5.2.4.26 0x033C TCON Data IO Enable Control1 (Default Value: 0x03FF\_0000)**

Offset: 0x033C			Register Name: TV_DATA_IO_TRI1_REG
Bit	Read/Write	Default/Hex	Description
31:26	/	/	/
25:16	R/W	0x3ff	B_CR_CH_DATA_OUT_TRI_EN B CR Channel Data Output Trigger Enable 1: Disable 0: Enable
15:0	/	/	/

**5.2.4.27 0x0340 TV Pixeldepth Mode Control Register (Default Value: 0x0000\_0000)**

Offset: 0x0340			Register Name: TV_PIXELDEPTH_MODE_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	COLORBAR_PD_MODE Colorbar Pixeldepth mode 0: 8-bit mode When data source is the embedded colorbar, the 8-bit colorbar pattern is transmitted. 1: 10-bit mode When data source is the embedded colorbar, the 10-bit colorbar pattern is transmitted.

## 5.3 MIPI DSI

### 5.3.1 Overview

The Display Serial Interface is a high-speed interface between a host processor and peripheral devices that adhere to MIPI Alliance specifications for mobile device interfaces. This DSI module is composed of a DSI controller which is compliance with MIPI DSI specification V1.01 and a D-PHY module which is compliance with MIPI DPHY specification V1.00.

The MIPI DSI includes the following features:

- Compliance with MIPI DSI v1.01
- Up to 4 lanes
- Supports 1280 x 720@60fps and 1920 x 1200@60fps
- Supports non-burst mode with sync pulse/sync event and burst mode
- Supports pixel format: RGB888, RGB666, RGB666 loosely packed and RGB565
- Supports continuous lane clock mode and non-continuous lane clock mode
- Compliance with MIPI DCS v1.01, bidirectional communication in LP through data lane 0
- Supports bidirectional communication of all generic commands in LP through data lane 0
- Supports low power data transmission
- Supports ULPS and Escape modes
- Hardware checksum capabilities

## 5.4 HDMI

### 5.4.1 Overview

The High Definition Multimedia Interface (HDMI) is a wired digital interconnect that replaces the analog SCART connection. HDMI can transfer uncompressed video, audio, and data using a single cable.

The system architecture of HDMI consists of sources (transmitter) and sinks (receiver). The HDMI cable and connectors carry four differential pairs that make up the TMDS data and clock channels. These channels are used to carry video, audio, and auxiliary data. In addition, HDMI carries a VESA Data Display Channel (DDC). The DDC is used for configuration and status exchange between a single source and a single sink. The optional CEC protocol provides high-level control functions between all of the various audiovisual products in a user's environment.

The HDMI includes the following features:

- Compatible with HDMI 1.4
- Supports DDC
- Integrated CEC hardware engine
- Optional color space converter (CSC): RGB (4:4:4) to/from YCbCr (4:4:4 or 4:2:2)
- Video formats:
  - Optional HDMI 1.4b video formats
  - All CEA-861-E video formats up to 1080p at 120 Hz
  - HDMI 1.4b 4K x 2K video formats
  - HDMI 1.4b 3D video modes with up to 340 MHz (TMDS clock)
- Audio formats:
  - Uncompressed audio formats: IEC60985 L-PCM audio samples, up to 192 kHz
  - Compressed audio formats: IEC61937 compressed audio, up to 768 kHz

## Contents

6	Memory.....	431
6.1	SDRAM Controller (DRAMC) .....	431
6.2	SD/MMC Host Controller (SMHC) .....	432
6.2.1	Overview .....	432
6.2.2	Block Diagram .....	432
6.2.3	Functional Description .....	433
6.2.4	Programming Guidelines.....	445
6.2.5	Register List.....	451
6.2.6	Register Description .....	452

# Figures

Figure 6-1 SMHC Block Diagram .....	433
Figure 6-2 Command Token Format .....	436
Figure 6-3 Response Token Format .....	436
Figure 6-4 Data Packet Format for SDR .....	437
Figure 6-5 Data Packet Format for DDR.....	438
Figure 6-6 Single-Block and Multi-Block Read Operation .....	439
Figure 6-7 Single-Block and Multi-Block Write Operation .....	439
Figure 6-8 Phase Offset of Command and Data in SDR Mode .....	441
Figure 6-9 Phase Offset of Command and Data in DDR4 Mode (SMHC_NTSR[31] = 0) .....	441
Figure 6-10 Phase Offset of Command and Data in DDR4 Mode .....	441
Figure 6-11 Phase Offset of Command and Data in DDR4 (2x Mode) (SMHC_NTSR[31] = 1) .....	442
Figure 6-12 IDMAC Descriptor Structure Diagram.....	443

## Tables

Table 6-1 SMHC Sub-blocks .....	433
Table 6-2 SMHC External Signals.....	434
Table 6-3 SMHC0/1 Clock Sources .....	434
Table 6-4 SMHC2 Clock Sources.....	434
Table 6-5 Speed Modes Supported by SD 3.0.....	440
Table 6-6 Speed Modes Supported by eMMC 5.0 .....	440
Table 6-7 Phase Offset of Command and Data in SDR Mode .....	441
Table 6-8 Phase Offset of Command and Data in DDR4 (2x) Mode .....	442
Table 6-9 DES0 Definition .....	443
Table 6-10 DES1 Definition .....	444
Table 6-11 DES2 Definition .....	444
Table 6-12 DES3 Definition .....	444

# 6 Memory

## 6.1 SDRAM Controller (DRAMC)

The DRAMC provides a simple, flexible, burst-optimized interface to all industry-standard DDR2/DDR3 SDRAM. It supports up to a 16 Gbits memory address space.

The DRAMC automatically handles memory management, initialization, and refresh operations. It gives the host CPU a simple command interface, hiding details of the required address, page, and burst handling procedures. All memory parameters are runtime-configurable, including timing, memory setting, SDRAM type, and Extended-Mode-Register settings.

The DRAMC has the following features:

- Supports 16-bit one channel
- Supports 2 Chip Select
- Supports DDR2/DDR3 SDRAM
- Maximum capacity up to 2 GB
- Supports clock frequency up to 533 MHz for DDR2
- Supports clock frequency up to 800 MHz for DDR3

## 6.2 SD/MMC Host Controller (SMHC)

### 6.2.1 Overview

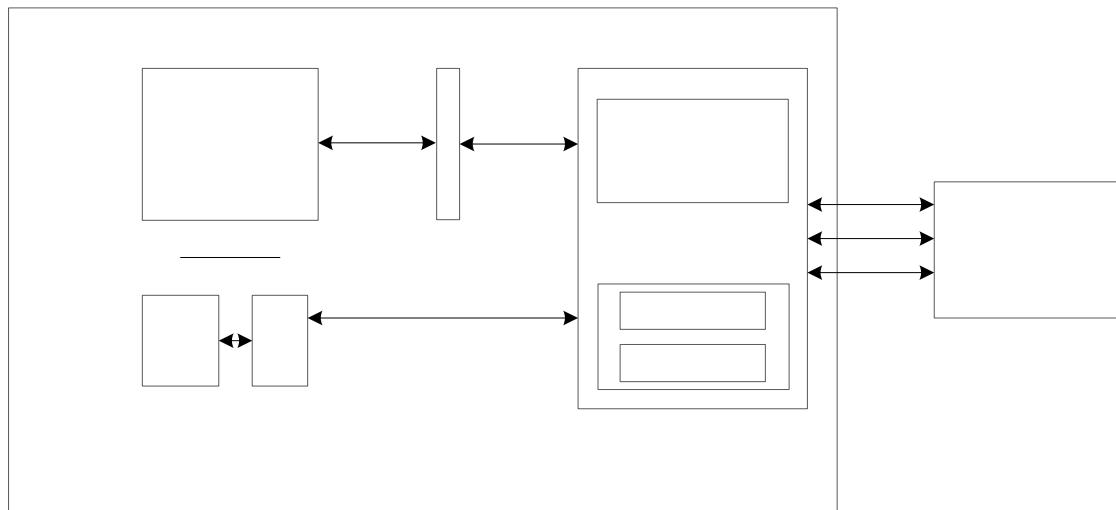
The SMHC controls the read/write operations on the secure digital (SD) cards, multimedia cards (MMC), and various extended devices that is based on the secure digital input/output (SDIO) protocol. The processor provides three SMHC interfaces for controlling the SD cards, MMCs, and SDIO devices.

The SMHC has the following features:

- Supports Command Completion signals and interrupts to host processor, and Command Completion signal disable feature
- The SMHC0 controls the devices that comply with the Secure Digital Memory (SD mem-version 3.0)
- The SMHC1 controls the device that complies with the protocol Secure Digital I/O (SDIO-version 3.0)
- The SMHC2 controls the device that complies with the Multimedia Card (eMMC-version 5.0)
- Maximum performance:
  - SDR mode: 150 MHz@1.8 V IO pad
  - DDR mode: 50 MHz@1.8 V IO pad
  - DDR mode: 50 MHz@3.3 V IO pad
- Supports 1-bit or 4-bit data width
- Hardware CRC generation and error detection
- Programmable baud rate
- Supports SDIO interrupt in 1-bit and 4-bit modes
- Block size of 1 to 65535 bytes
- Descriptor-based internal DMA controller
- Internal 1 KB RXFIFO and 1 KB TXFIFO

### 6.2.2 Block Diagram

The following figure shows a block diagram of the SMHC.

**Figure 6-1 SMHC Block Diagram**

SMHC contains the following sub-blocks:

**Table 6-1 SMHC Sub-blocks**

Sub-block	Description
Register	Used to configure the control signal for reading or writing the SD/SDIO/eMMC.
DMAC	The DMA controller controls the data transfer between the memory and SMHC.
FIFO	A buffer for the data stream between the memory and the SMHC asynchronous clock domain.
SYNC	Synchronizes the signals from the AHB clock domain to the SMHC clock domain.
CMD Path	Sends commands to or receives commands from the SD/SDIO/eMMC.
Data Path	Consists of Data TX and Data RX sub-modules. The Data TX sends data blocks and the CRC codes to the SD/SDIO/eMMC. The Data RX receives data blocks and the CRC codes from the SD/SDIO/eMMC.

### 6.2.3 Functional Description

#### 6.2.3.1 External Signals

The following table describes the external signals of SMHC.

**Table 6-2 SMHC External Signals**

<b>Port Name</b>	<b>Type</b>	<b>Description</b>
SDC0-CMD	I/O, OD	Command Signal for SD Card
SDC0-CLK	O	Clock for SD Card
SDC0-D[3:0]	I/O	Data Input and Output for SD Card
SDC0-RST	O	Reset for SD Card
SDC1-CMD	I/O, OD	Command Signal for SDIO Wi-Fi
SDC1-CLK	O	Clock for SDIO Wi-Fi
SDC1-D[3:0]	I/O	Data Input and Output for SDIO Wi-Fi
SDC2-CMD	I/O, OD	Command Signal for eMMC
SDC2-CLK	O	Clock for eMMC
SDC2-D[3:0]	I/O	Data Input and Output for eMMC

### 6.2.3.2 Clock Sources

The SMHC0/1 has 4 different clock sources. The SMHC2 has 5 different clock sources. You can select one of them as the SMHC clock source. The following table describes the clock sources of the SMHC.

For clock setting, configurations, and gating information, refer to section 3.3 “[CCU](#)”.

**Table 6-3 SMHC0/1 Clock Sources**

<b>Clock Sources</b>	<b>Description</b>
HOSC	24 MHz Crystal
PLL_PERI(1X)	Peripheral Clock, the default value is 600 MHz
PLL_PERI(2X)	Peripheral Clock, the default value is 1.2 GHz
PLL_AUDIO1(DIV2)	Audio clock, the default value is 1536 MHz

**Table 6-4 SMHC2 Clock Sources**

<b>Clock Sources</b>	<b>Description</b>
HOSC	24 MHz Crystal
PLL_PERI(1X)	Peripheral Clock, the default value is 600 MHz
PLL_PERI(2X)	Peripheral Clock, the default value is 1.2 GHz
PLL_PERI(800M)	Peripheral Clock, the default value is 800 MHz
PLL_AUDIO1(DIV2)	Audio clock, the default value is 1536 MHz

### 6.2.3.3 Timing Diagram

Refer to the following relative specifications:

- Physical Layer Specification Ver3.00 Final
- SDIO Specification Ver2.00
- Multimedia Cards (MMC – version 4.2)
- JEDEC Standard – JESD84-44, Embedded Multimedia Card (eMMC) Card Product Standard
- JEDEC Standard – JESD84-B45, Embedded Multimedia Card (eMMC) Electrical Standard (4.5 Device)
- JEDEC Standard – JESD84-B50, Embedded Multimedia Card (eMMC) Electrical Standard (5.0)

### 6.2.3.4 Data Path

The SMHC and SD/SDIO/eMMC contains the following interface buses: CLK, CMD, and DATA 1/4. During one clock cycle, the SMHC can transmit one bit command with one or two bits data in 1-ch DATA mode, or four or eight bits data in 4-ch DATA mode. The CMD is a bidirection channel for initializing the SD/SDIO/eMMC and transmitting commands. It can work in both the open-drain mode and push-pull mode. The DATA is also a bidirection channel. It works in the push-pull mode.

#### Reading Data from the SD/SDIO/eMMC

The register configures the signals for the read operation, and synchronize the signals to the SMHC clock domain. Then the Data RX reads data from the SD/SDIO/eMMC via the CLK/CMD/DATA interface buses and writes the data in the FIFO. After that, the DMAC transfers the data from the FIFO to the memory.

#### Writing Data to the SD/SDIO/eMMC

The register configures the signals for the write operation, and synchronize the signals to the SMHC clock domain. Then the DMAC reads data from the memory and writes the data to the FIFO. After that, the Data TX reads the data from the FIFO and writes the data to the SD/SDIO/eMMC via the CLK/CMD/DATA interface buses.

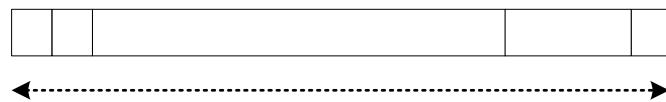
### 6.2.3.5 Package Format

Data transfer over the SD/eMMC bus is based on command and data bitstreams that are initiated by a start bit and terminated by a stop bit. There are three types of SD/eMMC packets: command token, response token, and data packet.

### Command Tokens

The command token starts an operation. A command is sent from the host to a device. It is transferred serially on the CMD line. Command tokens have the following coding scheme:

**Figure 6-2 Command Token Format**



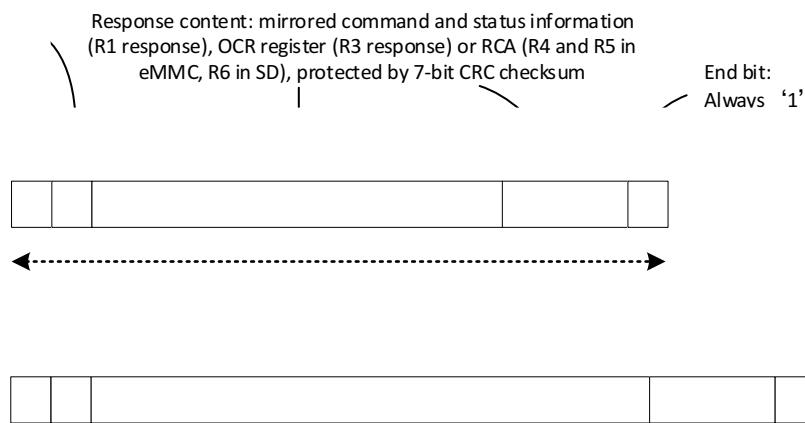
Each command token has 48 bits, preceded by a start bit ('0') and succeeded by an end bit ('1'). To detect transmission errors, each token is protected by CRC bits.

### Response Tokens

After receiving a command, the card returns a 48-bit or 136-bit response based on the command type.

A response token is sent from the device to the host as an answer to a previously received command. It is transferred serially on the CMD line.

**Figure 6-3 Response Token Format**



### Data Packets

Data can be transferred from the device to the host or vice versa. Data are transferred via the data lines.



H133 does not support the bus width of 8 bits.

**Figure 6-4 Data Packet Format for SDR**

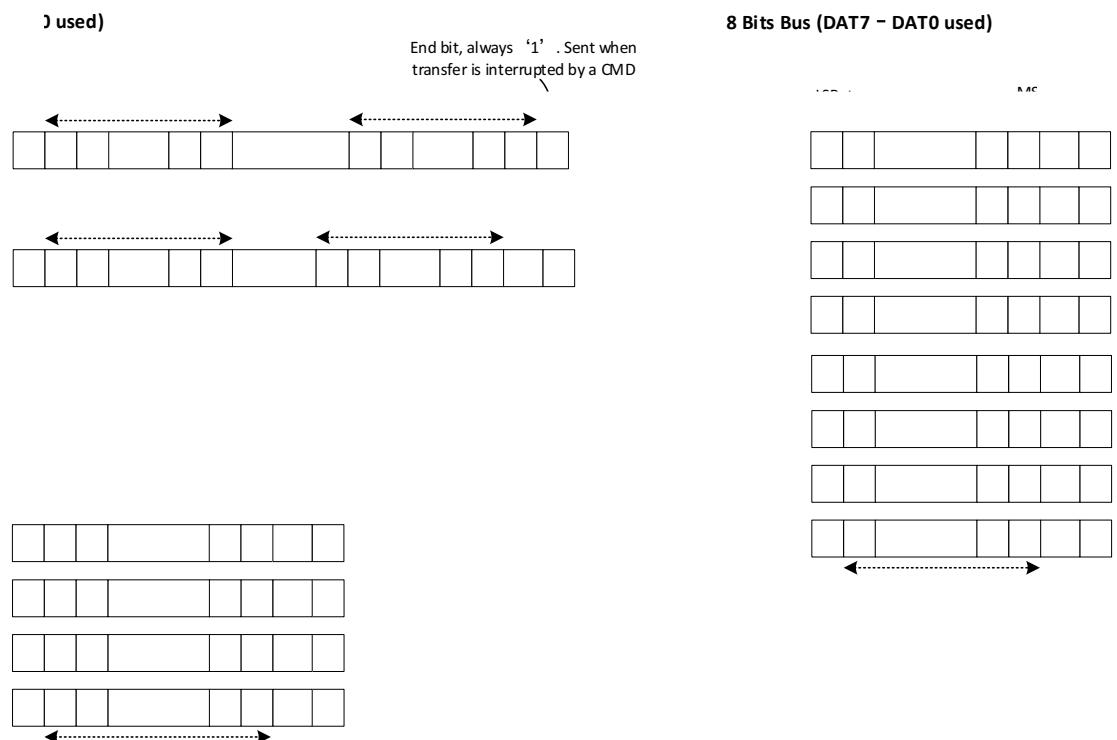
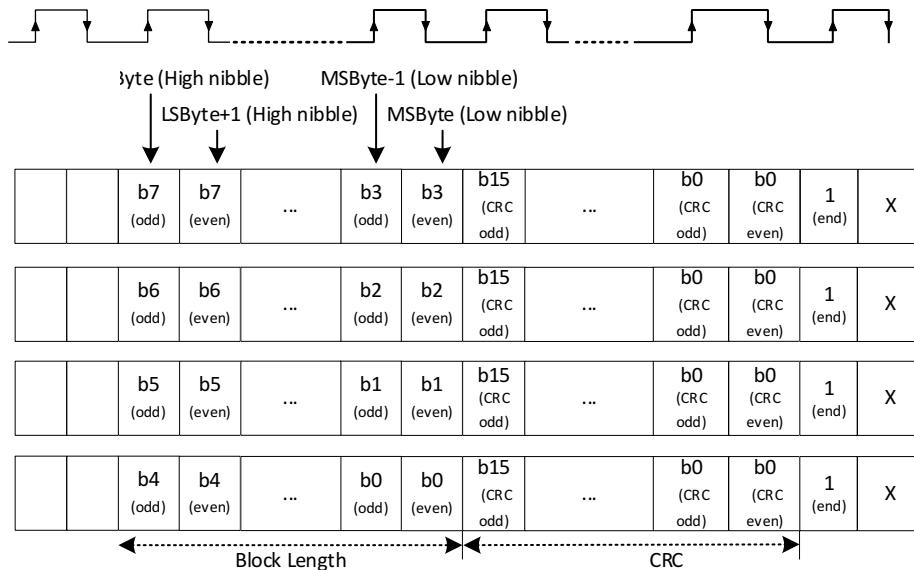
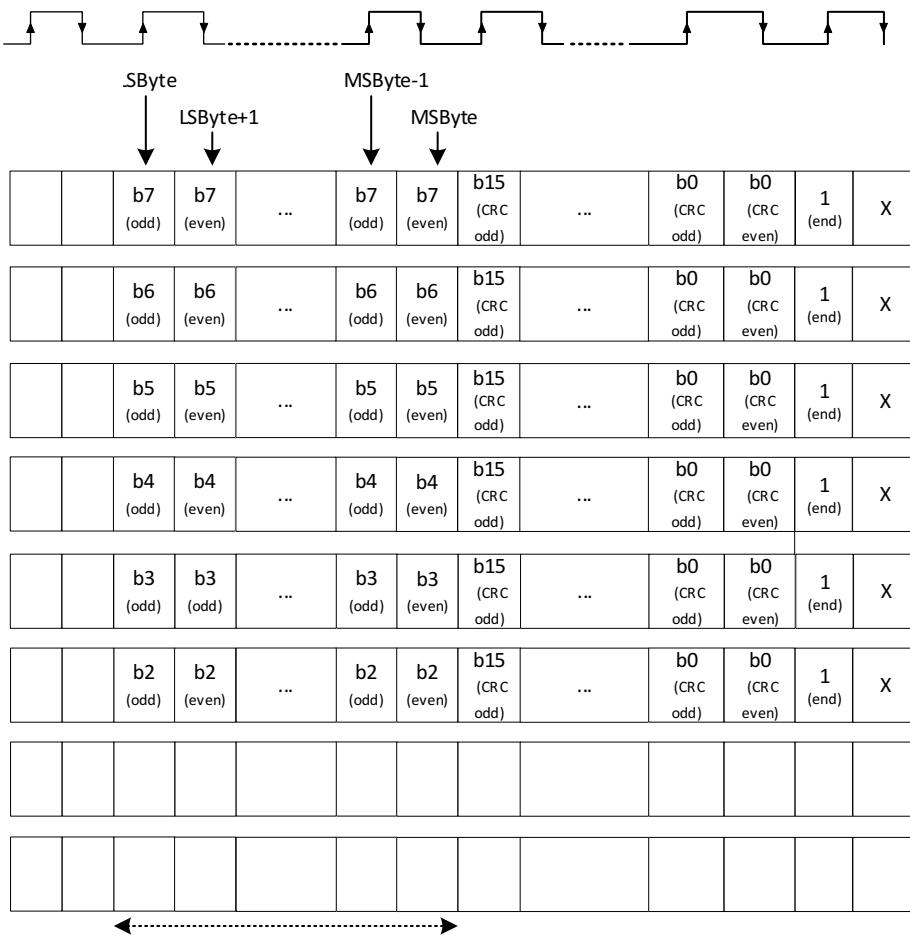


Figure 6-5 Data Packet Format for DDR

**T3 - DAT0 used)****T7 - DAT0 used)**

 NOTE

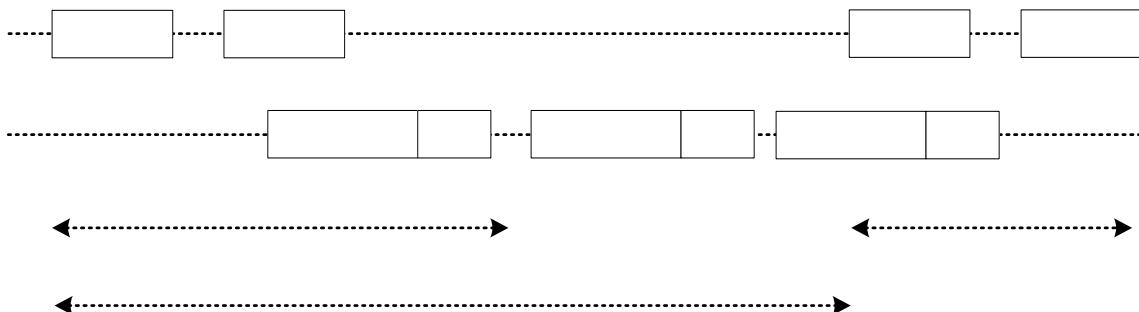
- Bytes data are not interleaved except CRCs.
- Start and end bits are only valid on the rising edge ("X" indicates "undefined").
- H133 does not support the bus width of 8 bits.

#### 6.2.3.6 Data Transfer

Data transferred to or from the SD/eMMC card are done in blocks. Single and multiple block operations are widely used during data transfer.

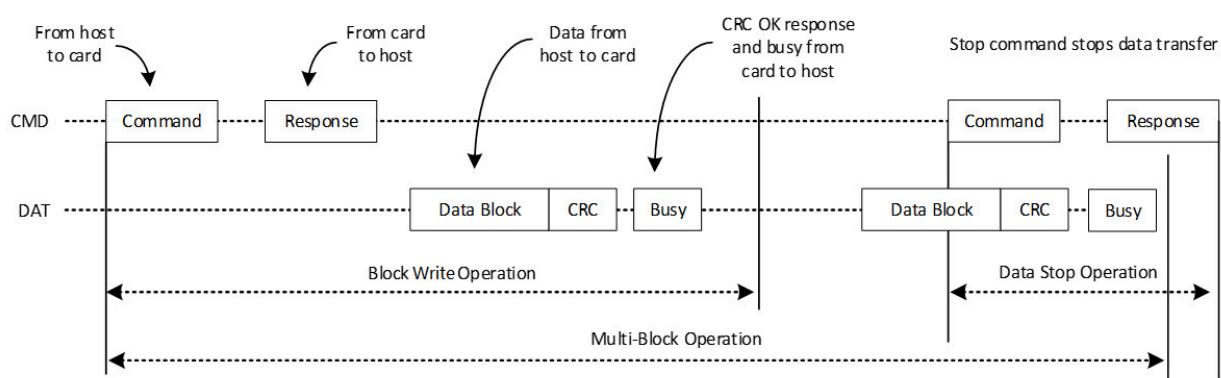
The following figure shows the single-block and multi-block read operation.

**Figure 6-6 Single-Block and Multi-Block Read Operation**



The following figure shows the single-block and multi-block write operation.

**Figure 6-7 Single-Block and Multi-Block Write Operation**



### 6.2.3.7 Bus Speed Modes

The following table shows the bus speed modes supported by SD 3.0.

**Table 6-5 Speed Modes Supported by SD 3.0**

Mode	I/O Voltage	Bus Width	Frequency	Throughput
SDR104	1.8 V	1, 4 bits	0 to 208 MHz	0 to 104 MB/s
SDR50	1.8 V	1, 4 bits	0 to 100MHz	0 to 50 MB/s
DDR50	1.8 V	1, 4 bits	0 to 50 MHz	0 to 50 MB/s
SDR25	1.8 V	1, 4 bits	0 to 50 MHz	0 to 25 MB/s
SDR12	1.8 V	1, 4 bits	0 to 25 MHz	0 to 12 MB/s
High Speed Mode (HS)	3.3 V	1, 4 bits	0 to 50 MHz	0 to 25 MB/s
Default Speed Mode (DS)	3.3 V	1, 4 bits	0 to 25 MHz	0 to 12 MB/s

The following table shows the bus speed modes supported by eMMC 5.0.

**Table 6-6 Speed Modes Supported by eMMC 5.0**

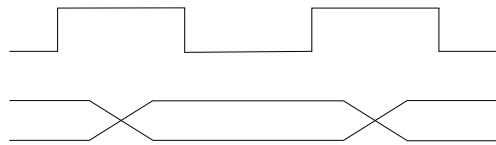
Mode	Data Rate	I/O Voltage	Bus Width	Frequency	Throughput
Backwards Compatibility with legacy MMC card	Single	3.3 V/1.8 V	1, 4 bits	0 to 26 MHz	0 to 26 MB/s
High Speed SDR	Single	3.3 V/1.8 V	1, 4 bits	0 to 52 MHz	0 to 52 MB/s
High Speed DDR	Dual	3.3 V/1.8 V	4 bits	0 to 52 MHz	0 to 104 MB/s
HS200	Single	1.8 V	4 bits	0 to 200 MHz	0 to 200 MB/s

### 6.2.3.8 Phase Offset of the Command and Data

You can configure the phase offset of the command and data by the [SMHC\\_DRV\\_DL](#) register.

#### SDR Mode

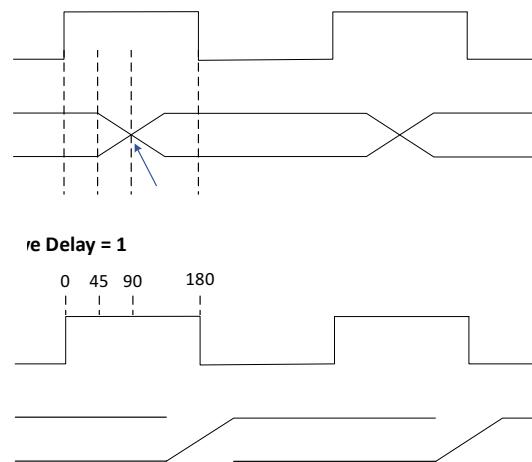
The following figure shows the phase offset of SDR command and data.

**Figure 6-8 Phase Offset of Command and Data in SDR Mode****Table 6-7 Phase Offset of Command and Data in SDR Mode**

Drive Delay	Command	Data
0	The command is updated in 90° clock position	The data is updated in 90° clock position
1	The command is updated in 180° clock position	The data is updated in 180° clock position

**DDR4 Mode**

The following figure shows the phase offset of DDR4 command and data.

**Figure 6-9 Phase Offset of Command and Data in DDR4 Mode (SMHC\_NTSR[31] = 0)****Figure 6-10 Phase Offset of Command and Data in DDR4 Mode**

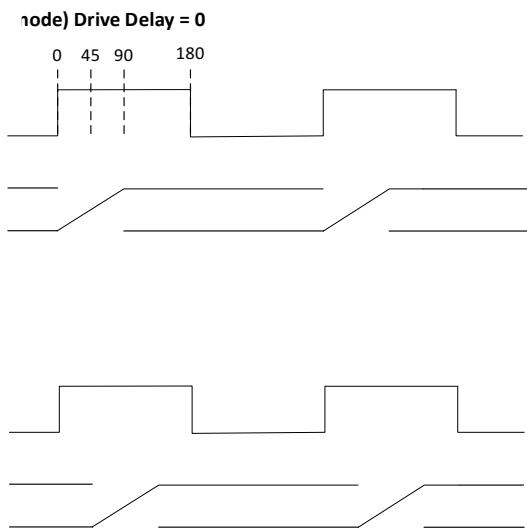
Drive Delay	Command	Data
0	The command is updated in 90° clock position	The data is updated in 90° clock position

Drive Delay	Command	Data
1	The command is updated in 180° clock position	The data is updated in 0° or 180° clock position

### DDR4 (2x) Mode

The following figure shows the phase offset of DDR4 (2x mode) command and data.

**Figure 6-11 Phase Offset of Command and Data in DDR4 (2x Mode) ([SMHC\\_NTSR\[31\]](#) = 1)**



**Table 6-8 Phase Offset of Command and Data in DDR4 (2x) Mode**

Drive Delay	Command	Data
0	The command is updated in 45° clock position	The data is updated in 45° clock position
1	The command is updated in 90° clock position	The data is updated in 90° clock position

#### 6.2.3.9 Internal DMA Controller Description

The SMHC has an internal DMA controller (IDMAC) to transfer data between the host memory and SMHC port. With a descriptor, the IDMAC can efficiently move data from the source to destination by automatically loading the next DMA transfer arguments, which needs less CPU intervention. Before transferring data in the IDMAC, the host driver should construct a descriptor list, configure arguments of every DMA transfer, and then launch the descriptor and start the DMA.

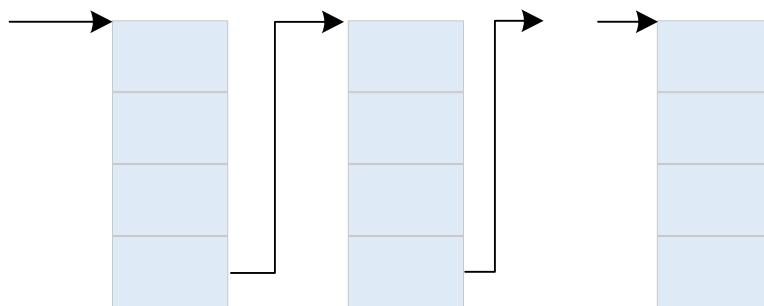
The IDMAC has an interrupt controller. When enabled, it generates an interrupt to the HOST CPU in situations such as data transmission is completed or some errors are occurred.

### IDMAC Descriptor Structure

The IDMAC uses a descriptor with a chain structure, and each descriptor points to a unique buffer and the next descriptor.

The following figure shows the internal formats of a descriptor.

**Figure 6-12 IDMAC Descriptor Structure Diagram**



This figure illustrates the internal formats of a descriptor. The descriptor address must be aligned to the bus width used for 32-bit buses. Each descriptor contains 16 bytes of control and status information.

DES0 corresponds to the [31:0] bits, DES1 corresponds to the [63:32] bits, DES2 corresponds to the [95:64] bits, and DES3 corresponds to the [127:96] bits in a descriptor.

The following table shows the bit definition of DES0.

**Table 6-9 DES0 Definition**

Bits	Name	Description
31	HOLD	DES_own_flag When set to 1, this bit indicates that the descriptor is owned by the IDMAC. When this bit is reset, it indicates that the descriptor is owned by the host. This bit is cleared when the transfer is over.
30	ERROR	ERR_flag When some errors occurred in transfer, this bit will be set to 1.
29:5	/	/
4	Chain Flag	CHAIN_mod When set to 1, this bit indicates that the second address in the descriptor is the next descriptor address. It must be set to 1.
3	First DES Flag	FIRST_flag When set to 1, this bit indicates that this descriptor contains the first buffer of data. It must be set to 1 in the first DES.

Bits	Name	Description
2	Last DES Flag	LAST_FLAG When set to 1, this bit indicates that the buffers this descriptor points to are the last data buffer.
1	Disable Interrupt on completion	CUR_TXRX_OVER_INT_DIS When set to 1, this bit will prevent the setting of the TX/RX interrupt bit of the IDMAC status register for data that ends in the buffer the descriptor points to.
0	/	/

The following table shows the bit definition of DES1.

**Table 6-10 DES1 Definition**

Bits	Name	Description
31:13	/	/
12:0	Buffer size	BUFF_SIZE The bits indicate the data buffer byte size, which must be a multiple of 4 bytes. If this field is 0, the DMA ignores this buffer and proceeds to the next descriptor.

The following table shows the bit definition of DES2.

**Table 6-11 DES2 Definition**

Bits	Name	Descriptor
31:0	Buffer pointer address	BUFF_ADDR The bits indicate the physical address of the data buffer. It is a word address.

The following table shows the bit definition of DES3.

**Table 6-12 DES3 Definition**

Bits	Name	Descriptor
31:0	Next descriptor address	NEXT_DESP_ADDR The bits indicate the pointer to the physical memory where the next descriptor is present. It is a word address.

### 6.2.3.10 Calibrating the Delay Chain

There are two delay chains in SMHC: data strobe delay chain and sample delay chain.

**Data strobe delay chain:** used to generate delay to make proper timing between data strobe and data signals.

**Sample delay chain:** used to generate delay to make proper timing between the internal card clock signal and data signals.

Each delay chain is made up with 64 delay cells. The delay time of one delay cell can be estimated through delay chain calibration.

Follow the steps below to calibrate the delay chain:

**Step 1** Enable SMHC. In order to calibrate the delay chain by the operation registers in SMHC, the SMHC must be enabled through [SMHC Bus Gating Reset Register](#) and [SMHCx Clock Register](#).

**Step 2** Configure a proper clock for SMHC. The delay chain calibration is based on the clock for SMHC from Clock Control Unit (CCU). The delay chain calibration is an internal function in SMHC without devices. So it is unnecessary to open the clock signal for devices. The recommended clock frequency is 200 MHz.

**Step 3** Set proper initial delay value. Writing 0xA0 to **delay control register** enables **Delay Software Enable** (bit[7]) and sets initial delay value 0x20 to **Delay chain** (bit[5:0]). Then write 0x0 to **delay control register** to clear the value.

**Step 4** Write 0x8000 to **delay control register** to start calibrating the delay chain.

**Step 5** Wait until the flag (bit14 in **delay control register**) of calibration done is set. The number of delay cells is shown at bit[13:8] in **delay control register**. The delay time generated by these delay cells is equal to the cycle of the SMHC clock nearly. This value is the result of calibration.

**Step 6** Calculate the delay time of one delay cell according to the cycle of the SMHC clock and the result of calibration.

### 6.2.4 Programming Guidelines

#### 6.2.4.1 Initializing SMHC

Before data and commands are exchanged between a card and the SMHC, the SMHC needs to be initialized.

Follow the steps below to initialize the SMHC:

**Step 1** Configure the corresponding GPIO register as an SMHC by Port Controller module; reset clock by writing 1 to [SMHC\\_BGR\\_REG](#)[SMHCx\_RST], and open clock gating by writing 1 to

[SMHC\\_BGR\\_REG](#)[SMHCx\_GATING]; select clock sources and set the division factor by configuring the [SMHCx\\_CLK\\_REG](#)( $x = 0, 1$ ) register.

- Step 2** Configure [SMHC\\_CTRL](#) to reset the FIFO and controller, and enable the global interrupt; configure [SMHC\\_INTMASK](#) to 0xFFCE to enable normal interrupts and error abnormal interrupts, and then register the interrupt function.
- Step 3** Configure [SMHC\\_CLKDIV](#) to open clock for devices; configure [SMHC\\_CMD](#) as the change clock command (for example 0x80202000); send the update clock command to deliver clocks to devices.
- Step 4** Configure [SMHC\\_CMD](#) as a normal command. Configure [SMHC\\_CMDARG](#) to set command parameters. Configure [SMHC\\_CMD](#) to set parameters like whether to send the response, the response type, and the response length and then send the commands. According to the initialization process in the protocol, you can finish SMHC initialization by sending the corresponding command one by one.

#### 6.2.4.2 Writing a Single Data Block

To write a single data block, follow the steps below:

- Step 1** Write 0x1 to [SMHC\\_CTRL](#)[DMA\_RST] to reset the internal DMA controller; write 0x82 to [SMHC\\_IDMAC](#) to enable the IDMAC interrupt and configure AHB master burst transfers; configure [SMHC\\_IDIE](#) to enable the transfer interrupt, receive interrupt, and abnormal interrupt.
- Step 2** Configure [SMHC\\_FIFOTH](#) to determine the burst size and TX/RX trigger level. For example, if [SMHC\\_FIFOTH](#) is configured as 0x300F00F0, it indicates the burst size is 16, TX\_TL is 15, and RX\_TL is 240. Configure [SMHC\\_DLBA](#) to determine the start address of the DMA descriptor.
- Step 3** To write one block data to sector1, configure [SMHC\\_BYTCNT](#)[BYTE\_CNT] to 0x200 and configure the descriptor according to the data size; set the data sector address of CMD24 (Single Data Block Write) to 0x1, write 0x80002758 to [SMHC\\_CMD](#), and send CMD24 command to write data to the device.
- Step 4** Check whether [SMHC\\_RINTSTS](#)[CC] is 1. If yes, the command is sent successfully; otherwise, continue to wait until timeout, and then exit the process.
- Step 5** Check whether [SMHC\\_IDST](#)[TX\_INT] is 1. If yes, the data transfer for writing DMA is completed. Write 0x337 to [SMHC\\_IDST](#) to clear the interrupt flag. Otherwise, continue to wait until timeout, and then exit the process.
- Step 6** Check whether [SMHC\\_RINTSTS](#)[DTC] is 1. If yes, the data transfer and CMD24 writing operations are completed. Otherwise, abnormality exists. Read [SMHC\\_RINTSTS](#) and [SMHC\\_STATUS](#) to query the existing abnormality.
- Step 7** Send CMD13 command to query whether the device writing operation is completed and returns to the idle status. For example, device RCA is 0x1234, first set [SMHC\\_CMDARG](#) to 0x12340000, write 0x8000014D to [SMHC\\_CMD](#), go to step4 to ensure command transfer completed, and then check

whether the highest bit of [SMHC\\_RESP0](#) (CMD13 response) is 1. If yes, the device is in the idle status, and the next command can be sent. Otherwise, the device is in the busy status. Continue to send CMD13 to wait for the device to enter the idle status until timeout.

#### 6.2.4.3 Reading a Single Data Block

To read a single data block, follow the steps below:

- Step 1** Write 0x1 to [SMHC\\_CTRL](#)[DMA\_RST] to reset the internal DMA controller; write 0x82 to [SMHC\\_IDMAC](#) to enable the IDMAC interrupt and configure AHB master burst transfers; configure [SMHC\\_IDIE](#) to enable the transfer interrupt, receive interrupt, and abnormal interrupt.
- Step 2** Configure [SMHC\\_FIFOTH](#) to determine the burst size and TX/RX trigger level. For example, if [SMHC\\_FIFOTH](#) is configured as 0x300F00F0, it indicates the burst size is 16, TX\_TL is 15, and RX\_TL is 240. Configure [SMHC\\_DLBA](#) to determine the start address of the DMA descriptor.
- Step 3** To read one block data from sector1, configure [SMHC\\_BYTCNT](#)[BYTE\_CNT] to 0x200 and configure the descriptor according to the data size; set the data sector address of CMD17 command (Single Data Block Read) to 0x1, write 0x80002351 to [SMHC\\_CMD](#), and send CMD17 command to read data from the device to DRAM/SRAM.
- Step 4** Check whether [SMHC\\_RINTSTS](#)[CC] is 1. If yes, the command is sent successfully; otherwise, continue to wait until timeout, and then exit the process.
- Step 5** Check whether [SMHC\\_IDST](#)[RX\_INT] is 1. If yes, the data transfer for writing DMA is completed. Write 0x337 to [SMHC\\_IDST](#) to clear the interrupt flag. Otherwise, continue to wait until timeout, and then exit the process.
- Step 6** Check whether [SMHC\\_RINTSTS](#)[DTC] is 1. If yes, data transfer and CMD17 reading operation are completed. Otherwise, abnormality exists. Read [SMHC\\_RINTSTS](#) and [SMHC\\_STATUS](#) to query the existing abnormality.

#### 6.2.4.4 Writing Open-Ended Multiple Data Blocks (CMD25 + Auto CMD12)

To write open-ended multiple data blocks, follow the steps below:

- Step 1** Write 0x1 to [SMHC\\_CTRL](#)[DMA\_RST] to reset the internal DMA controller; write 0x82 to [SMHC\\_IDMAC](#) to enable the IDMAC interrupt and configure AHB master burst transfers; configure [SMHC\\_IDIE](#) to enable the transfer interrupt, receive interrupt, and abnormal interrupt.

- Step 2** Configure [SMHC\\_FIFOTH](#) to determine the burst size and TX/RX trigger level. For example, if [SMHC\\_FIFOTH](#) is configured as 0x300F00F0, it indicates the burst size is 16, TX\_TL is 15, and RX\_TL is 240. Configure [SMHC\\_DLBA](#) to determine the start address of the DMA descriptor.
- Step 3** To write three blocks of data to sectors begin with sector0, configure [SMHC\\_BYTCNT](#)[BYTE\_CNT] to 0x600 and configure the descriptor according to the data size; set the data sector address of CMD25 command (Multiple Data Blocks Write) to 0x0, write 0x80003759 to [SMHC\\_CMD](#), and send CMD25 command to read data from the device to DRAM/SRAM.
- Step 4** Check whether [SMHC\\_RINTSTS](#)[CC] is 1. If yes, the command is sent successfully; otherwise, continue to wait until timeout, and then exit the process.
- Step 5** Check whether [SMHC\\_IDST](#)[RX\_INT] is 1. If yes, the data transfer for writing DMA is completed. Write 0x337 to [SMHC\\_IDST](#) to clear the interrupt flag. Otherwise, continue to wait until timeout, and then exit the process.
- Step 6** Check whether [SMHC\\_RINTSTS](#)[ACD] and [SMHC\\_RINTSTS](#)[DTC] are both 1. If yes, the data transfer, CMD12 transfer, and CMD25 writing operations are completed. Otherwise, abnormality exists. Read [SMHC\\_RINTSTS](#) and [SMHC\\_STATUS](#) to query the existing abnormality.
- Step 7** Send CMD13 command to query whether the device writing operation is completed and returns to the idle status. For example, device RCA is 0x1234, first set [SMHC\\_CMDARG](#) to 0x12340000, write 0x8000014D to [SMHC\\_CMD](#), go to step4 to ensure command transfer completed, and then check whether the highest bit of [SMHC\\_RESP0](#) (CMD13 response) is 1. If yes, the device is in the idle status, and the next command can be sent. Otherwise, the device is in the busy status. Continue to send CMD13 to wait for the device to enter the idle status until timeout.

#### 6.2.4.5 Reading Open-Ended Multiple Data Blocks (CMD18 + Auto CMD12)

To read open-ended multiple data blocks, follow the steps below:

- Step 1** Write 0x1 to [SMHC\\_CTRL](#)[DMA\_RST] to reset the internal DMA controller; write 0x82 to [SMHC\\_IDMAC](#) to enable the IDMAC interrupt and configure AHB master burst transfers; configure [SMHC\\_IDIE](#) to enable the transfer interrupt, receive interrupt, and abnormal interrupt.
- Step 2** Configure [SMHC\\_FIFOTH](#) to determine the burst size and TX/RX trigger level. For example, if [SMHC\\_FIFOTH](#) is configured as 0x300F00F0, it indicates the burst size is 16, TX\_TL is 15, and RX\_TL is 240. Configure [SMHC\\_DLBA](#) to determine the start address of the DMA descriptor.
- Step 3** To read three blocks of data from sectors begin with sector0, configure [SMHC\\_BYTCNT](#)[BYTE\_CNT] to 0x600 and configure the descriptor according to the data size; set the data sector address of CMD18 command (Multiple Data Blocks Read) to 0x0, write 0x80003352 to [SMHC\\_CMD](#), and send CMD18

command to read data to the device. When the data transfer is completed, CMD12 will be sent automatically.

- Step 4** Check whether [SMHC\\_RINTSTS](#)[CC] is 1. If yes, the command is sent successfully; otherwise, continue to wait until timeout, and then exit the process.
- Step 5** Check whether [SMHC\\_IDST](#)[RX\_INT] is 1. If yes, the data transfer for writing DMA is completed. Write 0x337 to [SMHC\\_IDST](#) to clear the interrupt flag. Otherwise, continue to wait until timeout, and then exit the process.
- Step 6** Check whether [SMHC\\_RINTSTS](#)[ACD] and [SMHC\\_RINTSTS](#)[DTC] are both 1. If yes, data transfer, CMD12 transfer, and CMD18 reading operation are completed. Otherwise, abnormality exists. Read [SMHC\\_RINTSTS](#) and [SMHC\\_STATUS](#) to query the existing abnormality.

#### 6.2.4.6 Writing Pre-Defined Multiple Data Blocks (CMD23 + CMD25)

To write pre-defined multiple data blocks, follow the steps below:

- Step 1** Write 0x1 to [SMHC\\_CTRL](#)[DMA\_RST] to reset the internal DMA controller; write 0x82 to [SMHC\\_IDMAC](#) to enable the IDMAC interrupt and configure AHB master burst transfers; configure [SMHC\\_IDIE](#) to enable the transfer interrupt, receive interrupt, and abnormal interrupt.
- Step 2** Configure [SMHC\\_FIFOTH](#) to determine the burst size and TX/RX trigger level. For example, if [SMHC\\_FIFOTH](#) is configured as 0x300F00F0, it indicates the burst size is 16, TX\_TL is 15, and RX\_TL is 240. Configure SMHC\_DLBA to determine the start address of the DMA descriptor.
- Step 3** To write three blocks of data, configure [SMHC\\_CMDARG](#) to 0x3 to specify the number of data blocks as three. Then write 0x80000157 to [SMHC\\_CMD](#) to send the CMD23 command. Check whether [SMHC\\_RINTSTS](#)[CC] is 1. If yes, the command is sent successful; otherwise, continue to wait until timeout, and then exit the process.
- Step 4** Configure [SMHC\\_BYTCNT](#)[BYTE\_CNT] to 0x600 and configure the descriptor according to the data size; set the data sector address of CMD25 command (Multiple Data Blocks Write) to 0x0, write 0x80002759 to [SMHC\\_CMD](#), and send CMD25 command to write data to the device.
- Step 5** Check whether [SMHC\\_RINTSTS](#)[CC] is 1. If yes, the command is sent successful; otherwise, continue to wait until timeout, and then exit the process.
- Step 6** Check whether [SMHC\\_IDST](#)[TX\_INT] is 1. If yes, the data transfer for writing DMA is completed. Write 0x337 to [SMHC\\_IDST](#) to clear the interrupt flag. Otherwise, continue to wait until timeout, and then exit the process.
- Step 7** Check whether [SMHC\\_RINTSTS](#)[DTC] is 1. If yes, the data transfer and CMD25 writing operations are completed. Otherwise, abnormality exists. Read [SMHC\\_RINTSTS](#) and [SMHC\\_STATUS](#) to query the existing abnormality.

**Step 8** Send CMD13 command to query whether the device writing operation is completed and returns to the idle status. For example, device RCA is 0x1234, first set [SMHC\\_CMDARG](#) to 0x12340000, write 0x80000014D to [SMHC\\_CMD](#), go to step4 to ensure command transfer completed, and then check whether the highest bit of [SMHC\\_RESP0](#) (CMD13 response) is 1. If yes, the device is in the idle status, and the next command can be sent. Otherwise, the device is in the busy status. Continue to send CMD13 to wait for the device to enter the idle status until timeout.

#### 6.2.4.7 Reading Pre-Defined Multiple Data Blocks (CMD23 + CMD18)

To read pre-defined multiple data blocks, follow the steps below:

- Step 1** Write 0x1 to [SMHC\\_CTRL](#)[DMA\_RST] to reset the internal DMA controller; write 0x82 to [SMHC\\_IDMAC](#) to enable the IDMAC interrupt and configure AHB master burst transfers; configure [SMHC\\_IDIE](#) to enable the transfer interrupt, receive interrupt, and abnormal interrupt.
- Step 2** Configure [SMHC\\_FIFOTH](#) to determine the burst size and TX/RX trigger level. For example, if [SMHC\\_FIFOTH](#) is configured as 0x300F00F0, it indicates the burst size is 16, TX\_TL is 15, and RX\_TL is 240. Configure [SMHC\\_DLBA](#) to determine the start address of the DMA descriptor.
- Step 3** To read three blocks of data, configure [SMHC\\_CMDARG](#) to 0x3 to specify the number of data blocks as three. Then write 0x80000157 to [SMHC\\_CMD](#) to send the CMD23 command. Check whether [SMHC\\_RINTSTS](#)[CC] is 1. If yes, the command is sent successful; otherwise, continue to wait until timeout, and then exit the process.
- Step 4** Configure [SMHC\\_BYTCNT](#)[BYTE\_CNT] to 0x600 and configure the descriptor according to the data size; set the data sector address of CMD18 (Multiple Data Blocks Read) to 0x0, write 0x80002352 to [SMHC\\_CMD](#), and send CMD18 command to read data from device to DRAM/SRAM.
- Step 5** Check whether [SMHC\\_RINTSTS](#)[CC] is 1. If yes, the command is sent successful; otherwise, continue to wait until timeout, and then exit the process.
- Step 6** Check whether [SMHC\\_IDST](#)[TX\_INT] is 1. If yes, the data transfer for writing DMA is completed. Write 0x337 to [SMHC\\_IDST](#) to clear the interrupt flag. Otherwise, continue to wait until timeout, and then exit the process.
- Step 7** Check whether [SMHC\\_RINTSTS](#)[DTC] is 1. If yes, the data transfer and CMD18 writing operations are completed. Otherwise, abnormality exists. Read [SMHC\\_RINTSTS](#) and [SMHC\\_STATUS](#) to query the existing abnormality.

### 6.2.5 Register List

Module Name	Base Address
SMHC0	0x04020000
SMHC1	0x04021000
SMHC2	0x04022000

Register Name	Offset	Description
SMHC_CTRL	0x0000	Control Register
SMHC_CLKDIV	0x0004	Clock Control Register
SMHC_TMOUT	0x0008	Time Out Register
SMHC_CTYPE	0x000C	Bus Width Register
SMHC_BLKSIZ	0x0010	Block Size Register
SMHC_BYTCNT	0x0014	Byte Count Register
SMHC_CMD	0x0018	Command Register
SMHC_CMDARG	0x001C	Command Argument Register
SMHC_RESP0	0x0020	Response 0 Register
SMHC_RESP1	0x0024	Response 1 Register
SMHC_RESP2	0x0028	Response 2 Register
SMHC_RESP3	0x002C	Response 3 Register
SMHC_INTMASK	0x0030	Interrupt Mask Register
SMHC_MINTSTS	0x0034	Masked Interrupt Status Register
SMHC_RINTSTS	0x0038	Raw Interrupt Status Register
SMHC_STATUS	0x003C	Status Register
SMHC_FIFOTH	0x0040	FIFO Water Level Register
SMHC_FUNS	0x0044	FIFO Function Select Register
SMHC_TCBCNT	0x0048	Transferred Byte Count between Controller and Card
SMHC_TBBCNT	0x004C	Transferred Byte Count between Host Memory and Internal FIFO
SMHC_DBGC	0x0050	Current Debug Control Register
SMHC_CSDC	0x0054	CRC Status Detect Control Registers
SMHC_A12A	0x0058	Auto Command 12 Argument Register
SMHC_NTSR	0x005C	SD New Timing Set Register
SMHC_HWRST	0x0078	Hardware Reset Register
SMHC_IDMAC	0x0080	IDMAC Control Register

Register Name	Offset	Description
SMHC_DLBA	0x0084	Descriptor List Base Address Register
SMHC_IDST	0x0088	IDMAC Status Register
SMHC_IDIE	0x008C	IDMAC Interrupt Enable Register
SMHC_THLD	0x0100	Card Threshold Control Register
SMHC_SFC	0x0104	Sample FIFO Control Register
SMHC_A23A	0x0108	Auto Command 23 Argument Register
EMMC_DDR_SBIT_DET	0x010C	eMMC4.5 DDR Start Bit Detection Control Register
SMHC_EXT_CMD	0x0138	Extended Command Register
SMHC_EXT RESP	0x013C	Extended Response Register
SMHC_DRV_DL	0x0140	Drive Delay Control Register
SMHC_SMAP_DL	0x0144	Sample Delay Control Register
SMHC_DS_DL	0x0148	Data Strobe Delay Control Register
SMHC_HS400_DL	0x014C	HS400 Delay Control Register
SMHC_FIFO	0x0200	Read/Write FIFO

## 6.2.6 Register Description

### 6.2.6.1 0x0000 SMHC Global Control Register (Default Value: 0x0000\_0100)

Offset: 0x0000			Register Name: SMHC_CTRL
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	FIFO_AC_MOD FIFO Access Mode 0: DMA bus 1: AHB bus
30:13	/	/	/
12	R/W	0x0	TIME_UNIT_CMD Time unit for command line The time unit is used to calculate the command line time out value defined in RTO_LMT. 0: 1 card clock period 1: 256 card clock period

Offset: 0x0000			Register Name: SMHC_CTRL
Bit	Read/Write	Default/Hex	Description
11	R/W	0x0	<p>TIME_UNIT_DAT Time unit for data line Time unit is used to calculate the data line time out value defined in DTO_LMT. 0: 1 card clock period 1: 256 card clock period</p>
10	R/W	0x0	<p>DDR_MOD_SEL DDR Mode Select Although the HS400 speed mode of eMMC is 8-bit DDR, this field should be cleared when HS400_MD_EN is set. 0: SDR mode 1: DDR mode</p>
9	/	/	/
8	R/W	0x1	<p>CD_DBC_ENB Card Detect (Data[3] status) Debounce Enable 0: Disable de-bounce 1: Enable de-bounce</p>
7:6	/	/	/
5	R/W	0x0	<p>DMA_ENB DMA Global Enable 0: Disable DMA to transfer data via AHB bus 1: Enable DMA to transfer data</p>
4	R/W	0x0	<p>INT_ENB Global Interrupt Enable 0: Disable interrupts 1: Enable interrupts</p>
3	/	/	/
2	R/W	0x0	<p>DMA_RST DMA Reset</p>
1	R/W	0x0	<p>FIFO_RST FIFO Reset 0: No effect 1: Reset the FIFO This bit is auto-cleared after the completion of the reset operation.</p>

Offset: 0x0000			Register Name: SMHC_CTRL
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	<p>SOFT_RST Software Reset 0: No effect 1: Reset SD/MMC controller This bit is auto-cleared after the completion of reset operation.</p>

#### 6.2.6.2 0x0004 SMHC Clock Control Register (Default Value: 0x0000\_0000)

Offset: 0x0004			Register Name: SMHC_CLKDIV
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<p>MASK_DATA0 0: Do not mask data0 when update clock 1: Mask data0 when update clock</p>
30:18	/	/	/
17	R/W	0x0	<p>CCLK_CTRL Card Clock Output Control 0: Card clock is always on 1: Turn off card clock when FSM is in the IDLE state</p>
16	R/W	0x0	<p>CCLK_ENB Card Clock Enable 0: Card Clock is off 1: Card Clock is on</p>
15:8	/	/	/
7:0	R/W	0x0	<p>CCLK_DIV Card Clock Divider n: Source clock is divided by 2*n. (n = 0 to 255) When HS400_MD_EN is set, this field must be cleared.</p>

#### 6.2.6.3 0x0008 SMHC Timeout Register (Default Value: 0xFFFF\_FF40)

Offset: 0x0008			Register Name: SMHC_TMOUT
Bit	Read/Write	Default/Hex	Description
31:8	R/W	0xffffffff	<p>DTO_LMT Data Timeout Limit This field can set the time that the host waits for the data from the device. Ensure to communicate with the device, this field must be set to the maximum that is greater than the time <math>N_{AC}</math>. About the <math>N_{AC}</math>, the explanation is as follows: When the Host read data, data transmission from the Device starts after the access time delay <math>N_{AC}</math> beginning from the end bit of the read command (ACMD51, CMD8, CMD17, and CMD18). When the host reads multiple block (CMD18), a next block's data transmission from the Device starts after the access time delay <math>N_{AC}</math> beginning from the end bit of the previous block. When the host writes data, the value is no effect.</p>
7:0	R/W	0x40	RTO_LMT Response Timeout Limit

#### 6.2.6.4 0x000C SMHC Bus Width Register (Default Value: 0x0000\_0000)

Offset: 0x000C			Register Name: SMHC_CTYPE
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1:0	R/W	0x0	CARD_WID Card Width 00: 1-bit width 01: 4-bit width 1x: 8-bit width

#### 6.2.6.5 0x0010 SMHC Block Size Register (Default Value: 0x0000\_0200)

Offset: 0x0010			Register Name: SMHC_BLKSIZ
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x200	BLK_SZ Block Size

#### 6.2.6.6 0x0014 SMHC Byte Count Register (Default Value: 0x0000\_0200)

Offset: 0x0014			Register Name: SMHC_BYTCNT
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x200	BYTE_CNT Byte counter The number of bytes to be transferred. It must be integer multiple of Block Size (BLK_SZ) for block transfers.

#### 6.2.6.7 0x0018 SMHC Command Register (Default Value: 0x0000\_0000)

Offset: 0x0018			Register Name: SMHC_CMD
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	CMD_LOAD Start Command This bit is automatically cleared when the current command is sent. If there is no response error occurs, a command complete interrupt bit (CMD_OVER) will be set in the interrupt register. Do not write any other commands until this bit is cleared.
30:29	/	/	/
28	R/W	0x0	VOL_SW Voltage Switch 0: Normal command 1: Voltage switch command, set for CMD11 only.
27	R/W	0x0	BOOT_ABST Boot Abort Setting this bit will terminate the boot operation.

Offset: 0x0018			Register Name: SMHC_CMD
Bit	Read/Write	Default/Hex	Description
26	R/W	0x0	<p>EXP_BOOT_ACK Expect Boot Acknowledge When the software sets this bit along in mandatory boot operation, the controller expects a boot acknowledge start pattern of 0-1-0 from the selected card.</p>
25:24	R/W	0x0	<p>BOOT_MOD Boot Mode 00: Normal command 01: Mandatory Boot operation 10: Alternate Boot operation 11: Reserved</p>
23:22	/	/	/
21	R/W	0x0	<p>PRG_CLK Change Clock 0: Normal command 1: Change Card Clock When this bit is set, the controller will change the clock domain and clock output. No commands will be sent.</p>
20:16	/	/	/
15	R/W	0x0	<p>SEND_INIT_SEQ Send Initialization 0: Normal command sending 1: Send initialization sequence before sending this command.</p>
14	R/W	0x0	<p>STOP_ABT_CMD Stop Abort Command 0: Normal command sending 1: Send <i>Stop</i> or <i>Abort</i> command to stop the current data transfer in progress. (CMD12, CMD52 for writing “I/O Abort” in SDIO CCCR)</p>
13	R/W	0x0	<p>WAIT_PRE_OVER Wait for Data Transfer Over 0: Send command at once, does not care about data transferring. 1: Wait for data transfer completion before sending the current command.</p>

Offset: 0x0018			Register Name: SMHC_CMD
Bit	Read/Write	Default/Hex	Description
12	R/W	0x0	<p>STOP_CMD_FLAG</p> <p>Send Stop CMD Automatically (CMD12)</p> <p>0: Do not send stop command at the end of the data transfer.</p> <p>1: Send stop command automatically at the end of the data transfer.</p> <p>If set, the SMHC_RESP1 will record the response of auto CMD12.</p>
11	R/W	0x0	<p>TRANS_MODE</p> <p>Transfer Mode</p> <p>0: Block data transfer command</p> <p>1: Stream data transfer command</p>
10	R/W	0x0	<p>TRANS_DIR</p> <p>Transfer Direction</p> <p>0: Read operation</p> <p>1: Write operation</p>
9	R/W	0x0	<p>DATA_TRANS</p> <p>Data Transfer</p> <p>0: Without data transfer</p> <p>1: With data transfer</p>
8	R/W	0x0	<p>CHK_RESP_CRC</p> <p>Check Response CRC</p> <p>0: Do not check response CRC</p> <p>1: Check response CRC</p>
7	R/W	0x0	<p>LONG_RESP</p> <p>Response Type</p> <p>0: Short Response (48 bits)</p> <p>1: Long Response (136 bits)</p>
6	R/W	0x0	<p>RESP_RCV</p> <p>Response Receive</p> <p>0: Command without response</p> <p>1: Command with response</p>
5:0	R/W	0x0	<p>CMD_IDX</p> <p>CMD Index</p> <p>Command index value</p>

**6.2.6.8 0x001C SMHC Command Argument Register (Default Value: 0x0000\_0000)**

Offset: 0x001C			Register Name: SMHC_CMDARG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	CMD_ARG Command argument

**6.2.6.9 0x0020 SMHC Response 0 Register (Default Value: 0x0000\_0000)**

Offset: 0x0020			Register Name: SMHC_RESP0
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	CMD_RESP0 Response 0 Bit[31:0] of response

**6.2.6.10 0x0024 SMHC Response 1 Register (Default Value: 0x0000\_0000)**

Offset: 0x0024			Register Name: SMHC_RESP1
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	CMD_RESP1 Response 1 Bit[63:31] of response

**6.2.6.11 0x0028 SMHC Response 2 Register (Default Value: 0x0000\_0000)**

Offset: 0x0028			Register Name: SMHC_RESP2
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	CMD_RESP2 Response 2 Bit[95:64] of response

#### 6.2.6.12 0x002C SMHC Response 3 Register (Default Value: 0x0000\_0000)

Offset: 0x002C			Register Name: SMHC_RESP3
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	CMD_RESP3 Response 3 Bit[127:96] of response

#### 6.2.6.13 0x0030 SMHC Interrupt Mask Register (Default Value: 0x0000\_0000)

Offset: 0x0030			Register Name: SMHC_INTMASK
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	CARD_REMOVAL_INT_EN Card Removed Interrupt Enable
30	R/W	0x0	CARD_INSERT_INT_EN Card Inserted Interrupt Enable
29:17	/	/	/
16	R/W	0x0	SDIO_INT_EN SDIO Interrupt Enable
15	R/W	0x0	DEE_INT_EN Data End-bit Error Interrupt Enable
14	R/W	0x0	ACD_INT_EN Auto Command Done Interrupt Enable
13	R/W	0x0	DSE_BC_INT_EN Data Start Error Interrupt Enable
12	R/W	0x0	CB_IW_INT_EN Command Busy and Illegal Write Interrupt Enable
11	R/W	0x0	FU_FO_INT_EN FIFO Underrun/Overflow Interrupt Enable
10	R/W	0x0	DSTO_VSD_INT_EN Data Starvation Timeout/V1.8 Switch Done Interrupt Enable
9	R/W	0x0	DTO_BDS_INT_EN Data Timeout/Boot Data Start Interrupt Enable
8	R/W	0x0	RTO_BACK_INT_EN Response Timeout/Boot ACK Received Interrupt Enable

Offset: 0x0030			Register Name: SMHC_INTMASK
Bit	Read/Write	Default/Hex	Description
7	R/W	0x0	DCE_INT_EN Data CRC Error Interrupt Enable
6	R/W	0x0	RCE_INT_EN Response CRC Error Interrupt Enable
5	R/W	0x0	DRR_INT_EN Data Receive Request Interrupt Enable
4	R/W	0x0	DTR_INT_EN Data Transmit Request Interrupt Enable
3	R/W	0x0	DTC_INT_EN Data Transfer Complete Interrupt Enable
2	R/W	0x0	CC_INT_EN Command Complete Interrupt Enable
1	R/W	0x0	RE_INT_EN Response Error Interrupt Enable
0	/	/	/

#### 6.2.6.14 0x0034 SMHC Masked Interrupt Status Register (Default Value: 0x0000\_0000)

Offset: 0x0034			Register Name: SMHC_MINTSTS
Bit	Read/Write	Default/Hex	Description
31	R	0x0	M_CARD_REMOVAL_INT Card Removed
30	R	0x0	M_CARD_INSERT Card Inserted
29:17	/	/	/
16	R	0x0	M_SDIO_INT SDIO Interrupt
15	R	0x0	M_DEF_INT Data End-bit Error When the bit is set during receiving data, it means that the host controller does not receive the valid data end bit. When the bit is set during transmitting data, it means that the host controller does not receive the CRC status token. This is a write-1-to-clear bit.

Offset: 0x0034			Register Name: SMHC_MINTSTS
Bit	Read/Write	Default/Hex	Description
14	R	0x0	M_ACD_INT Auto Command Done When set, it means auto-stop command (CMD12) completed.
13	R	0x0	M_DSE_BC_INT Data Start Error/Busy Clear When set during receiving data, it means that the host controller found an error start bit. When the bit is set during transmitting data, it means that the busy signal is cleared after the last block.
12	R	0x0	M_CB_IW_INT Command Busy and Illegal Write
11	R	0x0	M_FU_FO_INT FIFO Underrun/Overflow
10	R	0x0	M_DSTO_VSD_INT Data Starvation Timeout/V1.8 Switch Done
9	R	0x0	M.DTO_BDS_INT Data Timeout/Boot Data Start
8	R	0x0	M.RTO_BACK_INT Response Timeout/Boot ACK Received
7	R	0x0	M.DCE_INT Data CRC Error When the bit is set during receiving data, it means that the received data have data CRC error. When the bit is set during transmitting data, it means that the received CRC status taken is negative.
6	R	0x0	M.RCE_INT Response CRC Error
5	R	0x0	M.DRR_INT Data Receive Request When set, it means that there are enough data in FIFO during receiving data.
4	R	0x0	M.DTR_INT Data Transmit Request When set, it means that there is enough space in FIFO during transmitting data.

Offset: 0x0034			Register Name: SMHC_MINTSTS
Bit	Read/Write	Default/Hex	Description
3	R	0x0	M_DTC_INT Data Transfer Complete
2	R	0x0	M_CC_INT Command Complete
1	R	0x0	M_RE_INT Response Error When set, Transmit Bit error or End Bit error or CMD Index error may occurs.
0	/	/	/

#### 6.2.6.15 0x0038 SMHC Raw Interrupt Status Register (Default Value: 0x0000\_0000)

Offset: 0x0038			Register Name: SMHC_RINTSTS
Bit	Read/Write	Default/Hex	Description
31	R/W1C	0x0	CARD_REMOVAL Card Removed Write 1 to clear this bit.
30	R/W1C	0x0	CARD_INSERT Card Inserted Write 1 to clear this bit.
29:17	/	/	/
16	R/W1C	0x0	SDIOI_INT SDIO Interrupt Write 1 to clear this bit.
15	R/W1C	0x0	DEE Data End-bit Error When the bit is set during receiving data, it means that the host controller does not receive the valid data end bit. When the bit is set during transmitting data, it means that the host controller does not receive the CRC status token. Write 1 to clear this bit.

Offset: 0x0038			Register Name: SMHC_RINTSTS
Bit	Read/Write	Default/Hex	Description
14	R/W1C	0x0	<p>ACD Auto Command Done When set, it means that the auto-stop command (CMD12) is completed. Write 1 to clear this bit.</p>
13	R/W1C	0x0	<p>DSE_BC Data Start Error/Busy Clear When the bit is set during receiving data, it means that the host controller found an error start bit. When the bit is set during transmitting data, it means that the busy signal is cleared after the last block. Write 1 to clear this bit.</p>
12	R/W1C	0x0	<p>CB_IW Command Busy and Illegal Write Write 1 to clear this bit.</p>
11	R/W1C	0x0	<p>FU_FO FIFO Underrun/Overflow Write 1 to clear this bit.</p>
10	R/W1C	0x0	<p>DSTO_VSD Data Starvation Timeout/V1.8 Switch Done Write 1 to clear this bit.</p>
9	R/W1C	0x0	<p>DTO_BDS Data Timeout/Boot Data Start When the bit is set during receiving data, it means that some of the channel of DATA[3:0] lack of the start bit. Write 1 to clear this bit.</p>
8	R/W1C	0x0	<p>RTO_BACK Response Timeout/Boot ACK Received Write 1 to clear this bit.</p>
7	R/W1C	0x0	<p>DCE Data CRC Error When the bit is set during receiving data, it means that the received data have data CRC error. When the bit is set during transmitting data, it means that the received CRC status taken is negative. Write 1 to clear this bit.</p>

Offset: 0x0038			Register Name: SMHC_RINTSTS
Bit	Read/Write	Default/Hex	Description
6	R/W1C	0x0	<p>RCE</p> <p>Response CRC Error</p> <p>Write 1 to clear this bit.</p>
5	R/W1C	0x0	<p>DRR</p> <p>Data Receive Request</p> <p>When set, it means that there are enough data in FIFO during receiving data.</p> <p>Write 1 to clear this bit.</p>
4	R/W1C	0x0	<p>DTR</p> <p>Data Transmit Request</p> <p>When set, it means that there is enough space in FIFO during transmitting data.</p> <p>Write 1 to clear this bit.</p>
3	R/W1C	0x0	<p>DTC</p> <p>Data Transfer Complete</p> <p>When set, it means that the current command completes even through error occurs.</p> <p>Write 1 to clear this bit.</p>
2	R/W1C	0x0	<p>CC</p> <p>Command Complete</p> <p>When set, it means that the current command completes even through error occurs.</p> <p>Write 1 to clear this bit.</p>
1	R/W1C	0x0	<p>RE</p> <p>Response Error</p> <p>When set, it means that the transmit bit error, end bit error, or CMD index error may occur.</p> <p>Write 1 to clear this bit.</p>
0	/	/	/

**6.2.6.16 0x003C SMHC Status Register (Default Value: 0x0000\_0006)**

Offset: 0x003C			Register Name: SMHC_STATUS
Bit	Read/Write	Default/Hex	Description
31	R	0x0	DMA_REQ DMA Request DMA request signal state
30:26	/	/	/
25:17	R	0x0	FIFO_LEVEL FIFO Level Number of filled locations in FIFO
16:11	R	0x0	RESP_IDX Response Index Index of previous response, including any auto-stop sent by the controller.
10	R	0x0	FSM_BUSY Data FSM Busy Data transmit or receive state-machine is busy.
9	R	0x0	CARD_BUSY Card Data Busy Inverted version of DATA[0] 0: Card data is not busy. 1: Card data is busy.
8	R	0x0	CARD_PRESENT Data[3] Status The level of DATA[3], checks whether the card is present. 0: The card is not present. 1: The card is present.

Offset: 0x003C			Register Name: SMHC_STATUS
Bit	Read/Write	Default/Hex	Description
7:4	R	0x0	<p>FSM_STA                      Command FSM States                      0000: Idle                      0001: Send init sequence                      0010: TX CMD start bit                      0011: TX CMD TX bit                      0100: TX CMD index + argument                      0101: TX CMD CRC7                      0110: TX CMD end bit                      0111: RX response start bit                      1000: RX response IRQ response                      1001: RX response TX bit                      1010: RX response CMD index                      1011: RX response data                      1100: RX response CRC7                      1101: RX response end bit                      1110: CMD path wait NCC                      1111: Wait; CMD-to-response turn around</p>
3	R	0x0	<p>FIFO_FULL                      FIFO Full                      0: FIFO is not full                      1: FIFO is full</p>
2	R	0x1	<p>FIFO_EMPTY                      FIFO Empty                      0: FIFO is not empty                      1: FIFO is empty</p>
1	R	0x1	<p>FIFO_TX_LEVEL                      FIFO TX Water Level Flag                      0: FIFO does not reach the transmit trigger level                      1: FIFO reaches the transmit trigger level</p>
0	R	0x0	<p>FIFO_RX_LEVEL                      FIFO RX Water Level Flag                      0: FIFO does not reach the receive trigger level.                      1: FIFO reaches the receive trigger level.</p>

#### 6.2.6.17 0x0040 SMHC FIFO Water Level Register (Default Value: 0x000F\_0000)

Offset: 0x0040			Register Name: SMHC_FIFOTH
Bit	Read/Write	Default/Hex	Description
31	/	/	/
30:28	R/W	0x0	<p>BSIZE_OF_TRANS Burst Size of Multiple Transaction 000: 1 transfers 001: 4 010: 8 011: 16 Others: Reserved</p> <p>It should be programmed the same as the DMA controller multiple transaction size. The units for the transfer are the DWORD. A single transfer would be signaled based on this value. The value should be sub-multiple of (RX_TL + 1) and (FIFO_DEPTH - TX_TL)</p> <p>Recommended: FIFO_DEPTH = 256, FIFO_SIZE = 256 * 32 = 1K MSize = 16, TX_TL = 240, RX_TL = 15</p>
27:24	/	/	/
23:16	R/W	0xF	<p>RX_TL RX Trigger Level 0x0 to 0xFE: The RX trigger level is from 0 to 254. 0xFF: Reserved</p> <p>Indicates the FIFO threshold for the FIFO request host to receive data from the FIFO. When the FIFO data level is greater than this value, the DMA request is raised if DMA enabled, or the RX interrupt bit is set if interrupt enabled. At the end of the packet, if the last transfer is less than this level, the value is ignored and the relative request will be raised as usual.</p> <p>Recommended: 15 (means greater than 15)</p>
15:8	/	/	/

Offset: 0x0040			Register Name: SMHC_FIFOTH
Bit	Read/Write	Default/Hex	Description
7:0	R/W	0x0	<p>TX_TL TX Trigger Level 0x1 to 0xFF: The TX trigger level is 1 to 255. 0x0: No trigger</p> <p>Indicates the FIFO threshold for the FIFO request host to transmit data to the FIFO. When the FIFO data level is less than or equal to this value, the DMA TX request is raised if DMA enabled, or TX request interrupt bit is set if interrupt enabled. At the end of the packet, if the last transfer is less than this level, the value is ignored and the relative request will be raised as usual.</p> <p>Recommended: 240 (means less than or equal to 240)</p>

#### 6.2.6.18 0x0044 SMHC Function Select Register (Default Value: 0x0000\_0000)

Offset: 0x0044			Register Name: SMHC_FUNS
Bit	Read/Write	Default/Hex	Description
31:3	/	/	/
2	R/W	0x0	<p>ABT_RDATA Abort Read Data 0: Ignored 1: After the suspend command is issued during the read-transfer, the software polls card to find when the suspend happens. Once the suspend occurs, the software sets the bit to reset the data state-machine, which is waiting for the next block of data.</p> <p>This bit is used in the SDIO card suspends sequence and is auto-cleared once the controller resets to the idle state.</p>
1	R/W	0x0	<p>READ_WAIT Read Wait 0: Clear SDIO read wait 1: Assert SDIO read wait</p>

Offset: 0x0044			Register Name: SMHC_FUNS
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	<p>HOST_SEND_MMC_IRQRESQ Host Send MMC IRQ Response 0: Ignored 1: Send auto IRQ response</p> <p>When the host is waiting for the MMC card interrupt response, setting this bit will make the controller cancel the waiting state and return to the idle state, at which time, the controller will receive the IRQ response sent by itself.</p> <p>This bit is auto-cleared after the response is sent.</p>

#### 6.2.6.19 0x0048 SMHC Transferred Byte Count Register 0 (Default Value: 0x0000\_0000)

Offset: 0x0048			Register Name: SMHC_TBC0
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	<p>TBC0 Transferred Count 0 The number of bytes transferred between the card and internal FIFO. The register should be accessed in full to avoid read-coherency problems and read only after the data transfer completes.</p>

#### 6.2.6.20 0x004C SMHC Transferred Byte Count Register 1 (Default Value: 0x0000\_0000)

Offset: 0x004C			Register Name: SMHC_TBC1
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	<p>TBC1 Transferred Count 1 The number of bytes transferred between the Host/DMA memory and internal FIFO. The register should be accessed in full to avoid read-coherency problems and read only after the data transfer completes.</p>

#### 6.2.6.21 0x0054 SMHC CRC Status Detect Control Register (Default Value: 0x0000\_0003)

Offset: 0x0054			Register Name: SMHC_CSDC
Bit	Read/Write	Default/Hex	Description
31:4	/	/	/
3:0	R/W	0x3	CRC_DET_PARA 110: HS400 speed mode 011: Other speed mode Others: Reserved

#### 6.2.6.22 0x0058 SMHC Auto Command 12 Argument Register (Default Value: 0x0000\_FFFF)

Offset: 0x0058			Register Name: SMHC_A12A
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xffff	SD_A12A Auto CMD12 Argument The argument of command 12 automatically sent by the controller.

#### 6.2.6.23 0x005C SMHC New Timing Set Register (Default Value: 0x8171\_0000)

Offset: 0x005C			Register Name: SMHC_NTSR
Bit	Read/Write	Default/Hex	Description
31	R/W	0x1	MODE_SELECT 0: Old mode of Sample/Output Timing 1: New mode of Sample/Output Timing
30:25	/	/	/
24	R/W	0x1	CMD_DAT_RX_PHASE_CLR Clear the input phase of command lines and data lines during the update clock operation. 0: Disabled 1: Enabled
23	/	/	/

Offset: 0x005C			Register Name: SMHC_NTSR
Bit	Read/Write	Default/Hex	Description
22	R/W	0x1	DAT_CRC_STATUS_RX_PHASE_CLR Clear the input phase of data lines before receiving the CRC status. 0: Disabled 1: Enabled
21	R/W	0x1	DAT_TRANS_RX_PHASE_CLR Clear the input phase of data lines before transferring the data. 0: Disabled 1: Enabled
20	R/W	0x1	DAT_RECV_RX_PHASE_CLR Clear the input phase of data lines before receiving the data. 0: Disabled 1: Enabled
19:17	/	/	/
16	R/W	0x1	CMD_SEND_RX_PHASE_CLR Clear command RX phase before sending the command. 0: Disabled 1: Enabled
15:10	/	/	/
9:8	R/W	0x0	DAT_SAMPLE_TIMING_PHASE 00: Sample timing phase offset 90° 01: Sample timing phase offset 180° 10: Sample timing phase offset 270° 11: Sample timing phase offset 0°(only for SD2 hs400 mode)
7:6	/	/	/
5:4	R/W	0x0	CMD_SAMPLE_TIMING_PHASE 00: Sample timing phase offset 90° 01: Sample timing phase offset 180° 10: Sample timing phase offset 270° 11: Ignore
3:1	/	/	/
0	R/W	0x0	HS400_NEW_SAMPLE_EN 0: Disable hs400 new sample method 1: Enable hs400 new sample method

#### 6.2.6.24 0x0078 SMHC Hardware Reset Register (Default Value: 0x0000\_0001)

Offset: 0x0078			Register Name: SMHC_HWRST
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x1	<p>HW_RST 1: Active mode 0: Reset</p> <p>This bit causes the cards to enter the pre-idle state, which requires them to be re-initialized.</p>

#### 6.2.6.25 0x0080 SMHC IDMAC Control Register (Default Value: 0x0000\_0000)

Offset: 0x0080			Register Name: SMHC_IDMAC
Bit	Read/Write	Default/Hex	Description
31	W	0x0	<p>DES_LOAD_CTRL</p> <p>When IDMAC fetches a descriptor, if the valid bit of a descriptor is not set, IDMAC FSM will enter the suspend state. Setting this bit will make the IDMAC refetch the descriptor and transfer data normally.</p>
30:11	/	/	/
10:8	R	0x0	Reserved
7	R/W	0x0	<p>IDMAC_ENB</p> <p>IDMAC Enable</p> <p>When set, IDMAC is enabled.</p>
6:2	R/W	0x0	Reserved
1	R/W	0x0	<p>FIX_BUST_CTRL</p> <p>Fixed Burst</p> <p>Controls whether the AHB Master interface performs fixed burst transfers or not. When set, the AHB will use only SINGLE, INCR4, and INCR8 during the start of normal burst transfers. When reset, the AHB will use SINGLE and INCR burst transfer operations.</p>
0	R/W	0x0	<p>IDMAC_RST</p> <p>DMA Reset</p> <p>When set, the DMA Controller resets all its internal registers. It is automatically cleared after 1 clock cycle.</p>

#### 6.2.6.26 0x0084 SMHC Descriptor List Base Address Register (Default Value: 0x0000\_0000)

Offset: 0x0084			Register Name: SMHC_DLBA_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<p>DES_BASE_ADDR Start of Descriptor List Contains the base address of the First Descriptor. It is a word (4 Byte) address.</p>

#### 6.2.6.27 0x0088 SMHC IDMAC Status Register (Default Value: 0x0000\_0000)

Offset: 0x0088			Register Name: SMHC_IDST_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16:13	R	0x0	Reserved
12:10	R	0x0	<p>IDMAC_ERR_STA Error Bits Indicates the type of error that caused a Bus Error. Valid only with Fatal Bus Error bit (IDST[2]) set. This field will not generate an interrupt.</p> <p>001: Host Abort received during the transmission. 010: Host Abort received during the reception. Others: Reserved</p>
9	R/W1C	0x0	<p>ABN_INT_SUM (AIS) Abnormal Interrupt Summary Logical OR of the following: IDST[2]: Fatal Bus Interrupt IDST[4]: Descriptor Unavailable Bit Interrupt IDST[5]: Card Error Summary Interrupt Only unmasked bits affect this bit. This is a sticky bit and must be cleared each time a corresponding bit that causes AIS to be set is cleared.</p>

Offset: 0x0088			Register Name: SMHC_IDST_REG
Bit	Read/Write	Default/Hex	Description
8	R/W1C	0x0	<p>NOR_INT_SUM (NIS)  Normal Interrupt Summary  Logical OR of the following:  IDST[0]: Transmit Interrupt  IDST[1]: Receive Interrupt  Only unmasked bits affect this bit.  This is a sticky bit and must be cleared each time a corresponding bit that causes NIS to be set is cleared.</p>
7:6	/	/	/
5	R/W1C	0x0	<p>ERR_FLAG_SUM  Card Error Summary  Indicates the status of the transaction to/from the card; also present in RINTSTS. Indicates the logical OR of the following bits:  EBC: End Bit Error  RTO: Response Timeout  RCRC: Response CRC  SBE: Start Bit Error  DRTO: Data Read Timeout  DCRC: Data CRC for Receive  RE: Response Error  Writing 1 clears this bit.</p>
4	R/W1C	0x0	<p>DES_UNAVL_INT  Descriptor Unavailable Interrupt  This bit is set when the descriptor is unavailable due to OWN bit = 0 (DESO[31] =0).  Writing 1 clears this bit.</p>
3	/	/	/
2	R/W1C	0x0	<p>FATAL_BERR_INT  Fatal Bus Error Interrupt  Indicates that a Bus Error occurred (IDST[12:10]). When this bit is set, the DMA disables all its bus accesses.  Writing 1 clears this bit.</p>
1	R/W1C	0x0	<p>RX_INT  Receive Interrupt  Indicates the completion of data reception for a descriptor.  Writing 1 clears this bit.</p>

Offset: 0x0088			Register Name: SMHC_IDST_REG
Bit	Read/Write	Default/Hex	Description
0	R/W1C	0x0	<p>TX_INT Transmit Interrupt Indicates that data transmission is finished for a descriptor. Writing 1 clears this bit.</p>

#### 6.2.6.28 0x008C SMHC IDMAC Interrupt Enable Register (Default Value: 0x0000\_0000)

Offset: 0x008C			Register Name: SMHC_IDIE_REG
Bit	Read/Write	Default/Hex	Description
31:6	/	/	/
5	R/W	0x0	<p>ERR_SUM_INT_ENB Card Error Summary Interrupt Enable. When set, it enables the Card Interrupt Summary.</p>
4	R/W	0x0	<p>DES_UNAVL_INT_ENB Descriptor Unavailable Interrupt. When set along with Abnormal Interrupt Summary Enable, the Descriptor Unavailable Interrupt is enabled.</p>
3	/	/	/
2	R/W	0x0	<p>FERR_INT_ENB Fatal Bus Error Enable When set with Abnormal Interrupt Summary Enable, the Fatal Bus Error Interrupt is enabled. When reset, the Fatal Bus Error Enable Interrupt is disabled.</p>
1	R/W	0x0	<p>RX_INT_ENB Receive Interrupt Enable. When set with Normal Interrupt Summary Enable, Receive Interrupt is enabled. When reset, the Receive Interrupt is disabled.</p>
0	R/W	0x0	<p>TX_INT_ENB Transmit Interrupt Enable. When set with Normal Interrupt Summary Enable, Transmit Interrupt is enabled. When reset, the Transmit Interrupt is disabled.</p>

**6.2.6.29 0x0100 SMHC Card Threshold Control Register (Default Value: 0x0000\_0000)**

Offset: 0x0100			Register Name: SMHC_THLD
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27:16	R/W	0x0	CARD_WR_THLD Card Read/Write Threshold Size
15:3	/	/	/
2	R/W	0x0	CARD_WR_THLD_ENB Card Write Threshold Enable 0: Card write threshold disabled 1: Card write threshold enabled Host controller initiates write transfer only if the card threshold amount of data is available in transmit FIFO.
1	R/W	0x0	BCIG Busy Clear Interrupt Generation 0: Busy clear interrupt disabled 1: Busy clear interrupt enabled The application can disable this feature if it does not want to wait for a Busy Clear Interrupt.
0	R/W	0x0	CARD_RD_THLD_ENB Card Read Threshold Enable 0: Card read threshold disabled 1: Card read threshold enabled Host controller initiates Read Transfer only if the CARD_RD_THLD amount of space is available in receive FIFO.

**6.2.6.30 0x0104 SMHC Sample FIFO Control Register (Default Value: 0x0000\_0006)**

Offset: 0x0104			Register Name: SMHC_SFC
Bit	Read/Write	Default/Hex	Description
31:5	/	/	/

Offset: 0x0104			Register Name: SMHC_SFC
Bit	Read/Write	Default/Hex	Description
4:1	R/W	0x3	<p>STOP_CLK_CTRL Stop Clock Control When receiving data, if CARD_RD_THLD_ENB is set and CARD_RD_THLD is set the same to BLK_SZ, the device clock may stop at the block gap during data receiving.</p> <p>This field is used to control the position of the stopping clock. The value can be changed between 0x0 and 0xF. The available value and the position of the stopping clock must be decided by the actual situation.</p> <p>The value increases one in this field is linked to one cycle (two cycles in DDR mode) that the position of the stopping clock moved up.</p>
0	R/W	0x0	<p>BYPASS_EN Bypass enable When set, sample FIFO will be bypassed.</p>

#### 6.2.6.31 0x0108 SMHC Auto Command 23 Argument Register (Default Value: 0x0000\_0000)

Offset: 0x0108			Register Name: SMHC_A23A
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<p>A23A Auto CMD23 Argument The argument of command 23 is automatically sent by controller with this field.</p>

#### 6.2.6.32 0x010C SMHC eMMC4.5 DDR Start Bit Detection Control Register (Default Value: 0x0000\_0000)

Offset: 0x010C			Register Name: EMMC_DDR_SBIT_DET
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<p>HS400_MD_EN HS400 Mode Enable 0: Disabled 1: Enabled It is required to set this bit to '1' before initiating any data transfer CMD in HS400 mode.</p>
30:1	/	/	/

Offset: 0x010C			Register Name: EMMC_DDR_SBIT_DET
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	<p>HALF_START_BIT Control for start bit detection mechanism inside mstorage based on duration of start bit.</p> <p>For eMMC 4.5, start bit can be:</p> <p>0: Full cycle 1: Less than one full cycle</p> <p>Set HALF_START_BIT = 1 for eMMC 4.5 and above; set to 0 for SD applications.</p>

#### 6.2.6.33 0x0138 SMHC Extended Command Register (Default Value: 0x0000\_0000)

Offset: 0x0138			Register Name: SMHC_EXT_CMD
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	<p>AUTO_CMD23_EN Send CMD23 Automatically</p> <p>When setting this bit, send CMD23 automatically before sending the command specified in the SMHC_CMD register.</p> <p>When SOFT_RST is set, this field will be cleared.</p>

#### 6.2.6.34 0x013C SMHC Extended Response Register (Default Value: 0x0000\_0000)

Offset: 0x013C			Register Name: SMHC_EXT_RESP
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	<p>SMHC_EXT_RESP</p> <p>When AUTO_CMD23_EN is set, this register stores the response of CMD23.</p>

#### 6.2.6.35 0x0140 SMHC Drive Delay Control Register (Default Value: 0x0001\_0000)

Offset: 0x0140			Register Name: SMHC_DRV_DL
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/

Offset: 0x0140			Register Name: SMHC_DRV_DL
Bit	Read/Write	Default/Hex	Description
17	R/W	0x0	<p>DAT_DRV_PH_SEL Data Drive Phase Select</p> <p><b>When 0x005C[31]=0:</b></p> <p>0: Data drive phase offset is 90° at SDR mode, 45° at DDR8 mode, and 90° at DDR4 mode.</p> <p>1: Data drive phase offset is 180° at SDR mode, 90° at DDR8 mode, and 0° at DDR4 mode.</p> <p><b>When 0x005C[31]=1:</b></p> <p>0: Data drive phase offset is 90° at SDR mode, and 45° at DDR mode.</p> <p>1: Data drive phase offset is 180° at SDR mode, and 90° at DDR mode.</p>
16	R/W	0x1	<p>CMD_DRV_PH_SEL Command Drive Phase Select</p> <p><b>When 0x005C[31]=0:</b></p> <p>0: Command drive phase offset is 90° at SDR mode, 45° at DDR8 mode, and 90° at DDR4 mode.</p> <p>1: Command drive phase offset is 180° at SDR mode, 90° at DDR8 mode, and 180° at DDR4 mode.</p> <p><b>When 0x005C[31]=1:</b></p> <p>0: Command drive phase offset is 90° at SDR mode and 45° at DDR mode.</p> <p>1: Command drive phase offset is 180° at SDR mode and 90° at DDR mode.</p>
15:0	/	/	/

#### 6.2.6.36 0x0144 SMHC Sample Delay Control Register (Default Value: 0x0000\_2000)

Offset: 0x0144			Register Name: SMHC_SAMP_DL
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15	R/W	0x0	<p>SAMP_DL_CAL_START Sample Delay Calibration Start</p> <p>When set, it means that start sample delay chain calibration.</p>

Offset: 0x0144			Register Name: SMHC_SAMP_DL
Bit	Read/Write	Default/Hex	Description
14	R	0x0	<p>SAMP_DL_CAL_DONE Sample Delay Calibration Done When set, it means that sample delay chain calibration is done and the result of calibration is shown in SAMP_DL.</p>
13:8	R	0x20	<p>SAMP_DL Sample Delay It indicates the number of delay cells corresponding to the current card clock. The delay time generated by these delay cells is equal to the cycle of the card clock nearly. Generally, it is necessary to do drive delay calibration when the card clock is changed. This bit is valid only when SAMP_DL_CAL_DONE is set.</p>
7	R/W	0x0	<p>SAMP_DL_SW_EN Sample Delay Software Enable When set, it means that enable the sample delay specified at SAMP_DL_SW.</p>
6	/	/	/
5:0	R/W	0x0	<p>SAMP_DL_SW Sample Delay Software The relative delay between the clock line and command line, data line. It can be determined according to the value of SAMP_DL, the cycle of the card clock and the input timing requirement of the device.</p>

#### 6.2.6.37 0x0148 SMHC Data Strobe Delay Control Register (Default Value: 0x0000\_2000)

Offset: 0x0148			Register Name: SMHC_DS_DL
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15	R/W	0x0	<p>DS_DL_CAL_START Data Strobe Delay Calibration Start When set, it means that start sample delay chain calibration.</p>
14	R	0x0	<p>DS_DL_CAL_DONE Data Strobe Delay Calibration Done When set, it means that sample delay chain calibration is done and the result of calibration is shown in DS_DL.</p>

Offset: 0x0148			Register Name: SMHC_DS_DL
Bit	Read/Write	Default/Hex	Description
13:8	R	0x20	<p>DS_DL Data Strobe Delay</p> <p>It indicates the number of delay cells corresponding to the current card clock. The delay time generated by these delay cells is equal to the cycle of SMHC's clock nearly.</p> <p>This bit is valid only when SAMP_DL_CAL_DONE is set.</p>
7	R/W	0x0	<p>DS_DL_SW_EN Sample Delay Software Enable</p>
6	/	/	/
5:0	R/W	0x0	<p>DS_DL_SW Data Strobe Delay Software</p>

#### 6.2.6.38 0x014C SMHC HS400 New Timing Delay Control Register (Default Value: 0x0000\_8000)

Offset: 0x014C			Register Name: SMHC_HS400_DL
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15	R/W	0x0	<p>HS400_DL_CAL_START HS400 Delay Calibration Start</p> <p>When sets, it means that start sample delay chain calibration.</p>
14	R	0x0	<p>HS400_DL_CAL_DONE HS400 Delay Calibration Done</p> <p>When set, it means that sample delay chain calibration is done and the result of calibration is shown in HS400_DL.</p>
13:12	/	/	/
11:8	R	0x8	<p>HS400_DL HS400 Delay</p> <p>It indicates the number of delay cells corresponding to the current card clock. The delay time generated by these delay cells is equal to the cycle of the SMHC clock nearly.</p> <p>This bit is valid only when HS400_DL_CAL_DONE is set.</p>
7	R/W	0x0	<p>HS400_DL_SW_EN Sample Delay Software Enable</p>
6	/	/	/

Offset: 0x014C			Register Name: SMHC_HS400_DL
Bit	Read/Write	Default/Hex	Description
3:0	R/W	0x0	HS400_DL_SW HS400 Delay Software

#### 6.2.6.39 0x0200 SMHC FIFO Register (Default Value: 0x0000\_0000)

Offset: 0x0200			Register Name: SMHC_FIFO
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	TX/RX_FIFO Data FIFO

# Contents

7	Audio.....	488
7.1	I2S/PCM.....	488
7.1.1	Overview .....	488
7.1.2	Block Diagram .....	488
7.1.3	Functional Description .....	490
7.1.4	Programming Guidelines.....	496
7.1.5	Register List.....	498
7.1.6	Register Description .....	500
7.2	DMIC.....	548
7.2.1	Overview .....	548
7.2.2	Block Diagram .....	548
7.2.3	Functional Description .....	548
7.2.4	Register List.....	551
7.2.5	Register Description .....	551
7.3	OWA .....	566
7.3.1	Overview .....	566
7.3.2	Block Diagram .....	566
7.3.3	Functional Description .....	567
7.3.4	Register List.....	571
7.3.5	Register Description .....	572
7.4	Audio Codec .....	582
7.4.1	Overview .....	582
7.4.2	Block Diagram .....	582
7.4.3	Functional Description .....	583
7.4.4	Programming Guidelines.....	590
7.4.5	Register List.....	591
7.4.6	Register Description .....	593

# Figures

Figure 7-1 I2S/PCM Interface System Block Diagram .....	489
Figure 7-2 Typical Application of I2S/PCM Interface .....	489
Figure 7-3 I2S Standard Mode Timing.....	491
Figure 7-4 Left-Justified Mode Timing .....	491
Figure 7-5 Right-Justified Mode Timing .....	492
Figure 7-6 PCM Long Frame Mode Timing.....	492
Figure 7-7 PCM Short Frame Mode Timing.....	492
Figure 7-8 16-Channel Mapping Configuration.....	493
Figure 7-9 Timing Requirements for Inputs .....	494
Figure 7-10 Timing Requirements for Outputs .....	494
Figure 7-11 I2S/PCM Operation Flow .....	495
Figure 7-12 DMIC Block Diagram .....	548
Figure 7-13 DMIC Operation Mode.....	550
Figure 7-14 OWA Block Diagram.....	566
Figure 7-15 OWA Biphase-Mark Code .....	568
Figure 7-16 OWA Sub-Frame Format.....	569
Figure 7-17 OWA Frame/Block Format.....	570
Figure 7-18 OWA Operation Flow .....	571
Figure 7-19 Audio Codec Block Diagram.....	583
Figure 7-20 Audio Codec Clock Diagram.....	584
Figure 7-21 Audio Codec Digital Part Reset System.....	585
Figure 7-22 Audio Codec Analog Part Reset System.....	585
Figure 7-23 Audio Codec Data Path Diagram.....	586
Figure 7-24 Headphone Output Application.....	586
Figure 7-25 Audio Codec Interrupt System.....	587
Figure 7-26 DAP Data Flow .....	587
Figure 7-27 HPF Logic Structure .....	588
Figure 7-28 DRC Block Diagram .....	588
Figure 7-29 DRC Static Curve Parameters.....	588
Figure 7-30 Energy Filter Structure.....	589

---

Figure 7-31 Gain Smooth Filter ..... 590

## Tables

Table 7-1 I2S/PCM External Signals .....	490
Table 7-2 I2S/PCM Clock Sources.....	490
Table 7-3 DIN Slot ID and Encoder .....	493
Table 7-4 Proper MCLK Values with Different Fsin and Fsout .....	495
Table 7-5 DMIC External Signals .....	549
Table 7-6 DMIC Clock Sources.....	549
Table 7-7 OWA Sub-blocks.....	567
Table 7-8 OWA External Signals .....	567
Table 7-9 OWA Clock Sources .....	567
Table 7-10 Biphase-Mark Encoder .....	568
Table 7-11 Preamble Codes .....	569
Table 7-12 The Corresponding Relation between Different System Clock and Sample Ratio.....	570
Table 7-13 Audio Codec External Signals .....	583

# 7 Audio

## 7.1 I2S/PCM

### 7.1.1 Overview

The I2S/PCM controller is designed to transfer streaming audio-data between the system memory and the codec chip. The controller supports standard I2S format, Left-justified mode format, Right-justified mode format, PCM mode format, and TDM mode format.

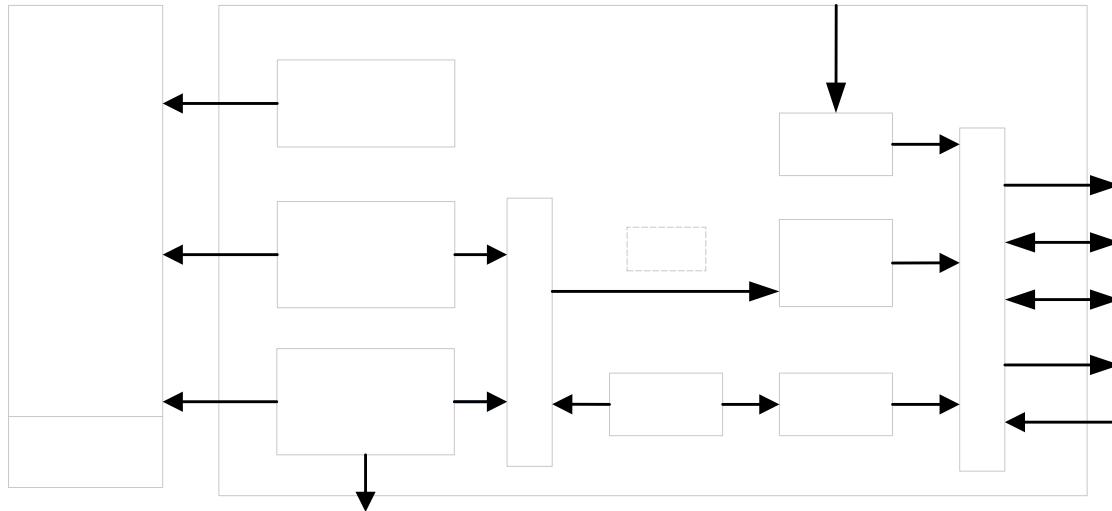
The I2S/PCM controller includes the following features:

- Two I2S/PCM external interfaces (I2S1, I2S2) for connecting external power amplifier and MIC ADC
- Compliant with standard Philips Inter-IC sound (I2S) bus specification
  - Left-justified, Right-justified, PCM mode, and Time Division Multiplexing (TDM) format
  - Programmable PCM frame width: 1 BCLK width (short frame) and 2 BCLKs width (long frame)
- Transmit and Receive data FIFOs
  - Programmable FIFO thresholds
  - 128 depth x 32-bit width TXFIFO and 64 depth x 32-bit width RXFIFO
- Supports multiple function clock
  - Clock up to 24.576 MHz Data Output of I2S/PCM in Master mode (Only if the IO PAD and Peripheral I2S/PCM satisfy Timing Parameters)
  - Clock up to 12.288 MHz Data Input of I2S/PCM in Master mode
- Supports TX/RX DMA Slave interface
- Supports multiple application scenarios
  - Up to 16 channels ( $f_s = 48$  kHz) which has adjustable width from 8-bit to 32-bit
  - Sample rate from 8 kHz to 384 kHz (CHAN = 2)
  - 8-bit u-law and 8-bit A-law companded sample
- Supports master/slave mode

### 7.1.2 Block Diagram

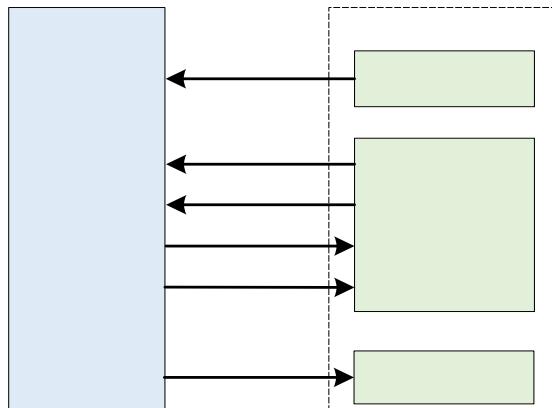
The following figure shows the functional block diagram of the I2S/PCM interface.

Figure 7-1 I2S/PCM Interface System Block Diagram



The following figure shows the typical application of the I2S/PCM interface.

Figure 7-2 Typical Application of I2S/PCM Interface



The I2S/PCM interface system integrates one I2S\_TX and one I2S\_RX.

- The I2S\_TX is for playing music in I2S or PCM format.
- The I2S\_RX is for receiving data in I2S or PCM format.
- When the I2S works in the master mode, the external I2S module provides BCLK\_in and LRCK\_in for the clock management unit (CMU), and the I2S\_TX and I2S\_RX work with the two external clocks.
- When the I2S works in the slave mode, the CMU provides clocks BCLK\_out and LRCK\_out for the external I2S module, and the I2S\_TX and I2S\_RX work with the internal clocks.

### 7.1.3 Functional Description

#### 7.1.3.1 External Signals

The following table describes the external signals of the I2S/PCM interface.

LRCK and BCLK are bidirectional I/O. When the I2S/PCM interface works in the Master mode, LRCK and BCLK are output pins. When the I2S/PCM interface works in the Slave mode, LRCK and BCLK are input pins.

MCLK is an output pin for external devices. DOUT are the serial data output pins and DIN are the serial data input pins. For details about General Purpose I/O port, refer to section 9.7 “[GPIO](#)”.

**Table 7-1 I2S/PCM External Signals**

Signal Name	Description	Type
I2S1-MCLK	I2S1 Master Clock	O
I2S1-LRCK	I2S1/PCM1 Sample Rate Clock/Sync	I/O
I2S1-BCLK	I2S1/PCM1 Bit Rate Clock	I/O
I2S1-DOUT[1:0]	I2S1/PCM1 Serial Data Output Channel [1:0]	O
I2S1-DIN[1:0]	I2S1/PCM1 Serial Data Input Channel [1:0]	I
I2S2-MCLK	I2S2 Master Clock	O
I2S2-LRCK	I2S2/PCM2 Sample Rate Clock/Sync	I/O
I2S2-BCLK	I2S2/PCM2 Bit Rate Clock	I/O
I2S2-DOUT[1:0], I2S2-DOUT3	I2S2/PCM2 Serial Data Output Channel [1:0] and Channel 3	O
I2S2-DIN[1:0], I2S2-DIN[3s]	I2S2/PCM2 Serial Data Input Channel [1:0] and Channel 3	I

#### 7.1.3.2 Clock Sources

The following table describes the clock sources for I2S/PCM. For clock setting, configurations, and gating information, refer to section 3.3 “[CCU](#)”.

**Table 7-2 I2S/PCM Clock Sources**

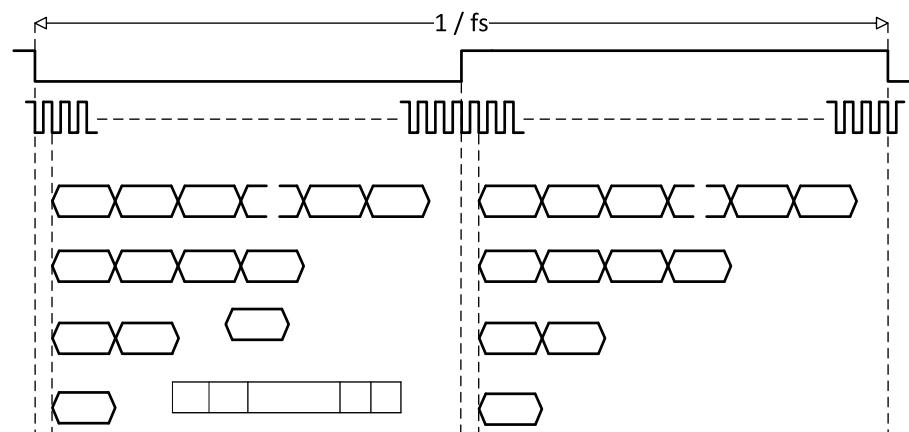
Clock Name	Description
PLL_AUDIO0(1X)	By default, PLL_AUDIO0(1X) is 24.5714 MHz, and PLL_AUDIO0(4X) is 98.2856 MHz.
PLL_AUDIO0(4X)	
PLL_AUDIO1(DIV2)	By default, PLL_AUDIO1 is 3072 MHz, PLL_AUDIO1(DIV2) is 1536 MHz, and
PLL_AUDIO1(DIV5)	PLL_AUDIO1(DIV5) is 614.4 MHz (24.576 MHz*25).

### 7.1.3.3 Timing Diagram

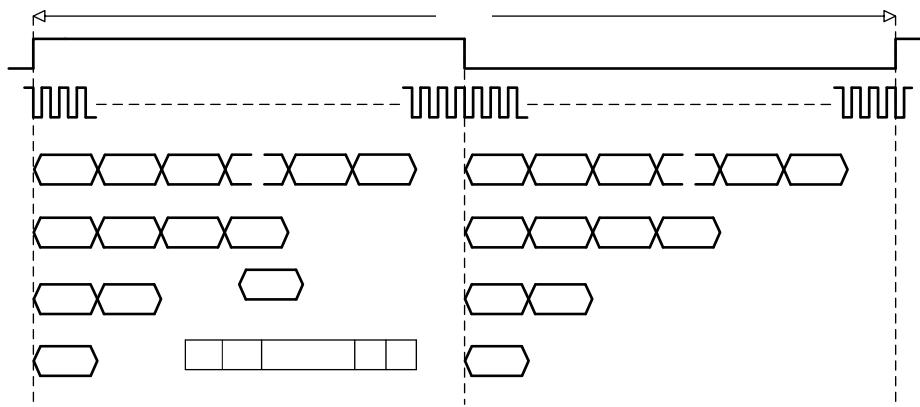
The I2S/PCM supports standard I2S mode, Left-justified I2S mode, Right-justified I2S mode, PCM mode, and TDM mode. The software can select the modes by setting [I2S/PCM CTL](#). The following figures describe the waveforms for SYNC, BCLK, DOUT, and DIN in different modes.

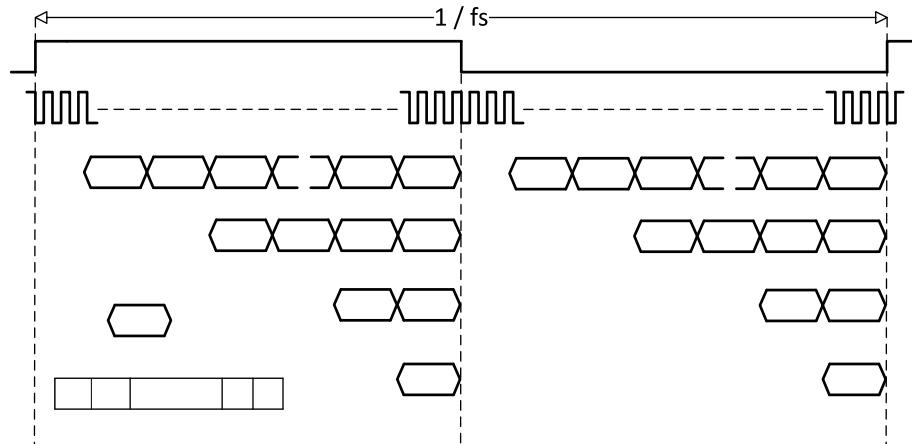
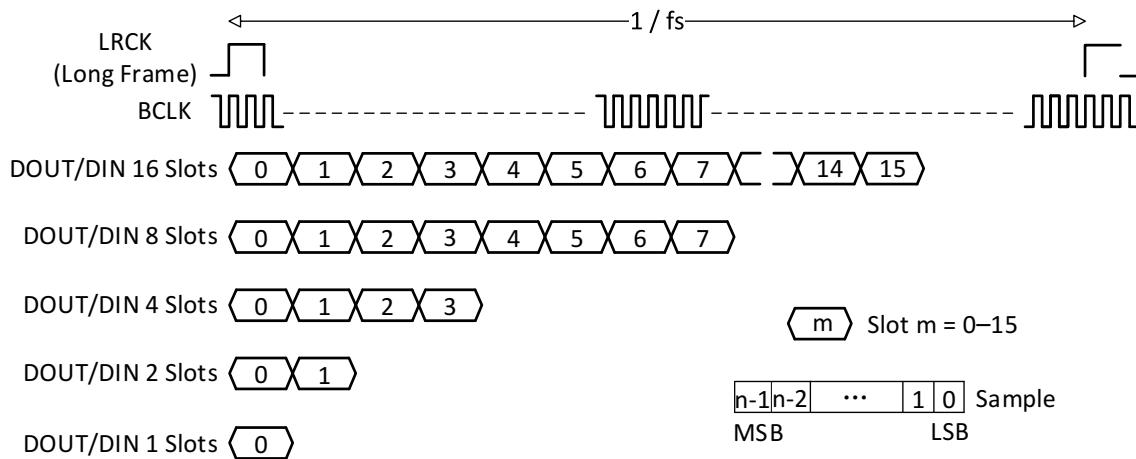
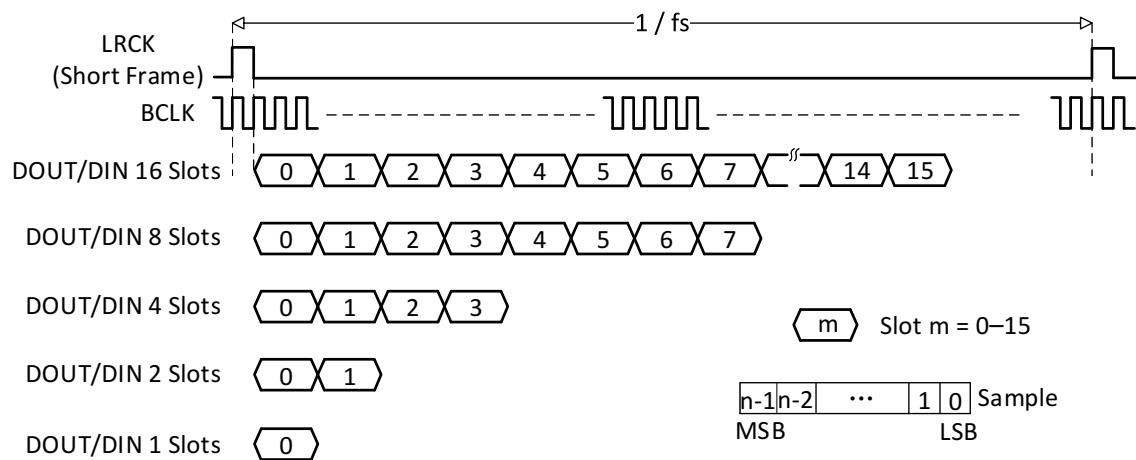
Each sampling period contains an LRCK. The low level of LRCK is the left channel corresponding to the even slots, and the high level is the right channel corresponding to the odd slots. Each slot is the sampling point of a mono channel. The sampling period can support the transmission of 2/4/8/16 slots. The BCLK corresponds to the serial data bit.

**Figure 7-3 I2S Standard Mode Timing**



**Figure 7-4 Left-Justified Mode Timing**



**Figure 7-5 Right-Justified Mode Timing**

**Figure 7-6 PCM Long Frame Mode Timing**

**Figure 7-7 PCM Short Frame Mode Timing**


#### 7.1.3.4 DIN Slot Mapping

The 4-wire DIN has 64 slots, each wire DIN has 16 slots. However, only 16 slots are valid and act as the RX channels.

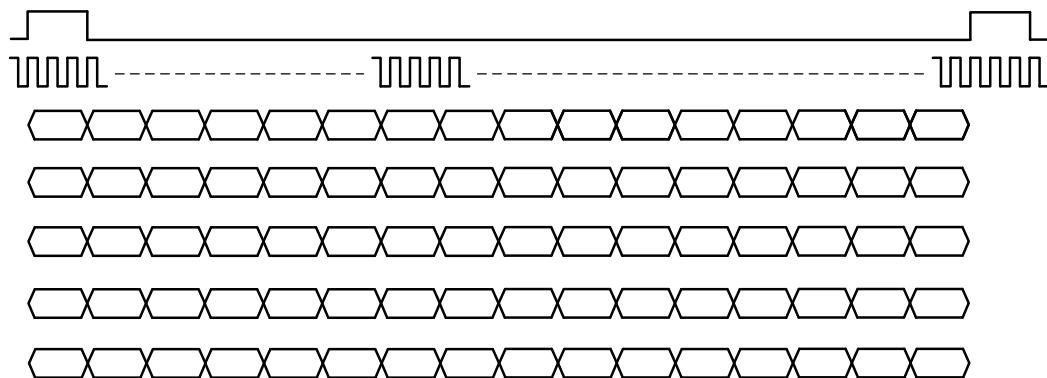
The following table shows the relationship between the slot id and encoder.

**Table 7-3 DIN Slot ID and Encoder**

DIN0 Slot ID	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
DIN1 Slot ID	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
DIN2 Slot ID	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
DIN3 Slot ID	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

There are 16 channels mapping configuration, each wire selects four slots for RX. The following figure shows the 16-channel mapping configuration.

**Figure 7-8 16-Channel Mapping Configuration**



#### 7.1.3.5 ASRC

The ASRC module supports sampling rate conversion between the up-sampling and down-sampling. The ASRC also supports sampling rate conversion between dual-channel audio data, and the size of the sampling data is up to 24 bits.

The ASRC module has the following features:

- Typical THD + N: -130 dB (Range: -125 dB to -139 dB)
- Supports sampling rate conversion between the up-sampling and down-sampling to implement the sampling rate conversion for stereo data
  - The up-sampling ratio ranges from 1 to 7.5x

- The down-sampling ratio ranges from 8 to 1x
- Supports sampling rate conversion between two identical frequencies
- Sampling rate for both the input and output range is from 8 kHz to 192 kHz and can be decimal
- Sampling rate can be configured manually or via adaptive generation
- The ASRC input is connected to I2S RX\_FIFO\_WDATA [31:8], and the input data is 24-bit MSB big-endian. For the input data that is less than 24 bits, use zeros to pad out the values at the low bits instead of high bits
- The ASRC needs some time to calculate the result. The output outsamplea/b will keep 0 during the calculation, and then change to the valid value when the result comes out

### Calculating the ASRC Latency

Calculate the ASRC up-sampling and down-sampling latency according to the following formulas.

$$\text{Upsampling Latency} = \text{Phase Delay} + \text{FIFO Delay} = 32 + 16 = 48 \text{ Input Sample Periods}$$

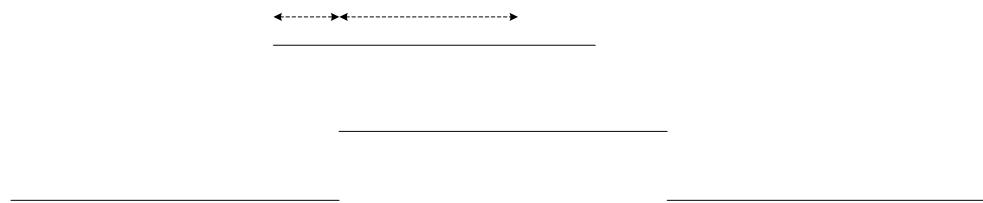
$$\text{Downsampling Latency} = \text{Phase Delay} + \text{FIFO Delay} = (32 * f_{sout}/f_{sin}) + 16 \text{ Input Sample Periods}$$

### ASRC Timing

The MCLK samples the input clock CLKIN to generate pulse signals.

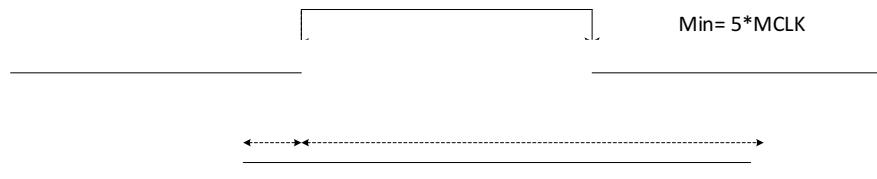
The following figure shows the timing requirements for the inputs.

**Figure 7-9 Timing Requirements for Inputs**



The following figure shows the timing requirements for the outputs.

**Figure 7-10 Timing Requirements for Outputs**



For the up-sampling,  $F_{MCLK} = F_{sout} * 1350$ .

For the down-sampling,  $F_{MCLK} = F_{sin} * 0.30 + F_{sout} * 295$ .

The following table provides the proper values of MCLK in MHz with different Fsin and Fsout in kHz.

**Table 7-4 Proper MCLK Values with Different Fsin and Fsout**

Fsin \ Fsout	32	44.1	48	88.2	96	144	192
32	45	60	65	120	130	195	260
44.1	55	60	65	120	130	195	260
48	60	65	65	120	130	195	260
88.2	105	105	110	120	130	195	260
96	110	115	115	125	130	195	260
144	160	165	165	175	180	195	260
192	210	215	215	225	230	245	260

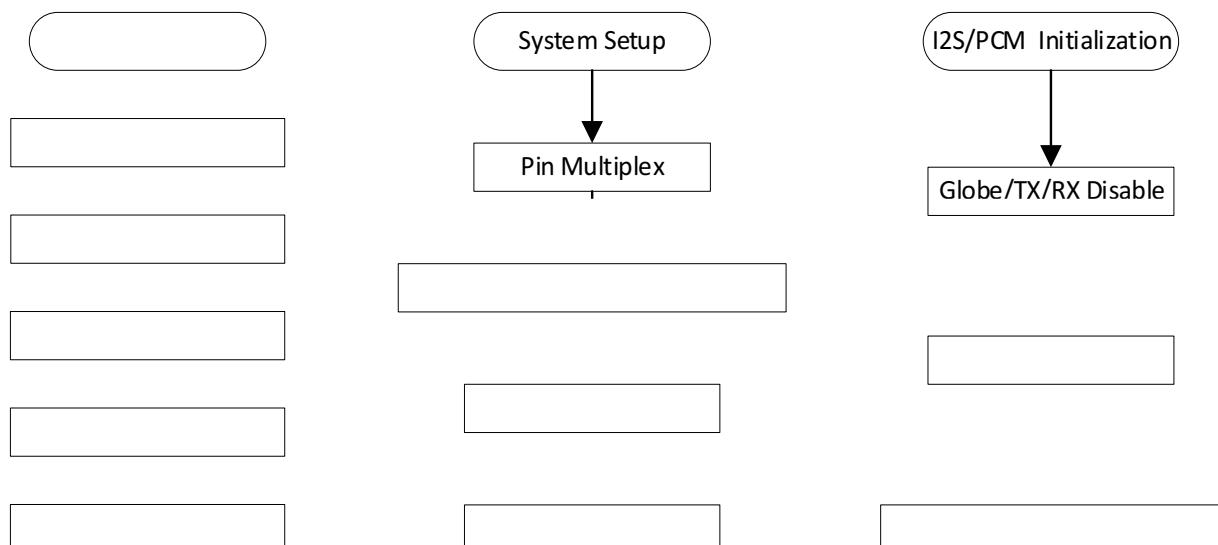
Note: The units for Fsin and Fsout are kHz and MCLK is MHz.

#### 7.1.3.6 Operation Modes

The software operation of the I2S/PCM is divided into five steps: system setup, I2S/PCM initialization, the channel setup, DMA setup, and Enable/Disable module.

The following figure shows the whole operation flow of I2S/PCM.

**Figure 7-11 I2S/PCM Operation Flow**



## 1. System Setup and I2S/PCM Initialization

The clock source for the I2S/PCM should be followed. Firstly, disable the PLL\_AUDIO through [PLL\\_AUDIOx Control Register](#)[PLL\_ENABLE] in the CCU. Secondly, set up the frequency of the PLL\_AUDIO in the [PLL\\_AUDIOx Control Register](#). After that, enable the I2S/PCM gating through the [I2S/PCMx CLK REG](#) when you checkout that the [PLL\\_AUDIOx Control Register](#)[LOCK] becomes to 1. At last, reset and enable the I2S/PCM bus gating by setting [I2S/PCM\\_BGR\\_REG](#).

After the system setup, the register of I2S/PCM can be setup. Firstly, initialize the I2S/PCM. You should close the Globe Enable bit ([I2S/PCM\\_CTL](#)[0]), Transmitter Block Enable bit ([I2S/PCM\\_CTL](#)[2]), and Receiver Block Enable bit ([I2S/PCM\\_CTL](#)[1]) by writing 0. After that, clear the TX/RX FIFO by writing 0 to the bit[25:24] of [I2S/PCM\\_FCTL](#). At last, you can clear the TX FIFO and RX FIFO counter by writing 0 to [I2S/PCM\\_TXCNT](#) and [I2S/PCM\\_RXCNT](#).

## 2. Channel Setup and DMA Setup

First, you can set up the I2S/PCM of master and slave. The configuration can be referred to the protocol of I2S/PCM. Then, you can set up the translation mode, the sample resolution, the wide of the slot, the channel slot number, and the trigger level, and so on. The setup of the register can be found in the specification.

The I2S/PCM supports two methods to transfer the data. The most common way is DMA, the setup of DMA can be found in the “[DMA](#)”. In this module, you just enable the DRQ.

## 3. Enable and Disable I2S/PCM

To enable the function, you can enable TX/RX by writing [I2S/PCM\\_CTL](#)[TXEN]/[I2S/PCM\\_CTL](#)[RXEN]. After that, enable I2S/PCM by writing 1 to [I2S/PCM\\_CTL](#)[Globe Enable]. Write 0 to the Globe Enable bit to disable I2S/PCM.

### 7.1.4 Programming Guidelines

#### 7.1.4.1 Application Example of Processing ASRC Input and Output Data

The following example shows a typical application of ASRC: the input data are 24-bit valid, and the output data are 32-bit data whose highest 24 bits are valid output and the lowest eight bits are padded out with zeros.

To implement the application, configure the sample resolution and slot width as 32 bits. Follow the steps below:

**Step 1** For the input register: 0x04 [6:4] sample\_res = 3'h7, 0x04 [2:0] slot\_width = 3'h7.

The format of the input data: 32'hXXXXXXXXX, where, bit[31] is the MSB and X is the valid data bit.

**Step 2** For the output register: 0x04 [6:4] sample\_res = 3'h7, 0x04 [2:0] slot\_width = 3'h7

The format of the output data: 32'hXXXXXX00, where, bit[31] is the MSB, X is the valid data bit, and bit[7:0] are the padded zeros.

#### 7.1.4.2 Converting the Sampling Rate with ASRC

Converting a 48 kHz sampling rate to 16 kHz is the most common scenario in actual applications. Follow the steps below to convert the sampling rate from 48 kHz to 16 kHz for the 32-bit data.

##### 1. Configure the PLL\_AUDIO Register

- a) Configure [PLL\\_AUDIO0\\_CTRL\\_REG](#)[31:0] as 0x8814AB01. That is, PLL\_AUDIO0 =  $24 * (171+1)/(1+1)/(1+0)/(1+20) = 98.286$  MHz. According to the relationship among the Fsin, Fsout, and MCLK, the MCLK should be greater than 60 MHz. In the simulation phase, the HOSC frequency is 25 MHz, so the output frequency of PLL\_AUDIO0 should be  $25 * (171+1)/(1+1)/(1+0)/(1+20) = 102.381$  MHz. In the IC test phase, configure the frequency of PLL\_AUDIO0 according to its actual output frequency.
- b) It is suggested that you configure the ASRC MCLK as an equal-duty-cycle signal. You can specify an odd number for bit[21:16] (PLL\_POST\_DIV\_P) of [PLL\\_AUDIO0\\_CTRL\\_REG](#) to get an equal-duty-cycle output clock of PLL\_AUDIO0.
- c) Configure bit[25:24] of [I2S/PCM2\\_ASRC\\_CLK\\_REG](#) as 0x00 to select the PLL\_AUDIO0(4X).

##### 2. Configure the I2S Registers

- a) Configure bit[7:4] (BCLKDIV) of [I2S/PCM\\_CLKD](#) as 4'h9, that is, the frequency of BCLK will be  $98.286$  MHz/ $32 = 3.072$  MHz.
- b) Configure bit[17:8] (LRCK\_PERIOD) of [I2S/PCM\\_FMT0](#) as 10'h1F. That is, the LRCK\_PERIOD width is configured as 32 BLCKs and can generate the ASRC CLKIN with a 48 kHz sampling rate.  
$$\left(\frac{3.072\text{MHz}}{32*2}\right) = 48\text{kHz}$$
- c) Configure bit[6:4] (Sample Resolution bits) of [I2S/PCM\\_FMT0](#) as 3'h7 to specify the sample resolution as 32-bit.
- d) Configure bit[2:0] (Slot Width bits) of [I2S/PCM\\_FMT0](#) as 3'h7 to specify the slot width as 32-bit.

##### 3. Configure the ASRC Registers

- a) Configure bit[16] (clock gate) of [MCLKCFG](#) as 1'h1 to open the clock gating.
- b) Configure bit[3:0] (division factor) of [MCLKCFG](#) as 1'h1 to specify the division factor as 1.
- c) Configure bit[20] (clock gate ) of [FSOUTCFG](#) as 1'h1 to open the clock gating.

- d) Configure bit[19:16] (clock select) of [FsoutCFG](#) as 4'h0 to select I2SASRC\_CLK as the clock source.
- e) Configure bit[7:4] (the first division factor) of [FsoutCFG](#) as 16'h13 to configure the first division factor as 128.
- f) Configure bit[3:0] (the second division factor) of [FsoutCFG](#) as 16'h10 to configure the second division factor as 48.
- g) Configure the ASRC ratio.

To configure the ASRC ratio manually, configure bit[31] ([Manual Configuration of ASRC Ratio Enable](#)) of ASRCMANCFG as 1'h1 to enable the manual configuration of ASRC ratio. Configure bit[25:0] of ASRCMANCFG as 26'h155555 to specify the ratio value as 0x155555. The calculation formula for the ratio value: Dec2Hex ( $F_{sout}/F_{sin} \times 2^{22}$ ). In this example,  $F_{sout}/F_{sin} = 16\text{ kHz}/48\text{ kHz} = 1/3$ , then the ratio is 0x155555.

To configure the ASRC ratio automatically, configure bit[31] ([Manual Configuration of ASRC Ratio Enable](#)) of ASRCMANCFG as 1'h0 to enable the automatic configuration of ASRC ratio. Then the system will automatically calculate the ratio value based on the MCLK, Fsout, and Fsin.

### 7.1.5 Register List

Module Name	Base Address
I2S/PCM1	0x02033000
I2S/PCM2	0x02034000

Register Name	Offset	Description
I2S/PCM_CTL	0x0000	I2S/PCM Control Register
I2S/PCM_FMT0	0x0004	I2S/PCM Format Register 0
I2S/PCM_FMT1	0x0008	I2S/PCM Format Register 1
I2S/PCMISTA	0x000C	I2S/PCM Interrupt Status Register
I2S/PCMRXFIFO	0x0010	I2S/PCM RXFIFO Register
I2S/PCM_FCTL	0x0014	I2S/PCM FIFO Control Register
I2S/PCM_FSTA	0x0018	I2S/PCM FIFO Status Register
I2S/PCM_INT	0x001C	I2S/PCM DMA & Interrupt Control Register
I2S/PCM_TXFIFO	0x0020	I2S/PCM TXFIFO Register
I2S/PCM_CLKD	0x0024	I2S/PCM Clock Divide Register
I2S/PCM_TXCNT	0x0028	I2S/PCM TX Sample Counter Register
I2S/PCM_RXCNT	0x002C	I2S/PCM RX Sample Counter Register

<b>Register Name</b>	<b>Offset</b>	<b>Description</b>
I2S/PCM_CHCFG	0x0030	I2S/PCM Channel Configuration Register
I2S/PCM_TX0CHSEL	0x0034	I2S/PCM TX0 Channel Select Register
I2S/PCM_TX1CHSEL	0x0038	I2S/PCM TX1 Channel Select Register
I2S/PCM_TX2CHSEL	0x003C	I2S/PCM TX2 Channel Select Register
I2S/PCM_TX3CHSEL	0x0040	I2S/PCM TX3 Channel Select Register
I2S/PCM_TX0CHMAP0	0x0044	I2S/PCM TX0 Channel Mapping Register0
I2S/PCM_TX0CHMAP1	0x0048	I2S/PCM TX0 Channel Mapping Register1
I2S/PCM_TX1CHMAP0	0x004C	I2S/PCM TX1 Channel Mapping Register0
I2S/PCM_TX1CHMAP1	0x0050	I2S/PCM TX1 Channel Mapping Register1
I2S/PCM_TX2CHMAP0	0x0054	I2S/PCM TX2 Channel Mapping Register0
I2S/PCM_TX2CHMAP1	0x0058	I2S/PCM TX2 Channel Mapping Register1
I2S/PCM_TX3CHMAP0	0x005C	I2S/PCM TX3 Channel Mapping Register0
I2S/PCM_TX3CHMAP1	0x0060	I2S/PCM TX3 Channel Mapping Register1
I2S/PCM_RXCHSEL	0x0064	I2S/PCM RX Channel Select Register
I2S/PCM_RXCHMAP0	0x0068	I2S/PCM RX Channel Mapping Register0
I2S/PCM_RXCHMAP1	0x006C	I2S/PCM RX Channel Mapping Register1
I2S/PCM_RXCHMAP2	0x0070	I2S/PCM RX Channel Mapping Register2
I2S/PCM_RXCHMAP3	0x0074	I2S/PCM RX Channel Mapping Register3
MCLKCFG	0x0080	ASRC MCLK Configuration Register
FsoutCFG	0x0084	ASRC Out Sample Rate Configuration Register
FsinEXTCFG	0x0088	ASRC Input Sample Pulse Extend Configuration Register
ASRCCFG	0x008C	ASRC Enable Register
ASRCMANCFG	0x0090	ASRC Manual Ratio Configuration Register
ASRCRATIOSTAT	0x0094	ASRC Status Register
ASRCFIFOSTAT	0x0098	ASRC FIFO Level Status Register
ASRCMBISTCFG	0x009C	ASRC MBIST Test Configuration Register
ASRCMBISTSTAT	0x00A0	ASRC MBIST Test Status Register

### 7.1.6 Register Description

#### 7.1.6.1 0x0000 I2S/PCM Control Register (Default Value: 0x0006\_0000)

Offset: 0x0000			Register Name: I2S/PCM_CTL
Bit	Read/Write	Default/Hex	Description
31:22	/	/	/
21	R/W	0x0	<b>RX_SYNC_EN_START</b> The bit takes effect only when RX_SYNC_EN is set to 1. I2S1/I2S2 Synchronize Enable Start. 0: Disabled 1: Enabled
20	R/W	0x0	<b>RX_SYNC_EN</b> I2S RX Synchronize Enable 0: Disabled 1: Enabled
19	/	/	/
18	R/W	0x1	<b>BCLK_OUT</b> Bit Clock Direction Select 0: Input 1: Output
17	R/W	0x1	<b>LRCK_OUT</b> LRCK Direction Select 0: Input 1: Output
16:12	/	/	/
11	R/W	0x0	<b>DOUT3_EN</b> Data3 Output Enable 0: Disabled, Hi-Z State 1: Enabled
10	R/W	0x0	<b>DOUT2_EN</b> Data2 Output Enable 0: Disabled, Hi-Z State 1: Enabled
9	R/W	0x0	<b>DOUT1_EN</b> Data1 Output Enable 0: Disabled, Hi-Z State 1: Enabled

Offset: 0x0000			Register Name: I2S/PCM_CTL
Bit	Read/Write	Default/Hex	Description
8	R/W	0x0	DOUT0_EN Data0 Output Enable 0: Disabled, Hi-Z State 1: Enabled
7	/	/	/
6	R/W	0x0	OUT_MUTE Data Output Mute Enable 0: Normal Transfer 1: Force DOUT to output 0
5:4	R/W	0x0	MODE_SEL Mode Selection 00: PCM Mode (offset 0: Long Frame; offset 1: Short Frame) 01: Left Mode (offset 0: LJ Mode; offset 1: I2S Mode) 10: Right-Justified Mode 11: Reserved
3	R/W	0x0	LOOP Loopback Test 0: Normal Mode 1: Loopback Test When set to '1', the bit indicates that the DOUT is connected to the DIN.
2	R/W	0x0	TXEN Transmitter Block Enable 0: Disabled 1: Enabled
1	R/W	0x0	RXEN Receiver Block Enable 0: Disabled 1: Enabled
0	R/W	0x0	GEN Globe Enable 0: Disabled 1: Enabled

**7.1.6.2 0x0004 I2S/PCM Format Register 0 (Default Value: 0x0000\_0033)**

Offset: 0x0004			Register Name: I2S/PCM_FMT0
Bit	Read/Write	Default/Hex	Description
31	/	/	/
30	R/W	0x0	LRCK_WIDTH LRCK Width (only applies to the PCM mode) 0: LRCK = 1 BCLK Width (Short Frame) 1: LRCK = 2 BCLK Width (Long Frame)
29:20	/	/	/
19	R/W	0x0	LRCK_POLARITY In I2S/Left-Justified/Right-Justified mode: 0: Left Channel when LRCK is low. 1: Left channel when LRCK is high. In PCM mode: 0: PCM LRCK asserted at the negative edge. 1: PCM LRCK asserted at the positive edge.
18	/	/	/
17:8	R/W	0x0	LRCK_PERIOD It is used to program the number of BCLKs per channel of the sample frame. This value is interpreted as follows. PCM mode: Number of BCLKs within (Left + Right) channel width. I2S/Left-Justified/Right-Justified mode: Number of BCLKs within each channel width (Left or Right). For example: N = 7: 8 BCLKs width ... N = 1023: 1024 BCLKs width
7	R/W	0x0	BCLK_POLARITY 0: Normal mode, DOUT drives data at negative edge 1: Invert mode, DOUT drives data at positive edge

Offset: 0x0004			Register Name: I2S/PCM_FMT0
Bit	Read/Write	Default/Hex	Description
6:4	R/W	0x3	<p>SR</p> <p>Sample Resolution</p> <p>000: Reserved</p> <p>001: 8-bit</p> <p>010: 12-bit</p> <p>011: 16-bit</p> <p>100: 20-bit</p> <p>101: 24-bit</p> <p>110: 28-bit</p> <p>111: 32-bit</p>
3	R/W	0x0	<p>EDGE_TRANSFER</p> <p>Edge Transfer</p> <p>0: DOUT drives data and DIN sample data at the different BCLK edge</p> <p>1: DOUT drives data and DIN sample data at the same BCLK edge</p> <p>BCLK_POLARITY = 0, EDGE_TRANSFER = 0, DIN sample data at positive edge;</p> <p>BCLK_POLARITY = 0, EDGE_TRANSFER = 1, DIN sample data at negative edge;</p> <p>BCLK_POLARITY = 1, EDGE_TRANSFER = 0, DIN sample data at negative edge;</p> <p>BCLK_POLARITY = 1, EDGE_TRANSFER = 1, DIN sample data at positive edge.</p>
2:0	R/W	0x3	<p>SW</p> <p>Slot Width Select</p> <p>000: Reserved</p> <p>001: 8-bit</p> <p>010: 12-bit</p> <p>011: 16-bit</p> <p>100: 20-bit</p> <p>101: 24-bit</p> <p>110: 28-bit</p> <p>111: 32-bit</p>

### 7.1.6.3 0x0008 I2S/PCM Format Register 1 (Default Value: 0x0000\_0030)

Offset: 0x0008			Register Name: I2S/PCM_FMT1
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R/W	0x0	RX MLS MSB/LSB First Select 0: MSB First 1: LSB First
6	R/W	0x0	TX MLS MSB/LSB First Select 0: MSB First 1: LSB First
5:4	R/W	0x3	SEXT Sign Extend in Slot [Sample Resolution < Slot Width] 00: Zeros or audio gain padding at LSB position 01: Sign extension at MSB position 10: Reserved 11: Transfer 0 after each sample in each Slot
3:2	R/W	0x0	RX_PDM PCM Data Mode 00: Linear PCM 01: Reserved 10: 8-bit u-law 11: 8-bit A-law
1:0	R/W	0x0	TX_PDM PCM Data Mode 00: Linear PCM 01: Reserved 10: 8-bit u-law 11: 8-bit A-law

### 7.1.6.4 0x000C I2S/PCM Interrupt Status Register (Default Value: 0x0000\_0010)

Offset: 0x000C			Register Name: I2S/PCMISTA
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/

Offset: 0x000C			Register Name: I2S/PCMISTA
Bit	Read/Write	Default/Hex	Description
6	R/W1C	0x0	<p>TXU_INT TXFIFO Underrun Pending Interrupt 0: No pending interrupt 1: TXFIFO underrun pending interrupt Write '1' to clear this interrupt.</p>
5	R/W1C	0x0	<p>TXO_INT TXFIFO Overrun Pending Interrupt 0: No pending interrupt 1: TXFIFO overrun pending interrupt Write '1' to clear this interrupt.</p>
4	R	0x1	<p>TXE_INT TXFIFO Empty Pending Interrupt 0: No pending IRQ 1: TXFIFO empty pending interrupt when data in TXFIFO are less than TX trigger level</p>
3	/	/	/
2	R/W1C	0x0	<p>RXU_INT RXFIFO Underrun Pending Interrupt 0: No pending interrupt 1: RXFIFO underrun pending interrupt Write '1' to clear this interrupt.</p>
1	R/W1C	0x0	<p>RXO_INT RXFIFO Overrun Pending Interrupt 0: No pending IRQ 1: RXFIFO overrun pending IRQ Write '1' to clear this interrupt.</p>
0	R/W	0x0	<p>RXA_INT RXFIFO Data Available Pending Interrupt 0: No pending IRQ 1: Data available pending IRQ when data in RXFIFO are more than RX trigger level</p>

**7.1.6.5 0x0010 I2S/PCM RXFIFO Register (Default Value: 0x0000\_0000)**

Offset: 0x0010			Register Name: I2S/PCM_RXFIFO
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	RX_DATA RX Sample The host can get one sample by reading this register. The left channel sample data is first and then the right channel sample.

**7.1.6.6 0x0014 I2S/PCM FIFO Control Register (Default Value: 0x0004\_00F0)**

Offset: 0x0014			Register Name: I2S/PCM_FCTL
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	HUB_EN Audio Hub Enable The bit takes effect only when TXEN is set to 1. I2S1/I2S2/OWA TXFIFO Hub Enable. 0: Disabled 1: Enabled
30:26	/	/	/
25	R/WAC	0x0	FTX Write '1' to flush TXFIFO, self clear to '0'.
24	R/WAC	0x0	FRX Write '1' to flush RXFIFO, self clear to '0'.
23:19	/	/	/
18:12	R/W	0x40	TXTL TXFIFO Empty Trigger Level Interrupt and DMA request trigger level for TXFIFO normal condition. Trigger Level = TXTL
11:10	/	/	/
9:4	R/W	0xF	RXTL RXFIFO Trigger Level Interrupt and DMA request trigger level for RXFIFO normal condition. Trigger Level = RXTL + 1
3	/	/	/

Offset: 0x0014			Register Name: I2S/PCM_FCTL
Bit	Read/Write	Default/Hex	Description
2	R/W	0x0	<p>TXIM TXFIFO Input Mode (Mode 0, 1) 0: Valid data at the MSB of TXFIFO register 1: Valid data at the LSB of TXFIFO register Example for 20-bit transmitted audio sample: Mode 0: TXFIFO[31:0] = {APB_WDATA[31:12], 12'h0} Mode 1: TXFIFO[31:0] = {APB_WDATA[19:0], 12'h0}</p>
1:0	R/W	0x0	<p>RXOM RXFIFO Output Mode (Mode 0, 1, 2, 3) 00: Expanding '0' at LSB of RXFIFO register 01: Expanding received sample sign bit at MSB of RXFIFO register 10: Truncating received samples at high half-word of RXFIFO register and low half-word of RXFIFO register is filled by '0' 11: Truncating received samples at low half-word of RXFIFO register and high half-word of RXFIFO register is expanded by its sign bit Example for 20-bit received audio sample: Mode 0: APB_RDATA[31:0] = {RXFIFO[31:12], 12'h0} Mode 1: APB_RDATA[31:0] = {12{RXFIFO[31]}, RXFIFO[31:12]} Mode 2: APB_RDATA [31:0] = {RXFIFO[31:16], 16'h0} Mode 3: APB_RDATA[31:0] = {16{RXFIFO[31]}, RXFIFO[31:16]}</p>

#### 7.1.6.7 0x0018 I2S/PCM FIFO Status Register (Default Value: 0x1080\_0080)

Offset: 0x0018			Register Name: I2S/PCM_FSTA
Bit	Read/Write	Default/Hex	Description
31:29	/	/	/
28	R	0x1	<p>TXE TXFIFO Empty 0: No room for new sample in TXFIFO 1: More than one room for new sample in TXFIFO (&gt;= 1 Word)</p>
27:24	/	/	/
23:16	R	0x80	<p>TXE_CNT TXFIFO Empty Space Word Counter</p>

Offset: 0x0018			Register Name: I2S/PCM_FSTA
Bit	Read/Write	Default/Hex	Description
15:9	/	/	/
8	R	0x0	RXA RXFIFO Available 0: No available data in RXFIFO 1: More than one sample in RXFIFO (>= 1 Word)
7	R	0x1	PLACE HOLDER NO Meaning
6:0	R	0x0	RXA_CNT RXFIFO available sample word counter

#### 7.1.6.8 0x001C I2S/PCM DMA & Interrupt Control Register (Default Value: 0x0000\_0000)

Offset: 0x001C			Register Name: I2S/PCM_INT
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R/W	0x0	TX_DRQ TXFIFO Empty DRQ Enable 0: Disabled 1: Enabled
6	R/W	0x0	TXUI_EN TXFIFO Underrun Interrupt Enable 0: Disabled 1: Enabled
5	R/W	0x0	TXOI_EN TXFIFO Overrun Interrupt Enable 0: Disabled 1: Enabled When set to '1', an interrupt happens when writing new audio data if TXFIFO is full.
4	R/W	0x0	TXEI_EN TXFIFO Empty Interrupt Enable 0: Disabled 1: Enabled

Offset: 0x001C			Register Name: I2S/PCM_INT
Bit	Read/Write	Default/Hex	Description
3	R/W	0x0	<p>RX_DRQ RXFIFO Data Available DRQ Enable 0: Disabled 1: Enabled When set to '1', RXFIFO DMA request line is asserted if data is available in RXFIFO.</p>
2	R/W	0x0	<p>RXUI_EN RXFIFO Underrun Interrupt Enable 0: Disabled 1: Enabled</p>
1	R/W	0x0	<p>RXOI_EN RXFIFO Overrun Interrupt Enable 0: Disabled 1: Enabled</p>
0	R/W	0x0	<p>RXAI_EN RXFIFO Data Available Interrupt Enable 0: Disabled 1: Enabled</p>

#### 7.1.6.9 0x0020 I2S/PCM TXFIFO Register (Default Value: 0x0000\_0000)

Offset: 0x0020			Register Name: I2S/PCM_TXFIFO
Bit	Read/Write	Default/Hex	Description
31:0	W	0x0	<p>TX_DATA TX Sample Transmitting left, right channel sample data should be written to this register one by one. The left channel sample data is first and then the right channel sample.</p>

#### 7.1.6.10 0x0024 I2S/PCM Clock Divide Register (Default Value: 0x0000\_0000)

Offset: 0x0024			Register Name: I2S/PCM_CLKD
Bit	Read/Write	Default/Hex	Description
31:9	/	/	/

Offset: 0x0024			Register Name: I2S/PCM_CLKD
Bit	Read/Write	Default/Hex	Description
8	R/W	0x0	<p>MCLKO_EN</p> <p>0: Disable MCLK Output</p> <p>1: Enable MCLK Output</p> <p><b>Note: Whether in slave or master mode, when this bit is set to '1', MCLK should be output.</b></p>
7:4	R/W	0x0	<p>BCLKDIV</p> <p>BCLK Divide Ratio from PLL_AUDIO</p> <p>0000: Reserved</p> <p>0001: Divide by 1</p> <p>0010: Divide by 2</p> <p>0011: Divide by 4</p> <p>0100: Divide by 6</p> <p>0101: Divide by 8</p> <p>0110: Divide by 12</p> <p>0111: Divide by 16</p> <p>1000: Divide by 24</p> <p>1001: Divide by 32</p> <p>1010: Divide by 48</p> <p>1011: Divide by 64</p> <p>1100: Divide by 96</p> <p>1101: Divide by 128</p> <p>1110: Divide by 176</p> <p>1111: Divide by 192</p>

Offset: 0x0024			Register Name: I2S/PCM_CLKD
Bit	Read/Write	Default/Hex	Description
3:0	R/W	0x0	<p>MCLKDIV</p> <p>MCLK Divide Ratio from PLL_AUDIO</p> <p>0000: Reserved</p> <p>0001: Divide by 1</p> <p>0010: Divide by 2</p> <p>0011: Divide by 4</p> <p>0100: Divide by 6</p> <p>0101: Divide by 8</p> <p>0110: Divide by 12</p> <p>0111: Divide by 16</p> <p>1000: Divide by 24</p> <p>1001: Divide by 32</p> <p>1010: Divide by 48</p> <p>1011: Divide by 64</p> <p>1100: Divide by 96</p> <p>1101: Divide by 128</p> <p>1110: Divide by 176</p> <p>1111: Divide by 192</p>

#### 7.1.6.11 0x0028 I2S/PCM TX Sample Counter Register (Default Value: 0x0000\_0000)

Offset: 0x0028			Register Name: I2S/PCM_TXCNT
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<p>TX_CNT</p> <p>TX Sample Counter</p> <p>The audio sample number of sending into TXFIFO. When one sample is put into TXFIFO by DMA or by host IO, the TX sample counter register increases by one. The TX sample counter register can be set to any initial value at any time. After being updated by the initial value, the counter register should count on the base of this initial value.</p>

**7.1.6.12 0x002C I2S/PCM RX Sample Counter Register (Default Value: 0x0000\_0000)**

Offset: 0x002C			Register Name: I2S/PCM_RXCNT
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<p>RX_CNT RX Sample Counter</p> <p>The audio sample number of writing into RXFIFO. When one sample is written by Digital Audio Engine, the RX sample counter register increases by one. The RX sample counter register can be set to any initial value at any time. After being updated by the initial value, the counter register should count on the base of this initial value.</p>

**7.1.6.13 0x0030 I2S/PCM Channel Configuration Register (Default Value: 0x0000\_0000)**

Offset: 0x0030			Register Name: I2S/PCM_CHCFG
Bit	Read/Write	Default/Hex	Description
31:10	/	/	/
9	R/W	0x0	<p>TX_SLOT_HIZ</p> <p>0: Normal mode for the last half-cycle of BCLK in the slot</p> <p>1: Turn to Hi-Z state for the last half-cycle of BCLK in the slot</p>
8	R/W	0x0	<p>TX_STATE</p> <p>0: Transfer level 0 in non-transferring slot</p> <p>1: Turn to Hi-Z State (TDM) in non-transferring slot</p>
7:4	R/W	0x0	<p>RX_SLOT_NUM</p> <p>RX Channel/Slot number between CPU/DMA and RXFIFO</p> <p>0000: 1 channel or slot</p> <p>...</p> <p>0111: 8 channels or slots</p> <p>1000: 9 channels or slots</p> <p>...</p> <p>1111: 16 channels or slots</p>

Offset: 0x0030			Register Name: I2S/PCM_CHCFG
Bit	Read/Write	Default/Hex	Description
3:0	R/W	0x0	<p>TX_SLOT_NUM</p> <p>TX Channel/Slot number between CPU/DMA and TXFIFO</p> <p>0000: 1 channel or slot</p> <p>...</p> <p>0111: 8 channels or slots</p> <p>1000: 9 channels or slots</p> <p>...</p> <p>1111: 16 channels or slots</p>

#### 7.1.6.14 0x0034 I2S/PCM TX0 Channel Select Register (Default Value: 0x0000\_0000)

Offset: 0x0034			Register Name: I2S/PCM_TX0CHSEL
Bit	Read/Write	Default/Hex	Description
31:22	/	/	/
21:20	R/W	0x0	<p>TX0_OFFSET</p> <p>TX0 Offset Tune (TX0 Data offset to LRCK)</p> <p>0: No offset</p> <p>n: Data is offset by n BCLKs to LRCK</p>
19:16	R/W	0x0	<p>TX0_CHSEL</p> <p>TX0 Channel (Slot) Number Select for Each Output</p> <p>0000: 1 channel or slot</p> <p>...</p> <p>0111: 8 channels or slots</p> <p>1000: 9 channels or slots</p> <p>...</p> <p>1111: 16 channels or slots</p>
15:0	R/W	0x0	<p>TX0_CHEN</p> <p>TX0 Channel (Slot) Enable</p> <p>The bit[15:0] refer to Slot [15:0]. When one or more slots are disabled, the affected slots are set to the disable state.</p> <p>0: Disabled</p> <p>1: Enabled</p>

**7.1.6.15 0x0038 I2S/PCM TX1 Channel Select Register (Default Value: 0x0000\_0000)**

Offset: 0x0038			Register Name: I2S/PCM_TX1CHSEL
Bit	Read/Write	Default/Hex	Description
31:22	/	/	/
21:20	R/W	0x0	<p>TX1_OFFSET TX1 Offset Tune (TX1 Data offset to LRCK) 0: No offset n: Data is offset by n BCLKs to LRCK</p>
19:16	R/W	0x0	<p>TX1_CHSEL TX1 Channel (Slot) Number Select for Each Output 0000: 1 channel or slot ... 0111: 8 channels or slots 1000: 9 channels or slots ... 1111: 16 channels or slots</p>
15:0	R/W	0x0	<p>TX1_CHEN TX1 Channel (Slot) Enable The bit[15:0] refers to Slot [15:0]. When one or more slots are disabled, the affected slots are set to the disable state. 0: Disabled 1: Enabled</p>

**7.1.6.16 0x003C I2S/PCM TX2 Channel Select Register (Default Value: 0x0000\_0000)**

Offset: 0x003C			Register Name: I2S/PCM_TX2CHSEL
Bit	Read/Write	Default/Hex	Description
31:22	/	/	/
21:20	R/W	0x0	<p>TX2_OFFSET TX2 Offset Tune (TX2 Data offset to LRCK) 0: No offset n: Data is offset by n BCLKs to LRCK</p>

Offset: 0x003C			Register Name: I2S/PCM_TX2CHSEL
Bit	Read/Write	Default/Hex	Description
19:16	R/W	0x0	<p>TX2_CHSEL</p> <p>TX2 Channel (Slot) Number Select for Each Output</p> <p>0000: 1 channel or slot</p> <p>...</p> <p>0111: 8 channels or slots</p> <p>1000: 9 channels or slots</p> <p>...</p> <p>1111: 16 channels or slots</p>
15:0	R/W	0x0	<p>TX2_CHEN</p> <p>TX2 Channel (Slot) Enable</p> <p>The bit[15:0] refers to Slot [15:0]. When one or more slots are disabled, the affected slots are set to the disable state.</p> <p>0: Disabled</p> <p>1: Enabled</p>

#### 7.1.6.17 0x0040 I2S/PCM TX3 Channel Select Register (Default Value: 0x0000\_0000)

Offset: 0x0040			Register Name: I2S/PCM_TX3CHSEL
Bit	Read/Write	Default/Hex	Description
31:22	/	/	/
21:20	R/W	0x0	<p>TX3_OFFSET</p> <p>TX3 Offset Tune (TX3 Data offset to LRCK)</p> <p>0: No offset</p> <p>n: Data is offset by n BCLKs to LRCK</p>
19:16	R/W	0x0	<p>TX3_CHSEL</p> <p>TX3 Channel (Slot) Number Select for Each Output</p> <p>0000: 1 channel or slot</p> <p>...</p> <p>0111: 8 channels or slots</p> <p>1000: 9 channels or slots</p> <p>...</p> <p>1111: 16 channels or slots</p>

Offset: 0x0040			Register Name: I2S/PCM_TX3CHSEL
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0x0	<p>TX3_CHEN</p> <p>TX3 Channel (Slot) Enable</p> <p>The bit[15:0] refers to Slot [15:0]. When one or more slots are disabled, the affected slots are set to the disable state.</p> <p>0: Disabled</p> <p>1: Enabled</p>

#### 7.1.6.18 0x0044 I2S/PCM TX0 Channel Mapping0 Register 0 (Default Value: 0x0000\_0000)

Offset: 0x0044			Register Name: I2S/PCM_TX0CHMAP0
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	<p>TX0_CH15_MAP</p> <p>TX0 Channel 15 Mapping</p> <p>0000: The first sample</p> <p>...</p> <p>0111: The eighth sample</p> <p>1000: The ninth sample</p> <p>...</p> <p>1111: The sixteenth sample</p>
27:24	R/W	0x0	<p>TX0_CH14_MAP</p> <p>TX0 Channel 14 Mapping</p> <p>0000: The first sample</p> <p>...</p> <p>0111: The eighth sample</p> <p>1000: The ninth sample</p> <p>...</p> <p>1111: The sixteenth sample</p>
23:20	R/W	0x0	<p>TX0_CH13_MAP</p> <p>TX0 Channel 13 Mapping</p> <p>0000: The first sample</p> <p>...</p> <p>0111: The eighth sample</p> <p>1000: The ninth sample</p> <p>...</p> <p>1111: The sixteenth sample</p>

Offset: 0x0044			Register Name: I2S/PCM_TX0CHMAP0
Bit	Read/Write	Default/Hex	Description
19:16	R/W	0x0	TX0_CH12_MAP TX0 Channel 12 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
15:12	R/W	0x0	TX0_CH11_MAP TX0 Channel 11 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
11:8	R/W	0x0	TX0_CH10_MAP TX0 Channel 10 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
7:4	R/W	0x0	TX0_CH9_MAP TX0 Channel 9 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

Offset: 0x0044			Register Name: I2S/PCM_TX0CHMAP0
Bit	Read/Write	Default/Hex	Description
3:0	R/W	0x0	TX0_CH8_MAP TX0 Channel 8 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

#### 7.1.6.19 0x0048 I2S/PCM TX0 Channel Mapping1 Register 1 (Default Value: 0x0000\_0000)

Offset: 0x0048			Register Name: I2S/PCM_TX0CHMAP1
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	TX0_CH7_MAP TX0 Channel 7 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
27:24	R/W	0x0	TX0_CH6_MAP TX0 Channel 6 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

Offset: 0x0048			Register Name: I2S/PCM_TX0CHMAP1
Bit	Read/Write	Default/Hex	Description
23:20	R/W	0x0	<p>TX0_CH5_MAP TX0 Channel 5 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample</p>
19:16	R/W	0x0	<p>TX0_CH4_MAP TX0 Channel 4 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample</p>
15:12	R/W	0x0	<p>TX0_CH3_MAP TX0 Channel 3 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample</p>
11:8	R/W	0x0	<p>TX0_CH2_MAP TX0 Channel 2 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample</p>

Offset: 0x0048			Register Name: I2S/PCM_TX0CHMAP1
Bit	Read/Write	Default/Hex	Description
7:4	R/W	0x0	TX0_CH1_MAP TX0 Channel 1 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
3:0	R/W	0x0	TX0_CH0_MAP TX0 Channel 0 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

#### 7.1.6.20 0x004C I2S/PCM TX1 Channel Mapping0 Register 0 (Default Value: 0x0000\_0000)

Offset: 0x004C			Register Name: I2S/PCM_TX1CHMAP0
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	TX1_CH15_MAP TX1 Channel 15 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

Offset: 0x004C			Register Name: I2S/PCM_TX1CHMAP0
Bit	Read/Write	Default/Hex	Description
27:24	R/W	0x0	TX1_CH14_MAP TX1 Channel 14 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
23:20	R/W	0x0	TX1_CH13_MAP TX1 Channel 13 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
19:16	R/W	0x0	TX1_CH12_MAP TX1 Channel 12 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
15:12	R/W	0x0	TX1_CH11_MAP TX1 Channel 11 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

Offset: 0x004C			Register Name: I2S/PCM_TX1CHMAP0
Bit	Read/Write	Default/Hex	Description
11:8	R/W	0x0	TX1_CH10_MAP TX1 Channel 10 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
7:4	R/W	0x0	TX1_CH9_MAP TX1 Channel 9 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
3:0	R/W	0x0	TX1_CH8_MAP TX1 Channel 8 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

**7.1.6.21 0x0050 I2S/PCM TX1 Channel Mapping1 Register 1 (Default Value: 0x0000\_0000)**

Offset: 0x0050			Register Name: I2S/PCM_TX1CHMAP1
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	TX1_CH7_MAP TX1 Channel 7 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
27:24	R/W	0x0	TX1_CH6_MAP TX1 Channel 6 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
23:20	R/W	0x0	TX1_CH5_MAP TX1 Channel 5 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
19:16	R/W	0x0	TX1_CH4_MAP TX1 Channel 4 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

Offset: 0x0050			Register Name: I2S/PCM_TX1CHMAP1
Bit	Read/Write	Default/Hex	Description
15:12	R/W	0x0	<p>TX1_CH3_MAP TX1 Channel 3 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample</p>
11:8	R/W	0x0	<p>TX1_CH2_MAP TX1 Channel 2 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample</p>
7:4	R/W	0x0	<p>TX1_CH1_MAP TX1 Channel 1 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample</p>
3:0	R/W	0x0	<p>TX1_CHO_MAP TX1 Channel 0 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample</p>

**7.1.6.22 0x0054 I2S/PCM TX2 Channel Mapping0 Register 0 (Default Value: 0x0000\_0000)**

Offset: 0x0054			Register Name: I2S/PCM_TX2CHMAP0
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	TX2_CH15_MAP TX2 Channel 15 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
27:24	R/W	0x0	TX2_CH14_MAP TX2 Channel 14 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
23:20	R/W	0x0	TX2_CH13_MAP TX2 Channel 13 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
19:16	R/W	0x0	TX2_CH12_MAP TX2 Channel 12 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

Offset: 0x0054			Register Name: I2S/PCM_TX2CHMAP0
Bit	Read/Write	Default/Hex	Description
15:12	R/W	0x0	TX2_CH11_MAP TX2 Channel 11 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
11:8	R/W	0x0	TX2_CH10_MAP TX2 Channel 10 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
7:4	R/W	0x0	TX2_CH9_MAP TX2 Channel 9 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
3:0	R/W	0x0	TX2_CH8_MAP TX2 Channel 8 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

**7.1.6.23 0x0058 I2S/PCM TX2 Channel Mapping1 Register 1 (Default Value: 0x0000\_0000)**

Offset: 0x0058			Register Name: I2S/PCM_TX2CHMAP1
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	TX2_CH7_MAP TX2 Channel 7 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
27:24	R/W	0x0	TX2_CH6_MAP TX2 Channel 6 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
23:20	R/W	0x0	TX2_CH5_MAP TX2 Channel 5 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
19:16	R/W	0x0	TX2_CH4_MAP TX2 Channel 4 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

Offset: 0x0058			Register Name: I2S/PCM_TX2CHMAP1
Bit	Read/Write	Default/Hex	Description
15:12	R/W	0x0	TX2_CH3_MAP TX2 Channel 3 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
11:8	R/W	0x0	TX2_CH2_MAP TX2 Channel 2 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
7:4	R/W	0x0	TX2_CH1_MAP TX2 Channel 1 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
3:0	R/W	0x0	TX2_CH0_MAP TX2 Channel 0 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

**7.1.6.24 0x005C I2S/PCM TX3 Channel Mapping0 Register 0 (Default Value: 0x0000\_0000)**

Offset: 0x005C			Register Name: I2S/PCM_TX3CHMAP0
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	TX3_CH15_MAP TX3 Channel 15 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
27:24	R/W	0x0	TX3_CH14_MAP TX3 Channel 14 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
23:20	R/W	0x0	TX3_CH13_MAP TX3 Channel 13 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
19:16	R/W	0x0	TX3_CH12_MAP TX3 Channel 12 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

Offset: 0x005C			Register Name: I2S/PCM_TX3CHMAP0
Bit	Read/Write	Default/Hex	Description
15:12	R/W	0x0	TX3_CH11_MAP TX3 Channel 11 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
11:8	R/W	0x0	TX3_CH10_MAP TX3 Channel 10 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
7:4	R/W	0x0	TX3_CH9_MAP TX3 Channel 9 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
3:0	R/W	0x0	TX3_CH8_MAP TX3 Channel 8 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

**7.1.6.25 0x0060 I2S/PCM TX3 Channel Mapping1 Register 1 (Default Value: 0x0000\_0000)**

Offset: 0x0060			Register Name: I2S/PCM_TX3CHMAP1
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	TX3_CH7_MAP TX3 Channel 7 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
27:24	R/W	0x0	TX3_CH6_MAP TX3 Channel 6 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
23:20	R/W	0x0	TX3_CH5_MAP TX3 Channel 5 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
19:16	R/W	0x0	TX3_CH4_MAP TX3 Channel 4 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

Offset: 0x0060			Register Name: I2S/PCM_TX3CHMAP1
Bit	Read/Write	Default/Hex	Description
15:12	R/W	0x0	TX3_CH3_MAP TX3 Channel 3 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
11:8	R/W	0x0	TX3_CH2_MAP TX3 Channel 2 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
7:4	R/W	0x0	TX3_CH1_MAP TX3 Channel 1 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
3:0	R/W	0x0	TX3_CH0_MAP TX3 Channel 0 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

**7.1.6.26 0x0064 I2S/PCM RX Channel Select Register (Default Value: 0x0000\_0000)**

Offset: 0x0064			Register Name: I2S/PCM_RXCHSEL
Bit	Read/Write	Default/Hex	Description
31:22	/	/	/
21:20	R/W	0x0	RX_OFFSET RX Offset Tune (RX Data offset to LRCK) 0: No offset n: Data is offset by n BCLKs to LRCK
19:16	R/W	0x0	RX_CHSEL RX Channel (Slot) Number Select for Input 0000: 1 channel or slot ... 0111: 8 channels or slots 1000: 9 channels or slots ... 1111: 16 channels or slots
15:0	/	/	/

**7.1.6.27 0x0068 I2S/PCM RX Channel Mapping Register0 (Default Value: 0x0000\_0000)**

Offset: 0x0068			Register Name: I2S/PCM_RXCHMAPO
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29:28	R/W	0x0	RX_CH15_SELECT RX Channel 15 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3

Offset: 0x0068			Register Name: I2S/PCM_RXCHMAP0
Bit	Read/Write	Default/Hex	Description
27:24	R/W	0x0	RX_CH15_MAP RX Channel 15 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
23:22	/	/	/
21:20	R/W	0x0	RX_CH14_SELECT RX Channel 14 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3
19:16	R/W	0x0	RX_CH14_MAP RX Channel 14 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
15:14	/	/	/
13:12	R/W	0x0	RX_CH13_SELECT RX Channel 13 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3

Offset: 0x0068			Register Name: I2S/PCM_RXCHMAP0
Bit	Read/Write	Default/Hex	Description
11:8	R/W	0x0	RX_CH13_MAP RX Channel 13 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
7:6	/	/	/
5:4	R/W	0x0	RX_CH12_SELECT RX Channel 12 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3
3:0	R/W	0x0	RX_CH12_MAP RX Channel 12 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

#### 7.1.6.28 0x006C I2S/PCM RX Channel Mapping Register1 (Default Value: 0x0000\_0000)

Offset: 0x006C			Register Name: I2S/PCM_RXCHMAP1
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29:28	R/W	0x0	RX_CH11_SELECT RX Channel 11 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3

Offset: 0x006C			Register Name: I2S/PCM_RXCHMAP1
Bit	Read/Write	Default/Hex	Description
27:24	R/W	0x0	<p>RX_CH11_MAP RX Channel 11 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample</p>
23:22	/	/	/
21:20	R/W	0x0	<p>RX_CH10_SELECT RX Channel 10 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3</p>
19:16	R/W	0x0	<p>RX_CH10_MAP RX Channel 10 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample</p>
15:14	/	/	/
13:12	R/W	0x0	<p>RX_CH9_SELECT RX Channel 9 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3</p>

Offset: 0x006C			Register Name: I2S/PCM_RXCHMAP1
Bit	Read/Write	Default/Hex	Description
11:8	R/W	0x0	RX_CH9_MAP RX Channel 9 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
7:6	/	/	/
5:4	R/W	0x0	RX_CH8_SELECT RX Channel 8 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3
3:0	R/W	0x0	RX_CH8_MAP RX Channel 8 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

#### 7.1.6.29 0x0070 I2S/PCM RX Channel Mapping Register2 (Default Value: 0x0000\_0000)

Offset: 0x0070			Register Name: I2S/PCM_RXCHMAP2
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29:28	R/W	0x0	RX_CH7_SELECT RX Channel 7 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3

Offset: 0x0070			Register Name: I2S/PCM_RXCHMAP2
Bit	Read/Write	Default/Hex	Description
27:24	R/W	0x0	<p>RX_CH7_MAP RX Channel 7 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample</p>
23:22	/	/	/
21:20	R/W	0x0	<p>RX_CH6_SELECT RX Channel 6 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3</p>
19:16	R/W	0x0	<p>RX_CH6_MAP RX Channel 6 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample</p>
15:14	/	/	/
13:12	R/W	0x0	<p>RX_CH5_SELECT RX Channel 5 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3</p>

Offset: 0x0070			Register Name: I2S/PCM_RXCHMAP2
Bit	Read/Write	Default/Hex	Description
11:8	R/W	0x0	RX_CH5_MAP RX Channel 5 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
7:6	/	/	/
5:4	R/W	0x0	RX_CH4_SELECT RX Channel 4 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3
3:0	R/W	0x0	RX_CH4_MAP RX Channel 4 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

#### 7.1.6.30 0x0074 I2S/PCM RX Channel Mapping Register3 (Default Value: 0x0000\_0000)

Offset: 0x0074			Register Name: I2S/PCM_RXCHMAP3
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29:28	R/W	0x0	RX_CH3_SELECT RX Channel 3 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3

Offset: 0x0074			Register Name: I2S/PCM_RXCHMAP3
Bit	Read/Write	Default/Hex	Description
27:24	R/W	0x0	<p>RX_CH3_MAP RX Channel 3 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample</p>
23:22	/	/	/
21:20	R/W	0x0	<p>RX_CH2_SELECT RX Channel 2 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3</p>
19:16	R/W	0x0	<p>RX_CH2_MAP RX Channel 2 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample</p>
15:14	/	/	/
13:12	R/W	0x0	<p>RX_CH1_SELECT RX Channel 1 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3</p>

Offset: 0x0074			Register Name: I2S/PCM_RXCHMAP3
Bit	Read/Write	Default/Hex	Description
11:8	R/W	0x0	RX_CH1_MAP RX Channel 1 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
7:6	/	/	/
5:4	R/W	0x0	RX_CHO_SELECT RX Channel 0 Select 00: SDIO 01: SDI1 10: SDI2 11: SDI3
3:0	R/W	0x0	RX_CHO_MAP RX Channel 0 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

#### 7.1.6.31 0x0080 I2S/PCM ASRC MCLK Configuration Register (Default Value: 0x0000\_0000)

Offset: 0x0080			Register Name: MCLKCFG
Bit	Read/Write	Default/Hex	Description
31:17	R	0x0	/
16	R/W	0x0	ASRC_MCLK_GATE ASRC Clock Gate Enable Control 0: Gated 1: Not gated
15:4	/	/	/

Offset: 0x0080			Register Name: MCLKCFG
Bit	Read/Write	Default/Hex	Description
3:0	R/W	0x0	<p>ASRC_MCLK_FREQ_DIV_COE  Frequency Division Coefficient</p> <p>4'd0 = Res (no output),  4'd1 = 1x,  4'd2 = 1/2x,  4'd3 = 1/4x,  4'd4 = 1/6x,  4'd5 = 1/8x,  4'd6 = 1/12x,  4'd7 = 1/16x,  4'd8 = 1/24x,  4'd9 = 1/32x,  4'd10 = 1/48x,  4'd11 = 1/64x,  4'd12 = 1/96x,  4'd13 = 1/128x,  4'd14 = 1/176x,  4'd15 = 1/192x,  others = Res</p>

#### 7.1.6.32 0x0084 I2S/PCM ASRC OUT Sample Configuration Register (Default Value: 0x0000\_0000)

Offset: 0x0084			Register Name: FSOUTCFG
Bit	Read/Write	Default/Hex	Description
31:21	R	0x0	/
20	R/W	0x0	<p>FSOUT_GATE  fsout Clock Gate Enable Control</p> <p>0: Gated  1: Not gated</p>

Offset: 0x0084			Register Name: FSOUTCFG
Bit	Read/Write	Default/Hex	Description
19:16	R/W	0x0	<p>FSOUT_CLK_SRC_SEL fsout Clock Source Select</p> <p>00: I2S_ASRC_CLK 01: ACLK 10: ACLKM 11: BCLK Others: Reserved</p>
15:8	/	/	/
7:4	R/W	0x0	<p>FSOUT_CLK_FREQ_DIV_COE1 fsout Frequency Division Coefficient 1</p> <p>The First Division Factor</p> <p>It has two levels of frequency division, the first level is bit[7:4], the second level is bit[3:0], and the frequency division factors are multiplied by the two division factors, the division relationship of the two divisions are the same.</p> <p>4'd0 = Res (no output),          4'd1 = 1x,          4'd2 = 1/2x,          4'd3 = 1/4x,          4'd4 = 1/6x,          4'd5 = 1/8x,          4'd6 = 1/12x,          4'd7 = 1/16x,          4'd8 = 1/24x,          4'd9 = 1/32x,          4'd10 = 1/48x,          4'd11 = 1/64x,          4'd12 = 1/96x,          4'd13 = 1/128x,          4'd14 = 1/176x,          4'd15 = 1/192x</p>

Offset: 0x0084			Register Name: F <sub>SOUT</sub> CFG
Bit	Read/Write	Default/Hex	Description
3:0	R/W	0x0	<p>FSOUT_CLK_FREQ_DIV_COE2</p> <p>fsout Frequency Division Coefficient 2</p> <p>The Second Division Factor</p> <p>4'd0 = Res (no output),</p> <p>4'd1 = 1x,</p> <p>4'd2 = 1/2x,</p> <p>4'd3 = 1/4x,</p> <p>4'd4 = 1/6x,</p> <p>4'd5 = 1/8x,</p> <p>4'd6 = 1/12x,</p> <p>4'd7 = 1/16x,</p> <p>4'd8 = 1/24x,</p> <p>4'd9 = 1/32x,</p> <p>4'd10 = 1/48x,</p> <p>4'd11 = 1/64x,</p> <p>4'd12 = 1/96x,</p> <p>4'd13 = 1/128x,</p> <p>4'd14 = 1/176x,</p> <p>4'd15 = 1/192x</p>

#### 7.1.6.33 0x0088 I2S/PCM IN Sample Pulse Extend Configuration Register (Default Value: 0x0000\_0000)

Offset: 0x0088			Register Name: F <sub>sinEXT</sub> CFG
Bit	Read/Write	Default/Hex	Description
31:17	R	0x0	/
16	R/W	0x0	<p>Extend Enable</p> <p>0: Disabled</p> <p>1: Enabled</p> <p>Enable the bit when using ASRC.</p>
15:0	R/W	0x0	<p>The Cycle Number of Pulse Extend</p> <p>The cycle is BCLK clock and is 1 at least.</p>

**7.1.6.34 0x008C I2S/PCM ASRC Enable Register (Default Value: 0x0000\_0000)**

Offset: 0x008C			Register Name: ASRCEN
Bit	Read/Write	Default/Hex	Description
31:1	R	0x0	/
0	R/W	0x0	ASRC Function Enable 0: Disabled 1: Enabled

**7.1.6.35 0x0090 I2S/PCM ASRC Manual Configuration Register (Default Value: 0x0000\_0000)**

Offset: 0x0090			Register Name: ASRCMANCFG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	ASRC_RATIO_MANUAL_EN Manual Configuration of ASRC Ratio Enable 0: Disabled 1: Enabled
30:26	R	0x0	/
25:0	R/W	0x0	ASRC_RATIO_VALUE_MANUAL_CFG ASRC Ratio Value Manual Configure The ratio value is an unsigned 26-bit number and uses 4.22 data format, which means there are 4 bits to the left of the decimal point and 22 bits to the right of the decimal point.

**7.1.6.36 0x0094 I2S/PCM ASRC Ratio State Register (Default Value: 0x0040\_0000)**

Offset: 0x0094			Register Name: ASRCRATIOSTAT
Bit	Read/Write	Default/Hex	Description
31:30	R	0x0	/
29	R	0x0	ASRC_BUF_OVERFLOW_STA ASRC Receive Data Buffer Overflow State It can control the mute with lock. 0: No overflow 1: Overflow

Offset: 0x0094			Register Name: ASRCRATIOSTAT
Bit	Read/Write	Default/Hex	Description
28	R	0x0	ADAPT_COMPUT_LOCK Adaptive Ratio Computational Lock 0: Unlocked 1: Locked
27:26	R	0x0	/
25:0	R	0x400000	ADAPT_COMPUT_VALUE Adaptive Ratio Computational Value

#### 7.1.6.37 0x0098 I2S/PCM ASRC FIFO State Register (Default Value: 0x0000\_0000)

Offset: 0x0098			Register Name: ASRCFIFOSTAT
Bit	Read/Write	Default/Hex	Description
31:9	/	/	/
8:0	R	0x0	ASRC_RX_FIFO_FULL_LEVEL ASRC RXFIFO Full Level The manually-configured FIFO fill level for the ratio value of the received data.

#### 7.1.6.38 0x009C I2S/PCM MBIST Test Configuration Register (Default Value: 0x0000\_0000)

Offset: 0x009C			Register Name: ASRCMBISTCFG
Bit	Read/Write	Default/Hex	Description
31:9	/	/	/
8	R/W	0x0	ASRC_RAM_BIST_EN ASRC RAM BIST Enable Enable the RAM BIST.
7:1	/	/	/
0	R/W	0x0	ASRC_ROM_BIST_EN ASRC ROM BIST Enable Enable the ROM BIST.

**7.1.6.39 0x00A0 I2S/PCM ASRC MBIST Test State Register (Default Value: 0x0000\_0002)**

Offset: 0x00A0			Register Name: ASRCMBISTSTA
Bit	Read/Write	Default/Hex	Description
31:19	/	/	/
18	R	0x0	ROM_BIST_ERROR_XOR ROM BIST error xor
17	R	0x0	ROM_BIST_ERROR_SUM ROM BIST error sum
16	R	0x0	ROM_BUSY_STATUS ROM BUSY STATUS 1: ROM busy 0: ROM idle
15:8	/	/	/
7	R	0x0	RAM_BIST_ERR_STATUS RAM BIST error status 1: Error 0: No effect
6:4	R	0x0	RAM_BIST_ERROR_PATTERN. RAM BIST error pattern
3:2	R	0x0	RAM_BIST_ERROR_CYCLE RAM BIST error cycle
1	R	0x1	RAM_STOP_STATUS RAM stop status 1: Stop 0: Running
0	R	0x0	RAM_BUSY_STATUS RAM busy status 1: RAM busy 0: RAM idle

## 7.2 DMIC

### 7.2.1 Overview

The DMIC controller supports one 8-channel digital microphone interface and can output 128 fs or 64 fs (fs = ADC sample rate).

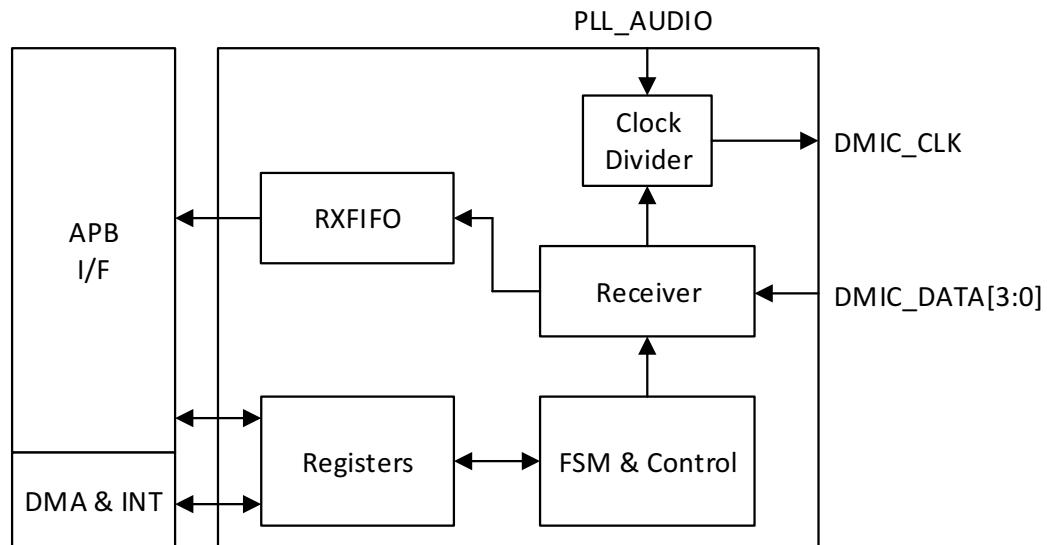
The DMIC controller includes the following features:

- Supports up to 8 channels
- Sample rate from 8 kHz to 48 kHz

### 7.2.2 Block Diagram

The following figure shows a block diagram of the DMIC.

**Figure 7-12 DMIC Block Diagram**



### 7.2.3 Functional Description

#### 7.2.3.1 External Signals

The following table describes the external signals of DMIC.

**Table 7-5 DMIC External Signals**

<b>Signal</b>	<b>Description</b>	<b>Type</b>
DMIC-CLK	Digital Microphone Clock Output	O
DMIC-DATA0	Digital Microphone Data Input	I
DMIC-DATA1	Digital Microphone Data Input	I
DMIC-DATA2	Digital Microphone Data Input	I
DMIC-DATA3	Digital Microphone Data Input	I

#### 7.2.3.2 Clock Sources

The following table describes the clock source for DMIC. For clock setting, configurations, and gating information, refer to section 3.3 “[CCU](#)”.

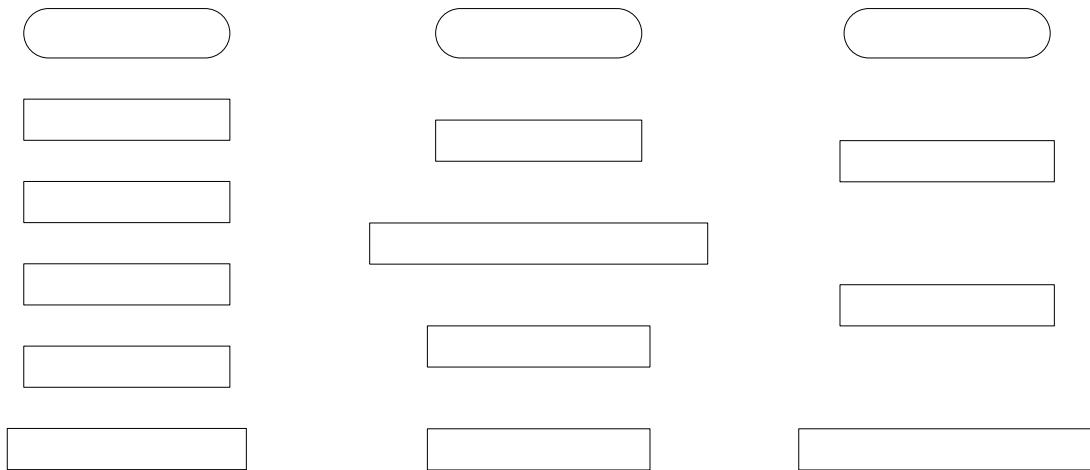
**Table 7-6 DMIC Clock Sources**

<b>Clock Sources</b>	<b>Description</b>
PLL_AUDIO0(1X)	By default, PLL_AUDIO0(1X) is 24.5714 MHz.
PLL_AUDIO1(DIV2)	By default, PLL_AUDIO1 is 3072 MHz, PLL_AUDIO1(DIV2) is 1536 MHz, and
PLL_AUDIO1(DIV5)	PLL_AUDIO1(DIV5) is 614.4 MHz (24.576 MHz*25).

#### 7.2.3.3 Operation Mode

The software operation of the DMIC contains five steps: system setup, DMIC initialization, channel setup, DMA setup, and Enable/Disable module.

The following figure shows the flow chart of the whole operation, the system setup, and the DMIC initialization.

**Figure 7-13 DMIC Operation Mode**

### 1. System Setup and DMIC Initialization

The first step in the system setup is properly programming the GPIO because the DMIC port is a multiplex pin. For functions of the multiplex pins, refer to the pin multiplex specification.

Perform the following steps for the clock source. Firstly, disable the PLL\_AUDIO through [PLL\\_AUDIOx Control Register](#)[PLL\_ENABLE]. Secondly, set up the frequency of the PLL\_AUDIO in [PLL\\_AUDIOx Control Register](#). Then enable PLL\_AUDIO. After that, enable the DMIC gating through [DMIC CLK REG](#) when you checkout that the LOCK bit of [PLL\\_AUDIOx Control Register](#) becomes 1. At last, reset and enable the DMIC bus gating by [DMIC BGR REG](#).

After the system setup, the register of DMIC can be setup. Firstly, initialize the DMIC. You should close the globe enable bit ([DMIC\\_EN](#)[8]), data channel enable bit ([DMIC\\_EN](#)[7:0]) by writing 0 to it. After that, flush the RXFIFO by writing 1 to [DMIC\\_RXFIFO\\_CTR](#)[31]. At last, you can clear the Data/RXFIFO counter by writing 1 to [DMIC\\_RXFIFO\\_STA](#), [DMIC\\_CNT](#).

### 2. Channel Setup and DMA Setup

You can set up the sample rate, the sample resolution, the over-sample rate, the channel number, the RXFIFO output mode, the RXFIFO trigger level, and so on. The setup of the register can be found in the specification.

The DMIC supports two methods to transfer the data. The most common way is DMA, the setup of DMA can be found in section 3.9 “[DMAC](#)”. In this module, you just enable the DRQ.

### 3. Enable and Disable DMIC

To enable the function, you can enable the data channel enable bit ([DMIC\\_EN](#)[7:0]) by writing 1 to it. After that, enable DMIC by writing 1 to the Globe Enable bit ([DMIC\\_EN](#)[8]). Write 0 to [DMIC\\_EN](#)[8] to disable DMIC.

### 7.2.4 Register List

Module Name	Base Address
DMIC	0x02031000

Register Name	Offset	Description
DMIC_EN	0x0000	DMIC Enable Control Register
DMIC_SR	0x0004	DMIC Sample Rate Register
DMIC_CTR	0x0008	DMIC Control Register
DMIC_DATA	0x0010	DMIC Data Register
DMIC_INTC	0x0014	DMIC Interrupt Control Register
DMIC_INTS	0x0018	DMIC Interrupt Status Register
DMIC_RXFIFO_CTR	0x001C	DMIC RXFIFO Control Register
DMIC_RXFIFO_STA	0x0020	DMIC RXFIFO Status Register
DMIC_CH_NUM	0x0024	DMIC Channel Numbers Register
DMIC_CH_MAP	0x0028	DMIC Channel Mapping Register
DMIC_CNT	0x002C	DMIC Counter Register
DATA0_DATA1_VOL_CTR	0x0030	Data0 and Data1 Volume Control Register
DATA2_DATA3_VOL_CTR	0x0034	Data2 And Data3 Volume Control Register
HPF_EN_CTR	0x0038	High Pass Filter Enable Control Register
HPF_COEF_REG	0x003C	High Pass Filter Coefficient Register
HPF_GAIN_REG	0x0040	High Pass Filter Gain Register

### 7.2.5 Register Description

#### 7.2.5.1 0x0000 DMIC Enable Control Register (Default Value: 0x0000\_0000)

Offset: 0x0000			Register Name: DMIC_EN
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/

Offset: 0x0000			Register Name: DMIC_EN
Bit	Read/Write	Default/Hex	Description
29	R/W	0x0	<p>RX_SYNC_EN_START Audio Subsys RX Synchronize Enable Start Includes Audio codec/ I2S1/I2S2/DMIC. The bit takes effect only when RX_SYNC_EN is set to 1. 0: Disabled 1: Enabled</p>
28	R/W	0x0	<p>RX_SYNC_EN DMIC RX Synchronize Enable 0: Disabled 1: Enabled</p>
27:9	/	/	/
8	R/W	0x0	<p>GLOBE_EN DMIC Globe Enable 0: Disabled 1: Enabled</p>
7	R/W	0x0	<p>DATA3_CHR_EN DATA3 Right Channel Enable 0: Disabled 1: Enabled</p>
6	R/W	0x0	<p>DATA3_CHL_EN DATA3 Left Channel Enable 0: Disabled 1: Enabled</p>
5	R/W	0x0	<p>DATA2_CHR_EN DATA2 Right Channel Enable 0: Disabled 1: Enabled</p>
4	R/W	0x0	<p>DATA2_CHL_EN DATA2 Left Channel Enable 0: Disabled 1: Enabled</p>
3	R/W	0x0	<p>DATA1_CHR_EN DATA1 Right Channel Enable 0: Disabled 1: Enabled</p>

Offset: 0x0000			Register Name: DMIC_EN
Bit	Read/Write	Default/Hex	Description
2	R/W	0x0	DATA1_CHL_EN DATA1 Left Channel Enable 0: Disabled 1: Enabled
1	R/W	0x0	DATA0_CHR_EN DATA0 Right Channel Enable 0: Disabled 1: Enabled
0	R/W	0x0	DATA0_CHL_EN DATA0 Left Channel Enable 0: Disabled 1: Enabled

#### 7.2.5.2 0x0004 DMIC Sample Rate Register (Default Value: 0x0000\_0000)

Offset: 0x0004			Register Name: DMIC_SR
Bit	Read/Write	Default/Hex	Description
31:3	/	/	/
2:0	R/W	0x0	DMIC_SR Sample Rate of DMIC 000: 48 kHz 010: 24 kHz 100: 12 kHz 110: Reserved 001: 32 kHz 011: 16 kHz 101: 8 kHz 111: Reserved 44.1 kHz/22.05 kHz/11.025 kHz can be supported by Audio PLL Configure Bit.

### 7.2.5.3 0x0008 DMIC Control Register (Default Value: 0x0000\_0000)

Offset: 0x0008			Register Name: DMIC_CTR
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:9	R/W	0x0	<p>DMICFDT DMIC RXFIFO Delay Time for Writing Data after GLOBE_EN</p> <p>00: 5 ms 01: 10 ms 10: 20 ms 11: 30 ms</p>
8	R/W	0x0	<p>DMICDFEN DMIC RXFIFO Delay Function for Writing Data after GLOBE_EN</p> <p>0: Disabled 1: Enabled</p>
7	R/W	0x0	<p>DATA3 Left Data and Right Data Swap Enable</p> <p>0: Disabled 1: Enabled</p>
6	R/W	0x0	<p>DATA2 Left Data and Right Data Swap Enable</p> <p>0: Disabled 1: Enabled</p>
5	R/W	0x0	<p>DATA1 Left Data and Right Data Swap Enable</p> <p>0: Disabled 1: Enabled</p>
4	R/W	0x0	<p>DATA0 Left Data and Right Data Swap Enable</p> <p>0: Disabled 1: Enabled</p>
3:1	/	/	/
0	R/W	0x0	<p>DMIC Oversample Rate</p> <p>0: 128 (Supports 8 kHz to 24 kHz) 1: 64 (Supports 16 kHz to 48 kHz)</p>

### 7.2.5.4 0x0010 DMIC DATA Register (Default Value: 0x0000\_0000)

Offset: 0x0010			Register Name: DMIC_DATA
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	DMIC_DATA

#### 7.2.5.5 0x0014 DMIC Interrupt Control Register (Default Value: 0x0000\_0000)

Offset: 0x0014			Register Name: DMIC_INTC
Bit	Read/Write	Default/Hex	Description
31:3	/	/	/
2	R/W	0x0	<p>RXFIFO_DRQ_EN DMIC RXFIFO Data Available DRQ Enable 0: Disabled 1: Enabled</p>
1	R/W	0x0	<p>RXFIFO_OVERRUN_IRQ_EN DMIC RXFIFO Overrun IRQ Enable 0: Disabled 1: Enabled</p>
0	R/W	0x0	<p>DATA_IRQ_EN DMIC RXFIFO Data Available IRQ Enable 0: Disabled 1: Enabled</p>

#### 7.2.5.6 0x0018 DMIC Interrupt Status Register (Default Value: 0x0000\_0000)

Offset: 0x0018			Register Name: DMIC_INTS
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1	R/W1C	0x0	<p>RXFIFO_OVERRUN_IRQ_PENDING DMIC RXFIFO Overrun Pending Interrupt 0: No pending IRQ 1: RXFIFO overrun pending IRQ Writing '1' to clear this interrupt or automatically clear if the interrupt condition fails.</p>
0	R/W1C	0x0	<p>RXFIFO_DATA_IRQ_PENDING DMIC RXFIFO Data Available Pending Interrupt 0: No pending IRQ 1: Data available pending IRQ Writing '1' to clear this interrupt or automatically clear if the interrupt condition fails.</p>

### 7.2.5.7 0x001C DMIC RXFIFO Control Register (Default Value: 0x0000\_0040)

Offset: 0x001C			Register Name: DMIC_RXFIFO_CTR
Bit	Read/Write	Default/Hex	Description
31	R/W1C	0x0	DMIC_RXFIFO_FLUSH DMIC RXFIFO Flush Writing '1' to flush RXFIFO, self clear to '0'
30:10	/	/	/
9	R/W	0x0	RXFIFO_MODE RXFIFO Output Mode (Mode 0, 1) 0: Expanding '0' at LSB of RXFIFO register 1: Expanding received sample sign bit at MSB of RXFIFO register For 24-bit received audio sample: Mode 0: RXDATA[31:0] = {RXFIFO_O[20:0], 11'h0} Mode 1: RXDATA[31:0] = {8{RXFIFO_O[20]}, RXFIFO_O[20:0], 3'h0} For 16-bit received audio sample: Mode 0: RXDATA[31:0] = {RXFIFO_O[20:5], 16'h0} Mode 1: RXDATA[31:0] = {16{RXFIFO_O[20]}, RXFIFO_O[20:5]}
8	R/W	0x0	Sample_Resolution 0: 16-bit 1: 24- bit
7:0	R/W	0x40	RXFIFO_TRG_LEVEL RXFIFO Trigger Level (TRLV[7:0]) Interrupt and DMA request trigger level for DMIC RXFIFO normal condition IRQ/DRQ Generated when WLEVEL > TRLV[7:0] WLEVEL represents the number of valid samples in the DMIC RXFIFO

### 7.2.5.8 0x0020 DMIC RXFIFO Status Register (Default Value: 0x0000\_0000)

Offset: 0x0020			Register Name: DMIC_RXFIFO_STA
Bit	Read/Write	Default/Hex	Description
31:9	/	/	/
8	R	0x1	Reserved

Offset: 0x0020			Register Name: DMIC_RXFIFO_STA
Bit	Read/Write	Default/Hex	Description
7:0	R/W	0x0	DMIC_DATA_CNT DMIC RXFIFO Available Sample Word Counter

#### 7.2.5.9 0x0024 DMIC Channel Numbers Register (Default Value: 0x0000\_0001)

Offset: 0x0024			Register Name: DMIC_CH_NUM
Bit	Read/Write	Default/Hex	Description
31:3	/	/	/
2:0	R/W	0x1	DMIC_CH_NUM DMIC enable channel numbers are (N + 1).

#### 7.2.5.10 0x0028 DMIC Channel Mapping Register (Default Value: 0x7654\_3210)

Offset: 0x0028			Register Name: DMIC_CH_MAP
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x7	DMIC_CH7_MAP DMIC Channel 7 Mapping 0000: DATA0 Left Channel 0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel 0110: DATA3 Left Channel 0111: DATA3 Right Channel

Offset: 0x0028			Register Name: DMIC_CH_MAP
Bit	Read/Write	Default/Hex	Description
27:24	R/W	0x6	DMIC_CH6_MAP DMIC Channel 6 Mapping 0000: DATA0 Left Channel 0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel 0110: DATA3 Left Channel 0111: DATA3 Right Channel
23:20	R/W	0x5	DMIC_CH5_MAP DMIC Channel 5 Mapping 0000: DATA0 Left Channel 0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel 0110: DATA3 Left Channel 0111: DATA3 Right Channel
19:16	R/W	0x4	DMIC_CH4_MAP DMIC Channel 4 Mapping 0000: DATA0 Left Channel 0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel 0110: DATA3 Left Channel 0111: DATA3 Right Channel

Offset: 0x0028			Register Name: DMIC_CH_MAP
Bit	Read/Write	Default/Hex	Description
15:12	R/W	0x3	<p>DMIC_CH3_MAP</p> <p>DMIC Channel 3 Mapping</p> <p>0000: DATA0 Left Channel</p> <p>0001: DATA0 Right Channel</p> <p>0010: DATA1 Left Channel</p> <p>0011: DATA1 Right Channel</p> <p>0100: DATA2 Left Channel</p> <p>0101: DATA2 Right Channel</p> <p>0110: DATA3 Left Channel</p> <p>0111: DATA3 Right Channel</p>
11:8	R/W	0x2	<p>DMIC_CH2_MAP</p> <p>DMIC Channel 2 Mapping</p> <p>0000: DATA0 Left Channel</p> <p>0001: DATA0 Right Channel</p> <p>0010: DATA1 Left Channel</p> <p>0011: DATA1 Right Channel</p> <p>0100: DATA2 Left Channel</p> <p>0101: DATA2 Right Channel</p> <p>0110: DATA3 Left Channel</p> <p>0111: DATA3 Right Channel</p>
7:4	R/W	0x1	<p>DMIC_CH1_MAP</p> <p>DMIC Channel 1 Mapping</p> <p>0000: DATA0 Left Channel</p> <p>0001: DATA0 Right Channel</p> <p>0010: DATA1 Left Channel</p> <p>0011: DATA1 Right Channel</p> <p>0100: DATA2 Left Channel</p> <p>0101: DATA2 Right Channel</p> <p>0110: DATA3 Left Channel</p> <p>0111: DATA3 Right Channel</p>

Offset: 0x0028			Register Name: DMIC_CH_MAP
Bit	Read/Write	Default/Hex	Description
3:0	R/W	0x0	<p>DMIC_CH0_MAP DMIC Channel0 Mapping</p> <p>0000: DATA0 Left Channel 0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel 0110: DATA3 Left Channel 0111: DATA3 Right Channel</p>

#### 7.2.5.11 0x002C DMIC Counter Register (Default Value: 0x0000\_0000)

Offset: 0x002C			Register Name: DMIC_CNT
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<p>DMIC_CNT RX Sample Counter</p> <p>The audio sample number of writing into RXFIFO. When one sample is read by Digital Audio Engine, the RX sample counter register increases by one. The RX sample counter register can be set to any initial value at any time. After being updated by the initial value, the counter register should count on the base of this initial value.</p> <p><b>Note:</b> It is used for Audio/Video Synchronization.</p>

**7.2.5.12 0x0030 DATA0 and DATA1 Volume Control Register (Default Value: 0xA0A0\_A0A0)**

Offset: 0x0030			Register Name: DATA0_DATA1_VOL_CTR
Bit	Read/Write	Default/Hex	Description
31:24	R/W	0xA0	DATA1L_VOL Data1 Left Channel Volume Control (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB ... 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB ... 0xFF: 71.25 dB
23:16	R/W	0xA0	DATA1R_VOL Data1 Right Channel Volume Control (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB ... 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB ... 0xFF: 71.25 dB
15:8	R/W	0xA0	DATA0L_VOL Data0 Left Channel Volume Control (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB ... 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB ... 0xFF: 71.25 dB

Offset: 0x0030			Register Name: DATA0_DATA1_VOL_CTR
Bit	Read/Write	Default/Hex	Description
7:0	R/W	0xA0	<p>DATA0R_VOL Data0 Right Channel Volume Control (-119.25 dB to 71.25 dB, 0.75 dB/Step)</p> <p>0x00: Mute 0x01: -119.25 dB 0x02: ... 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB 0xA2: ... 0xFF: 71.25 dB</p>

#### 7.2.5.13 0x0034 DATA2 and DATA3 Volume Control Register (Default Value: 0xA0A0\_A0A0)

Offset: 0x0034			Register Name: DATA2_DATA3_VOL_CTR
Bit	Read/Write	Default/Hex	Description
31:24	R/W	0xA0	<p>DATA3L_VOL Data3 Light Channel Volume Control (-119.25 dB to 71.25 dB, 0.75 dB/Step)</p> <p>0x00: Mute 0x01: -119.25 dB 0x02: ... 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB 0xA2: ... 0xFF: 71.25 dB</p>

Offset: 0x0034			Register Name: DATA2_DATA3_VOL_CTR
Bit	Read/Write	Default/Hex	Description
23:16	R/W	0xA0	DATA3R_VOL Data3 Right Channel Volume Control (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB ... 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB ... 0xFF: 71.25 dB
15:8	R/W	0xA0	DATA2L_VOL Data2 Light Channel Volume Control (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB ... 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB ... 0xFF: 71.25 dB
7:0	R/W	0xA0	DATA2R_VOL Data2 Right Channel Volume Control (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB ... 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB ... 0xFF: 71.25 dB

**7.2.5.14 0x0038 High Pass Filter Enable Control Register (Default Value: 0x0000\_0000)**

Offset: 0x0038			Register Name: HPF_EN_CTR
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R/W	0x0	HPF_DATA3_CHR_EN High Pass Filter DATA3 Right Channel Enable 0: Disabled 1: Enabled
6	R/W	0x0	HPF_DATA3_CHL_EN High Pass Filter DATA3 Left Channel Enable 0: Disabled 1: Enabled
5	R/W	0x0	HPF_DATA2_CHR_EN High Pass Filter DATA2 Right Channel Enable 0: Disabled 1: Enabled
4	R/W	0x0	HPF_DATA2_CHL_EN High Pass Filter DATA2 Left Channel Enable 0: Disabled 1: Enabled
3	R/W	0x0	HPF_DATA1_CHR_EN High Pass Filter DATA1 Right Channel Enable 0: Disabled 1: Enabled
2	R/W	0x0	HPF_DATA1_CHL_EN High Pass Filter DATA1 Left Channel Enable 0: Disabled 1: Enabled
1	R/W	0x0	HPF_DATA0_CHR_EN High Pass Filter DATA0 Right Channel Enable 0: Disabled 1: Enabled
0	R/W	0x0	HPF_DATA0_CHL_EN High Pass Filter DATA0 Left Channel Enable 0: Disabled 1: Enabled

**7.2.5.15 0x003C High Pass Filter Coefficient Register (Default Value: 0x00FF\_AA45)**

Offset: 0x003C			Register Name: HPF_COEF_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x00FFAA45	HPF_COE High Pass Filter Coefficient

**7.2.5.16 0x0040 High Pass Filter Gain Register (Default Value: 0x00FF\_D522)**

Offset: 0x0040			Register Name: HPF_GAIN_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x00FFD522	HPF_GAIN High Pass Filter Gain

## 7.3 OWA

### 7.3.1 Overview

The One Wire Audio (OWA) provides a serial bus interface for audio data. This interface is widely used for consumer audio.

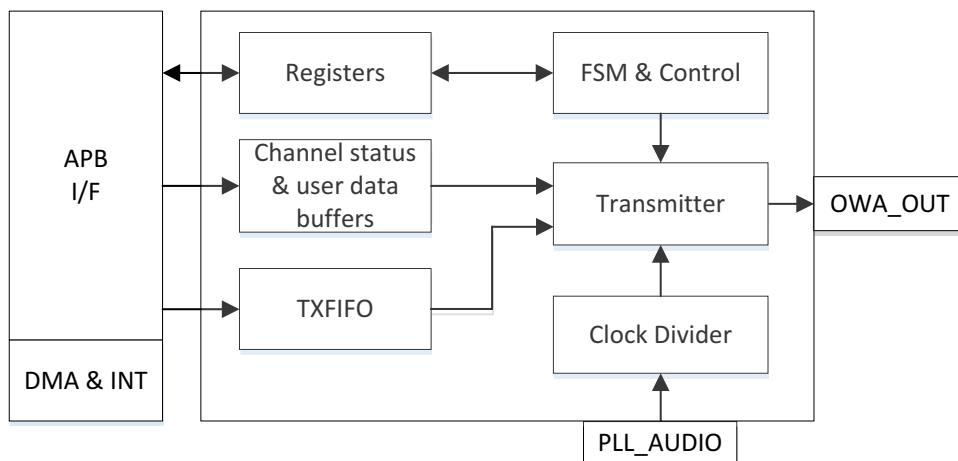
The OWA includes the following features:

- One OWA TX
- Compliance with S/PDIF interface
- IEC-60958 transmitter functionality
- Supports 16-bit, 20-bit, and 24-bit data formats
- One 128×24bits TXFIFO for audio data transfer
- Programmable FIFO thresholds
- Supports TX DMA slave interface
- Function clock includes 24.576 MHz and 22.579 MHz frequency
- Hardware parity generation on the transmitter
- Supports channel status insertion for the transmitter
- Supports interrupts and DMA

### 7.3.2 Block Diagram

The following figure shows the OWA block diagram.

**Figure 7-14 OWA Block Diagram**



OWA contains the following sub-blocks:

**Table 7-7 OWA Sub-blocks**

Sub-block	Description
Registers	Analyze the configuration parameter, DMA requests, and IRQ feedbacks.
Transmitter	Send the data
FSM	Finite state machine
Clock Divider	Clock divider circuit

### 7.3.3 Functional Description

#### 7.3.3.1 External Signals

The OWA is a Biphase-Mark Encoding Digital Audio Transfer protocol. In this protocol, the CLK signal and data signal are transferred in the same line. The following figure describes the external signals of OWA. OWA-OUT is the output pin for the output CLK and DATA.

**Table 7-8 OWA External Signals**

Signal Name	Description	Type
OWA-OUT	OWA output	O

#### 7.3.3.2 Clock Sources

The following tables describe the clock sources for OWA. For clock setting, configurations and gating information, refer to section 3.3 “[CCU](#)”.

**Table 7-9 OWA Clock Sources**

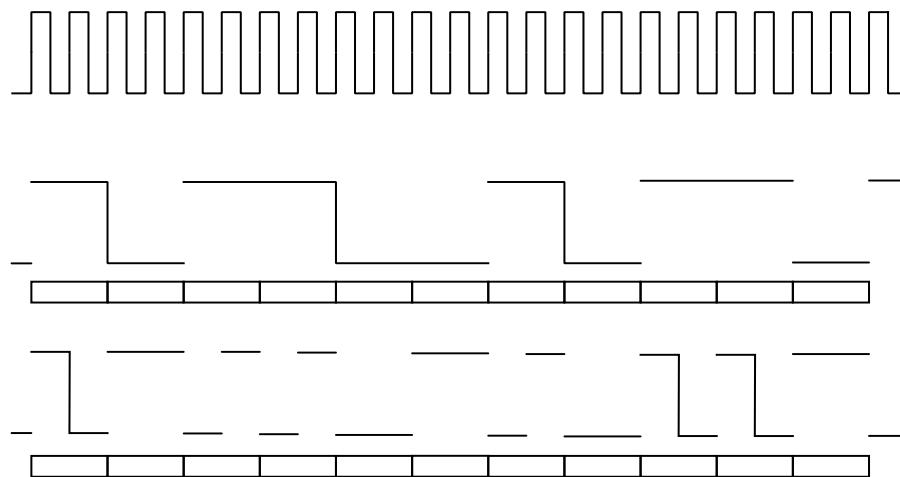
Clock Name	Description
PLL_AUDIO0(1X)	By default, PLL_AUDIO0(1X) is 24.5714 MHz, and PLL_AUDIO0(4X) is 98.2856
PLL_AUDIO0(4X)	MHz.
PLL_AUDIO1(DIV2)	By default, PLL_AUDIO1 is 3072 MHz, PLL_AUDIO1(DIV2) is 1536 MHz, and
PLL_AUDIO1(DIV5)	PLL_AUDIO1(DIV5) is 614.4 MHz (24.576 MHz*25).

#### 7.3.3.3 Biphase-Mark Code (BMC)

In the OWA format, the digital signal is coded using the biphase-mark code (BMC). The clock, frame, and data are embedded in only one signal—the data pin. In the BMC system, each data bit is encoded into two logical states (00, 01, 10, or 11) at the pin. The following figure and table show how data is encoded to the BMC format.

The frequency of the clock is twice the data bit rate, as shown in the following figure. Also, the clock is always programmed to  $128 \times fs$ , where  $fs$  is the sample rate. The device receiving in the OWA format can recover the clock and frame information from the BMC signal.

**Figure 7-15 OWA Biphas-Mark Code**



**Table 7-10 Biphas-Mark Encoder**

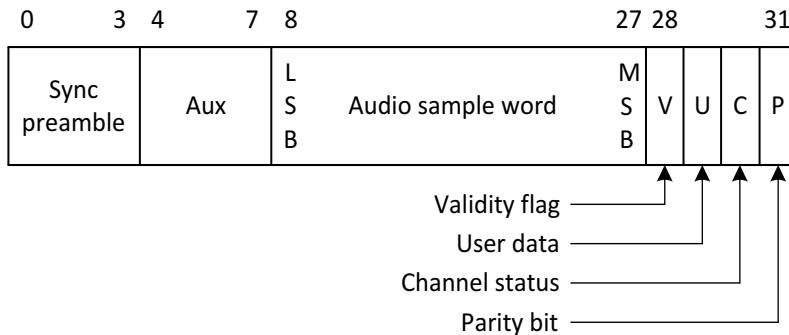
Data	Previous State	BMC
0	0	11
0	1	00
1	0	10
1	1	01

#### 7.3.3.4 OWA Transmit Format

The OWA supports digital audio data transfer.

Every audio sample transmitted in a sub-frame consists of 32-bit, numbered from 0 to 31. The following figure shows a sub-frame.

Figure 7-16 OWA Sub-Frame Format



**Bits 0-3** carry one of the four permitted preambles to signify the type of audio sample in the current sub-frame. The preamble is not encoded in BMC format, and therefore the preamble code can contain more than two consecutive 0 or 1 logical states in a row.

**Bits 4-27** carry the audio sample word in linear 2s-complement representation. The most-significant bit (MSB) is carried by bit 27. When a 24-bit coding range is used, the least-significant bit (LSB) is in bit 4. When a 20-bit coding range is used, bit[8:27] carry the audio sample word with the LSB in bit 8. Bit[4:7] may be used for other applications and are designated auxiliary sample bits.

If the source provides fewer bits than the interface allows (either 20 or 24), the unused LSBs are set to logical 0. For a nonlinear PCM audio application or a data application, the main data field may carry any other information.

**Bit 28** carries the validity bit (V) associated with the main data field in the sub-frame.

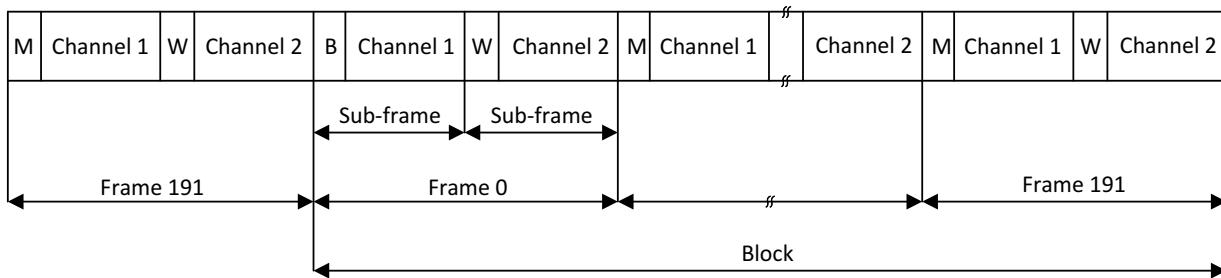
**Bit 29** carries the user data channel (U) associated with the main data field in the sub-frame.

**Bit 30** carries the channel status information (C) associated with the main data field in the sub-frame. The channel status indicates if the data in the sub-frame is a digital audio or some other type of data.

**Bit 31** carries a parity bit (P) such that bit 4-31 carry an even number of 1s and an even number of 0s (even parity). As shown in the following table, the preambles (bit 0-3) are also defined with even parity.

Table 7-11 Preamble Codes

Preamble Code	Previous Logical State	Logical State	Description
B (or Z)	0	1110 1000	Start of a block and sub-frame 1
M (or X)	0	1110 0010	Sub-frame 1
W (or Y)	0	1110 0100	Sub-frame 2

**Figure 7-17 OWA Frame/Block Format**

### 7.3.3.5 Audio Sample Ratio Detection

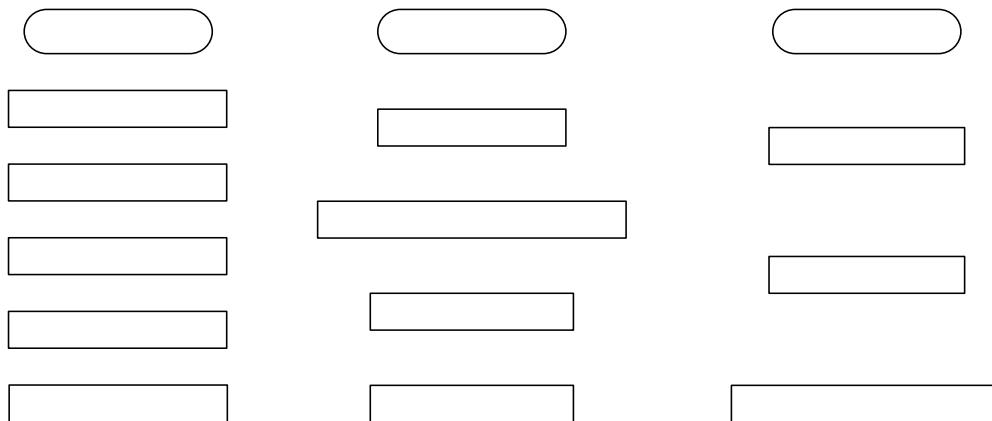
The sampling rate is calculated according to the data pulse back-stepping method. In the first phase lock of the CDR, find 1 Frame period, count by using the high-speed sampling clock, and read the counting value of the pulse, then the sampling rate can be calculated.

**Table 7-12 The Corresponding Relation between Different System Clock and Sample Ratio**

		Sample Clock Cycles	
TX Sample Rate(kHz)		196.608 MHz-SysClk	200 MHz-SysClk
22.05		8916( $\pm 5$ )	9070( $\pm 5$ )
24		8192( $\pm 5$ )	8333( $\pm 5$ )
32		6144( $\pm 5$ )	6250( $\pm 5$ )
44.1		4458( $\pm 5$ )	4535( $\pm 5$ )
48		4096( $\pm 5$ )	4166( $\pm 5$ )
96		2048( $\pm 5$ )	2083( $\pm 5$ )
176.4		1114( $\pm 5$ )	1133( $\pm 5$ )
192		1024( $\pm 5$ )	1041( $\pm 5$ )

### 7.3.3.6 Operation Modes

The software operation of the OWA is divided into five steps: system setup, OWA initialization, channel setup, DMA setup and enable/disable module. The following sections describe these five steps.

**Figure 7-18 OWA Operation Flow**

### 1. System Setup and OWA Initialization

The first step in the OWA initialization is properly programming the GPIO because the OWA port is a multiplex pin. You can find the function in section 9.7 “[GPIO](#)”.

The clock source for the OWA should be followed. Firstly, reset the audio PLL in [PLL\\_AUDIOx Control Register](#). Secondly, set up the frequency of the Audio PLL in the [PLL\\_AUDIOx Control Register](#). After that, enable the OWA gating. Lastly, enable the OWA bus gating.

After the system setup, the register of OWA can be set up. Firstly, reset the OWA by writing 1 to [OWA\\_CTL\[0\]](#) and clear the TX FIFO by writing 1 to [OWA\\_FCTL\[17:16\]](#). After that, enable the globe enable bit by writing 1 to [OWA\\_CTL\[1\]](#) and clear the interrupt and TX counter by setting [OWAISTA](#) and [OWATX\\_CNT](#).

### 2. Channel Setup and DMA Setup

You can set up the audio type, clock divider ratio, the sample format, and the trigger level, and so on. The setup of the register can be found in the specification.

The OWA supports two methods to transfer the data. The most common way is DMA, the configuration of DMA can be found in section 3.9 “[DMAC](#)”. In this module, you just enable the DRQ in [OWA\\_INT\[7\]](#).

### 3. Enable and Disable OWA

To enable the function, you can enable TX by writing [OWATX\\_CFIG\[31\]](#). After that, enable OWA by writing 1 to [OWA\\_CTL\[1\]](#). Writing 0 to [OWA\\_CTL\[1\]](#) to disable process.

#### 7.3.4 Register List

Module Name	Base Address
OWA	0x02036000

Register Name	Offset	Description
OWA_GEN_CTL	0x0000	OWA General Control Register
OWA_TX_CFIG	0x0004	OWA TX Configuration Register
OWAISTA	0x000C	OWA Interrupt Status Register
OWAFCTL	0x0014	OWA FIFO Control Register
OWAFSTA	0x0018	OWA FIFO Status Register
OWAINT	0x001C	OWA Interrupt Control Register
OWATX_FIFO	0x0020	OWA TX FIFO Register
OWATX_CNT	0x0024	OWA TX Counter Register
OWATX_CHSTA0	0x002C	OWA TX Channel Status Register0
OWATX_CHSTA1	0x0030	OWA TX Channel Status Register1

### 7.3.5 Register Description

#### 7.3.5.1 0x0000 OWA General Control Register (Default Value: 0x0000\_0080)

Offset: 0x0000			Register Name: OWA_CTL
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R	0x1	Reserved
6:2	/	/	/
2	R/W	0x0	LOOP Loopback Test 0: Normal Mode 1: Loopback Test When the bit is set to '1', the DOUT and DIN need be connected.
1	R/W	0x0	GEN Global Enable Disabling this bit overrides the operations of enabling and flushing all FIFOs by any other blocks or channels. 0: Disabled 1: Enabled

Offset: 0x0000			Register Name: OWA_CTL
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	RST_TX Reset TX 0: Normal 1: Reset Self clear to 0.

### 7.3.5.2 0x0004 OWA TX Configure Register (Default Value: 0x0000\_00F0)

Offset: 0x0004			Register Name: OWA_TX_CFIG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	TX_SINGLE_MODE Tx Single Channel Mode 0: Disabled 1: Enabled
30:18	/	/	/
17	R/W	0x0	ASS Audio Sample Select when TX FIFO Underrun 0: Sending 0 1: Sending the last audio <b>Note: This bit is only valid in PCM mode.</b>
16	R/W	0x0	TX_AUDIO TX Data Type 0: Linear PCM (Valid bit of both sub-frame set to 0 ) 1: Non-audio (Valid bit of both sub-frame set to 1)
15:9	/	/	/
8:4	R/W	0xF	TX_RATIO TX Clock Divide Ratio Clock divide ratio = TX_TATIO + 1 $F_s = PLL\_AUDIO / [(TX\_TATIO + 1) * 64 * 2]$
3:2	R/W	0x0	TX_SF TX Sample Format 00: 16 bits 01: 20 bits 10: 24 bits 11: Reserved

Offset: 0x0004			Register Name: OWA_TX_CFIG
Bit	Read/Write	Default/Hex	Description
1	R/W	0x0	TX_CHM CHSTMODE 0: Channel status A and B set to 0 1: Channel status A and B generated from TX_CHSTA
0	R/W	0x0	TXEN TX Enable 0: Disabled 1: Enabled

### 7.3.5.3 0x000C OWA Interrupt Status Register (Default Value: 0x0000\_0010)

Offset: 0x000C			Register Name: OWAISTA
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6	R/W1C	0x0	TXU_INT TX FIFO Underrun Pending Interrupt 0: No Pending IRQ 1: FIFO Underrun Pending Interrupt Writing "1" to clear this interrupt.
5	R/W1C	0x0	TXO_INT TX FIFO Overrun Pending Interrupt 0: No Pending IRQ 1: FIFO Overrun Pending Interrupt Writing "1" to clear this interrupt.
4	R/W1C	0x1	TXE_INT TX FIFO Empty Pending Interrupt 0: No Pending IRQ 1: FIFO Empty Pending Interrupt Writing "1" to clear this interrupt or automatically clear if the interrupt condition fails.
3:0	/	/	/

#### 7.3.5.4 0x0014 OWA FIFO Control Register (Default Value: 0x0004\_0200)

Offset: 0x0014			Register Name: OWA_FCTL
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	HUB_EN Audio Hub Enable The bit takes effect only when the TXEN is set to 1. Audio Codec/ I2S1/I2S2/OWA TXFIFO Hub Enable. 0: Disabled 1: Enabled
30	R/W1C	0x0	FTX Write '1' to flush TXFIFO, self clear to '0'.
29:20	/	/	/
19:12	R/W	0x40	TXTL TX FIFO Empty Trigger Level Interrupt and DMA request trigger level for TX FIFO normal condition. Trigger Level = TXTL
11:3	/	/	/
2	R/W	0x0	TXIM TXFIFO Input Mode (Mode0, 1) 0: Valid data at the MSB of TXFIFO Register 1: Valid data at the LSB of TXFIFO Register Example for 20-bit transmitted audio sample: Mode 0: TXFIFO[23:0] = {APB_WDATA[31:12], 4'h0} Mode 1: TXFIFO[23:0] = {APB_WDATA[19:0], 4'h0}
1:0	R/W	0x0	RXOM RXFIFO Output Mode (Mode 0,1,2,3) 00: Expanding '0' at LSB of RXFIFO Register 01: Expanding received sample sign bit at MSB of RXFIFO Register 10: Truncating received samples at high half-word of RXFIFO Register and low half-word of RXFIFO Register is filled by '0' 11: Truncating received samples at low half-word of RXFIFO Register and high half-word of RXFIFO Register is expanded by its sign bit Mode 0: APB_RDATA[31:0] = {RXFIFO[23:0], 8'h0} Mode 1: APB_RDATA[31:0] = {8'RXFIFO[23], RXFIFO[23:0]} Mode 2: APB_RDATA[31:0] = {RXFIFO[23:8], 16'h0} Mode 3: APB_RDATA[31:0] = {16'RXFIFO[23], RXFIFO[23:8]}

### 7.3.5.5 0x0018 OWA FIFO Status Register (Default Value: 0x8080\_0000)

Offset: 0x0018			Register Name: OWA_FSTA
Bit	Read/Write	Default/Hex	Description
31	R	0x1	TXE TXFIFO Empty (indicate the TXFIFO is not full) 0: No room for new sample in TXFIFO 1: More than one room for new sample in TXFIFO ( >= 1 Word )
30:24	/	/	/
23:16	R	0x80	TXE_CNT TXFIFO Empty Space Word Counter
15:0	R	0x0	RXA RXFIFO Available 0: No available data in RXFIFO 1: More than one sample in RXFIFO ( >= 1 Word )

### 7.3.5.6 0x001C OWA Interrupt Control Register (Default Value: 0x0000\_0000)

Offset: 0x001C			Register Name: OWA_INT
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R/W	0x0	TX_DRQ TXFIFO Empty DRQ Enable 0: Disabled 1: Enabled
6	R/W	0x0	TXUI_EN TXFIFO Underrun Interrupt Enable 0: Disabled 1: Enabled
5	R/W	0x0	TXOI_EN TXFIFO Overrun Interrupt Enable 0: Disabled 1: Enabled

Offset: 0x001C			Register Name: OWA_INT
Bit	Read/Write	Default/Hex	Description
4	R/W	0x0	TXEI_EN TXFIFO Empty Interrupt Enable 0: Disabled 1: Enabled
3:0	/	/	/

#### 7.3.5.7 0x0020 OWA TX FIFO Register (Default Value: 0x0000\_0000)

Offset: 0x0020			Register Name: OWA_TXFIFO
Bit	Read/Write	Default/Hex	Description
31:0	W	0x0	TX_DATA Transmitting A, B channel data should be written this register one by one. A channel data is first, and then the B channel data.

#### 7.3.5.8 0x0024 OWA TX Counter Register (Default Value: 0x0000\_0000)

Offset: 0x0024			Register Name: OWA_TX_CNT
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	TX_CNT TX Sample Counter The audio sample number of sending into TXFIFO. When one sample is put into TXFIFO by DMA or by host IO, the TX sample counter register increases by one. The TX sample counter register can be set to any initial value at any time. After updated by the initial value, the counter register should count on the base of this initial value.

#### 7.3.5.9 0x002C OWA TX Channel Status Register0 (Default Value: 0x0000\_0000)

Offset: 0x002C			Register Name: OWA_TX_CHSTA0
Bit	Read/Write	Default/Hex	Description
31: 30	/	/	/

Offset: 0x002C			Register Name: OWA_TX_CHSTA0
Bit	Read/Write	Default/Hex	Description
29:28	R/W	0x0	<p>CA Clock Accuracy 00: Level 2 01: Level 1 10: Level 3 11: Not matched</p>
27:24	R/W	0x0	<p>FREQ Sampling Frequency 0000: 44.1 kHz 0001: Not indicated 0010: 48 kHz 0011: 32 kHz 0100: 22.05 kHz 0101: Reserved 0110: 24 kHz 0111: Reserved 1000: Reserved 1001: 768 kHz 1010: 96 kHz 1011: Reserved 1100: 176.4 kHz 1101: Reserved 1110: 192 kHz 1111: Reserved</p>
23:20	R/W	0x0	<p>CN Channel Number</p>
19:16	R/W	0x0	<p>SN Source Number</p>
15:8	R/W	0x0	<p>CC Category Code Indicates the kind of equipment that generates the digital audio interface signal.</p>
7:6	R/W	0x0	<p>MODE Mode 00: Default Mode 01 to 11: Reserved</p>

Offset: 0x002C			Register Name: OWA_TX_CHSTA0
Bit	Read/Write	Default/Hex	Description
5:3	R/W	0x0	<p>EMP Emphasis Additional format information For bit 1 = “0”, Linear PCM audio mode: 000: 2 audio channels without pre-emphasis 001: 2 audio channels with 50 µs/15 µs pre-emphasis 010: Reserved (for 2 audio channels with pre-emphasis) 011: Reserved (for 2 audio channels with pre-emphasis) 100 to 111: Reserved For bit 1 = “1”, other than Linear PCM applications: 000: Default state 001 to 111: Reserved</p>
2	R/W	0x0	<p>CP Copyright 0: Copyright is asserted 1: No copyright is asserted</p>
1	R/W	0x0	<p>TYPE Audio Data Type 0: Linear PCM samples 1: Non-linear PCM audio</p>
0	R/W	0x0	<p>PRO Application Type 0: Consumer application 1: Professional application This bit must be fixed to “0”.</p>

#### 7.3.5.10 0x0030 OWA TX Channel Status Register1 (Default Value: 0x0000\_0000)

Offset: 0x0030			Register Name: OWA_TX_CHSTA1
Bit	Read/Write	Default/Hex	Description
31:10	/	/	/
9:8	R/W	0x0	<p>CGMS_A 00: Copying is permitted without restriction 01: One generation of copies may be made 10: Condition not be used 11: No copying is permitted</p>

Offset: 0x0030			Register Name: OWA_TX_CHSTA1
Bit	Read/Write	Default/Hex	Description
7:4	R/W	0x0	<p>ORIG_FREQ Original Sampling Frequency</p> <p>0000: Not indicated 0001: 192 kHz 0010: 12 kHz 0011: 176.4 kHz 0100: Reserved 0101: 96 kHz 0110: 8 kHz 0111: 88.2 kHz 1000: 16 kHz 1001: 24 kHz 1010: 11.025 kHz 1011: 22.05 kHz 1100: 32 kHz 1101: 48 kHz 1110: Reserved 1111: 44.1 kHz</p>
3:1	R/W	0x0	<p>WL Sample Word Length</p> <p>For bit 0 = "0":</p> <p>000: Not indicated 001: 16 bits 010: 18 bits 100: 19 bits 101: 20 bits 110: 17 bits 111: Reserved</p> <p>For bit 0 = "1":</p> <p>000: Not indicated 001: 20 bits 010: 22 bits 100: 23 bits 101: 24 bits 110: 21 bits 111: Reserved</p>

Offset: 0x0030			Register Name: OWA_TX_CHSTA1
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	MWL Max Word Length 0: Maximum audio sample word length is 20 bits 1: Maximum audio sample word length is 24 bits

## 7.4 Audio Codec

### 7.4.1 Overview

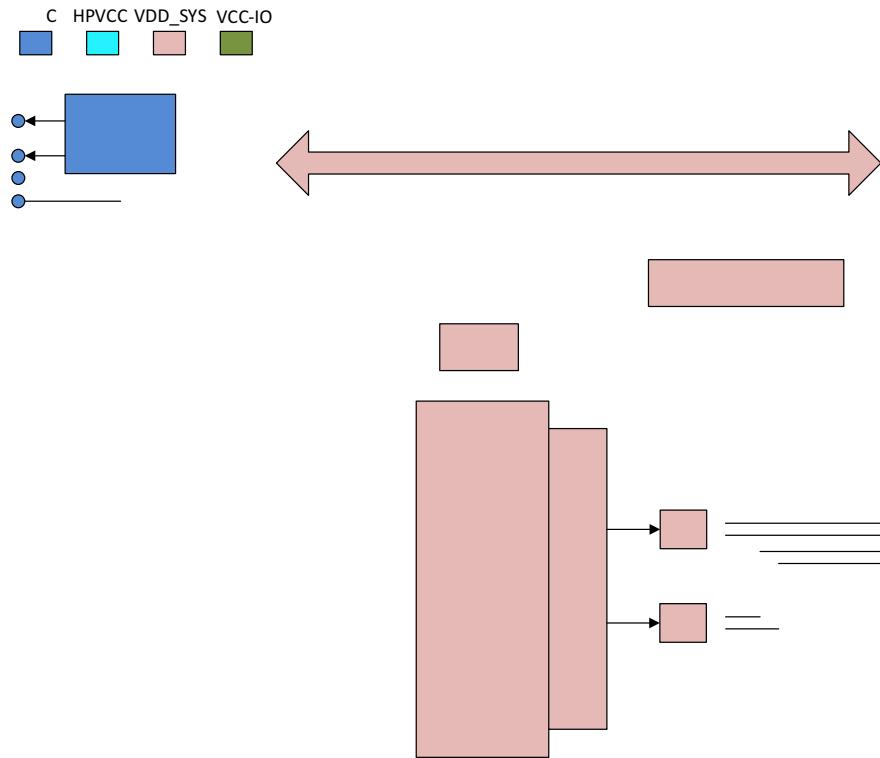
The Audio Codec is high-performance audio encoder and decoder module which supports DAC, dynamic range controller (DRC) and dynamic voltage controller (DVC) functions.

The Audio Codec has the following features:

- Two audio digital-to-analog converter (DAC) channels
  - Supports the DAC sample rate from 8 kHz to 192 kHz
  - $95 \pm 2$  dB SNR@A-weight,  $-85 \pm 3$  dB THD + N
  - Supports 16-bit and 20-bit audio sample resolution
- One audio outputs
  - One stereo headphone output: HPOUTL/R
- Supports Dynamic Range Controller (DRC) adjusting the DAC playback
- One 128x20-bit FIFO for DAC data transmit
- Programmable FIFO thresholds
- Supports interrupts and DMA
- Internal HPLDO output for HPVCC
- Internal ALDO output for AVCC

### 7.4.2 Block Diagram

The following figure shows the block diagram of Audio Codec.

**Figure 7-19 Audio Codec Block Diagram**

### 7.4.3 Functional Description

#### 7.4.3.1 External Signals

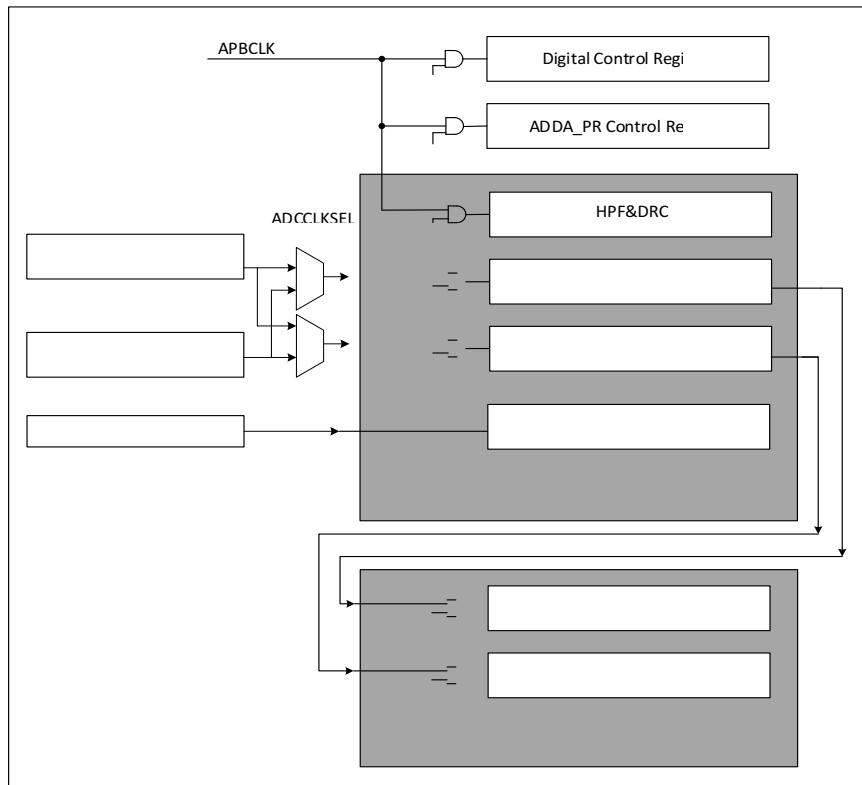
**Table 7-13 Audio Codec External Signals**

Signal	Type	Description
HPOUTL	AO	Headphone Light Output
HPOUTR	AO	Headphone Right Output
HPOUTFB	AI	Pseudo Differential Headphone Ground Reference
VRA1	AO	Internal Reference Voltage
VRA2	AO	Internal Reference Voltage
HPVCC	P	Headphone Power
AVCC	P	Analog Power
AGND	G	Analog Ground

#### 7.4.3.2 Clock Sources

The following figure describes the clock source of Audio Codec. For clock setting, configuration, and gating information, refer to section 3.3 “[CCU](#)”.

**Figure 7-20 Audio Codec Clock Diagram**



The clock source for the digital part is the PLL\_AUDIO0 and PLL\_AUDIO1. For the DAC clock, configure [AUDIO\\_CODEC\\_DAC\\_CLK\\_REG\[25:24\]](#) to select the clock source. The PK-PK jitter of PLL\_AUDIO0 and PLL\_AUDIO1 should be less than 200 ps.

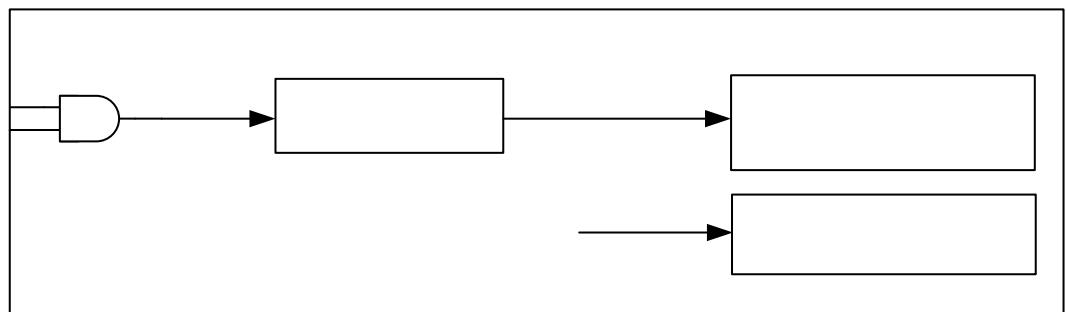
The clock source for the analog part is the CK\_DAC, of which is divided from the digital part.

### 7.4.3.3 Reset System

#### Digital Part Reset System

The following figure shows the reset system of the audio codec digital part.

Figure 7-21 Audio Codec Digital Part Reset System

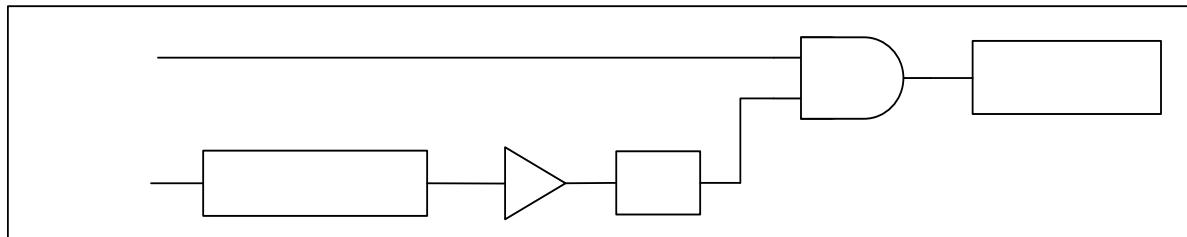


The SYS\_RST comes from the VDD-SYS domain and is produced by the RTC domain. Each domain has the debounce to confirm the reset system is strong. For the codec register part, MIX can be reset by the SYS\_RST when being powered on or the system soft is writing the reset control logic. The other parts can be reset by the soft configuration through writing the register.

#### Analog Part Reset System

The following figure shows the reset system of the audio codec analog part.

Figure 7-22 Audio Codec Analog Part Reset System

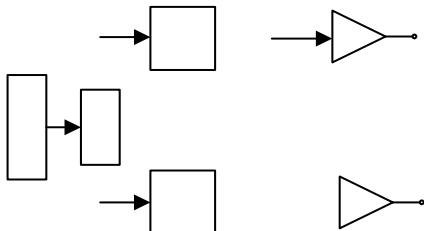


When AVCC is powered on, it sends the AVCC\_POR signal. The AVCC\_POR signal passes the level shift and RC filter part to the ADDA logic core.

#### 7.4.3.4 Data Path Diagram

The following figure shows a data path of the Audio Codec.

**Figure 7-23 Audio Codec Data Path Diagram**



#### 7.4.3.5 Stereo DAC

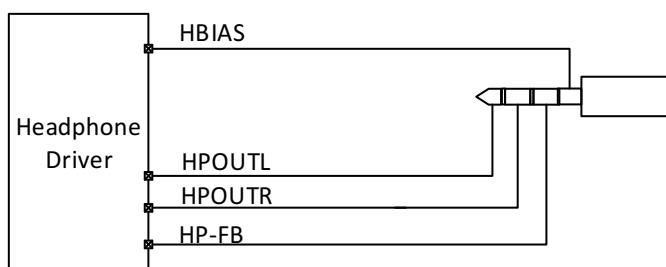
The stereo DAC sample rate can be configured by setting the register. To save power, the analog DACL can be enabled or disabled by setting the bit[15] of the [DAC\\_REG](#) register, and the analog DACR can be enabled or disabled by setting the bit[14] of the [DAC\\_REG](#) register. The digital DAC part can be enabled or disabled by the bit[31] of the [AC\\_DAC\\_DPC](#) register.

#### 7.4.3.6 Analog Audio Output Path

The Audio Codec has one analog output paths: HPOUTL/R

The headphone PA is powered up or down by HP\_REG[bit15] (HPPA\_EN). HPOUTL/R can drive a 16R or 32R headphone load without DC capacitors by using Charge Pump to generate the negative rails. HP-FB is the ground loop noise rejection feedback. HBIAS provides reference voltage for electret condenser type (ECM) microphones.

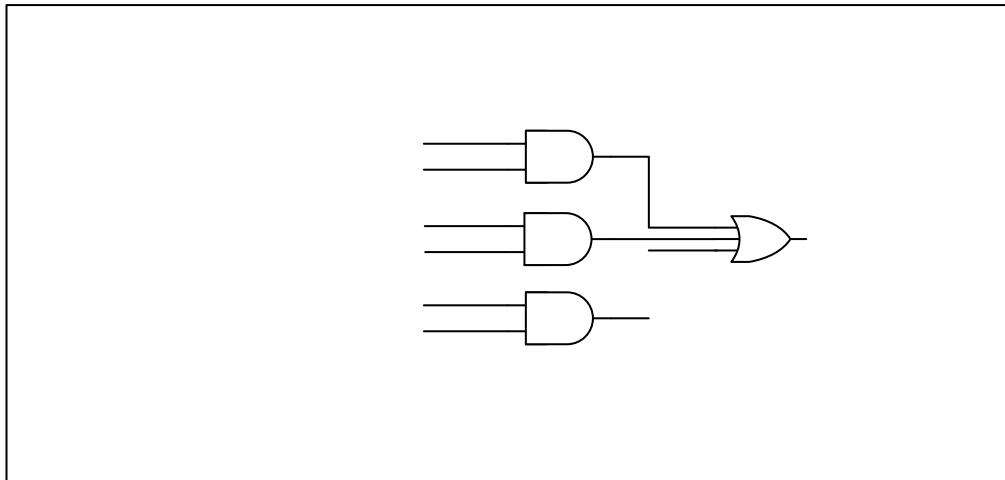
**Figure 7-24 Headphone Output Application**



#### 7.4.3.7 Interrupts

The Audio Codec has two interrupts. The following figure describes the Audio Codec interrupt system.

**Figure 7-25 Audio Codec Interrupt System**

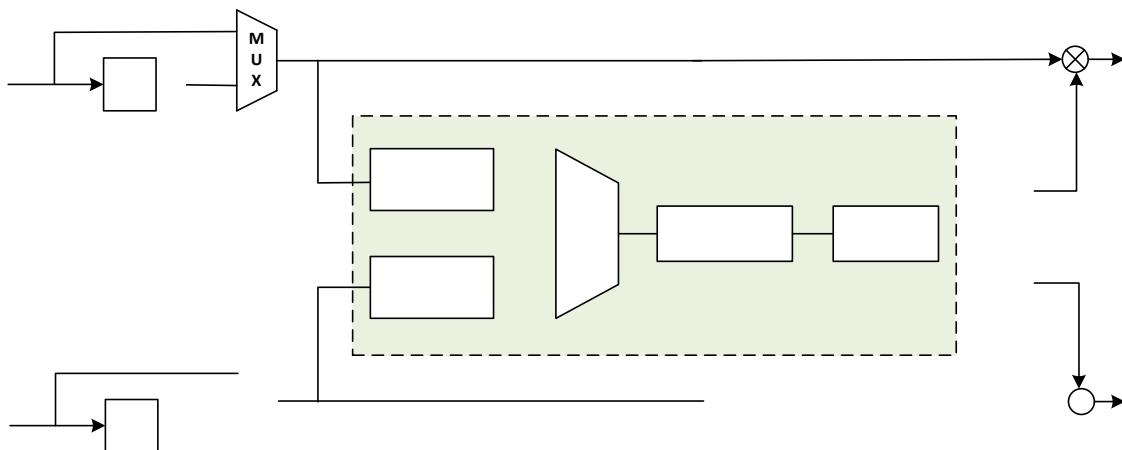


#### 7.4.3.8 Digital Audio Processor (DAP)

The DAP module is used to remove the DC offset and automatically adjusts the volume to a flatten volume level. It mainly consists of two HPF and one DRC.

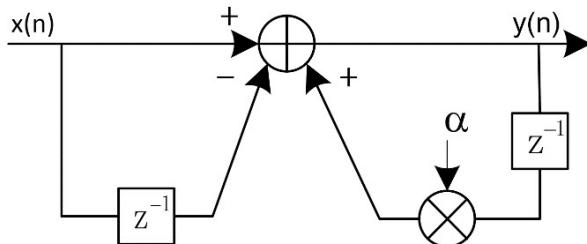
The following figure shows the DAP data flow.

**Figure 7-26 DAP Data Flow**

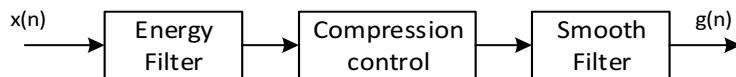
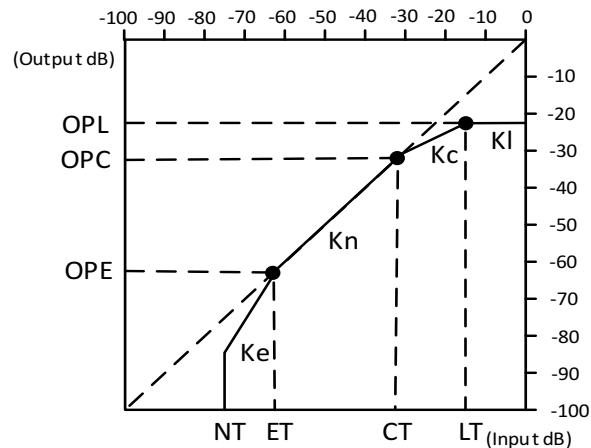


##### HPF Function

The DAP has individual channel high pass filter (HPF, -3 dB cutoff < 1 Hz) that can be enabled and disabled. The filter cutoff frequency is less than 1 Hz that can be removed DC offset. The HPF can also be bypassed.

**Figure 7-27 HPF Logic Structure****DRC Function**

The DRC scheme has three thresholds, three offsets, and four slopes (all programmable). There is one ganged DRC for the left and right channels. The following figure shows the diagram of DRC input/output.

**Figure 7-28 DRC Block Diagram****Figure 7-29 DRC Static Curve Parameters**

Professional-quality dynamic range compression automatically adjusts the volume to flatten volume level.

One DRC for left/right and one DRC for the subwoofer.

Each DRC has an adjustable threshold, offset, and compression levels, programmable energy, attack, and decay time constants.

**Transparent compression:** Compressors can attack fast enough to avoid apparent clipping before engaging, and decay times can be set slow enough to avoid pumping.

Configure the DRC parameters according to the following guidelines:

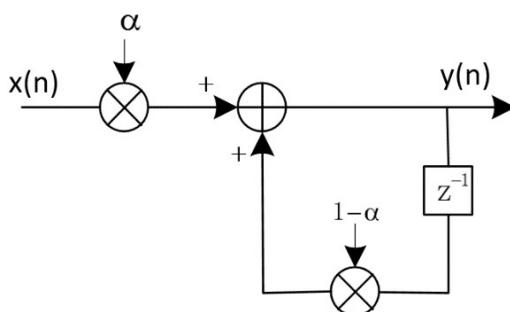
- **Number format**

The Number format is N.M which means there are N bits to the left of the decimal point including the sign bit and M bits to the right of the decimal point. For example, Numbers formatted 9.23 means that there are 9 bits at the left of the decimal point and 23 bits at the right decimal point.

- **Energy Filter**

The following figure shows the structure of the energy filter.

**Figure 7-30 Energy Filter Structure**



The Energy Filter is to estimate the RMS value of the audio data stream into DRC and has two parameters, which determine the time window over which RMS to be made. The parameter is computed by

$$\alpha = 1 - e^{-2.2T_s/ta}$$

For the Compression Control, there are ten parameters (ET, CT, LT, Ke, Kn, Kc, Kl, OPL, OPC, and OPE), which are all programmable, and the computation will be explained as follows.

- **Threshold Parameter Computation (T parameter)**

The threshold is the value that determines the signal to be compressed or not. When the RMS of the signal is larger than the threshold, the signal will be compressed. The value of threshold input to the coefficient

register is computed by  $T_{in} = -\frac{T_{dB}}{6.0206}$ .

Where,  $T_{dB}$  must less than zero, the positive value is illegal.

For example, it is desired to set CT = -40 dB, then the Tin require to set CT to -40 dB is  $CT_{in} = -(-40 \text{ dB})/6.0206 = 6.644$ ,  $CT_{in}$  is entered as a 32-bit number in 8.24 format.

Therefore,  $CT_{in} = 6.644 = 0000\ 0110.1010\ 0100\ 1101\ 0011\ 1100\ 0000 = 0x06A4\ D3C0$  in 8.24 format.

- **Slope Parameter Computation (K parameter)**

The K is the slope within the compression region. For example, an n: 1 compression means that an output increase of 1 dB is for n dB RMS input. The k input to the coefficient ram is computed by  $K = -$

Where, n is from 1 to 50, and must be an integer.

For example, it is desired to set to 2:1, then the Kc requires to set to 2:1, is  $Kc = 1/2 = 0.5$ , Kc is entered as a 32-bit number in 8.24 format.

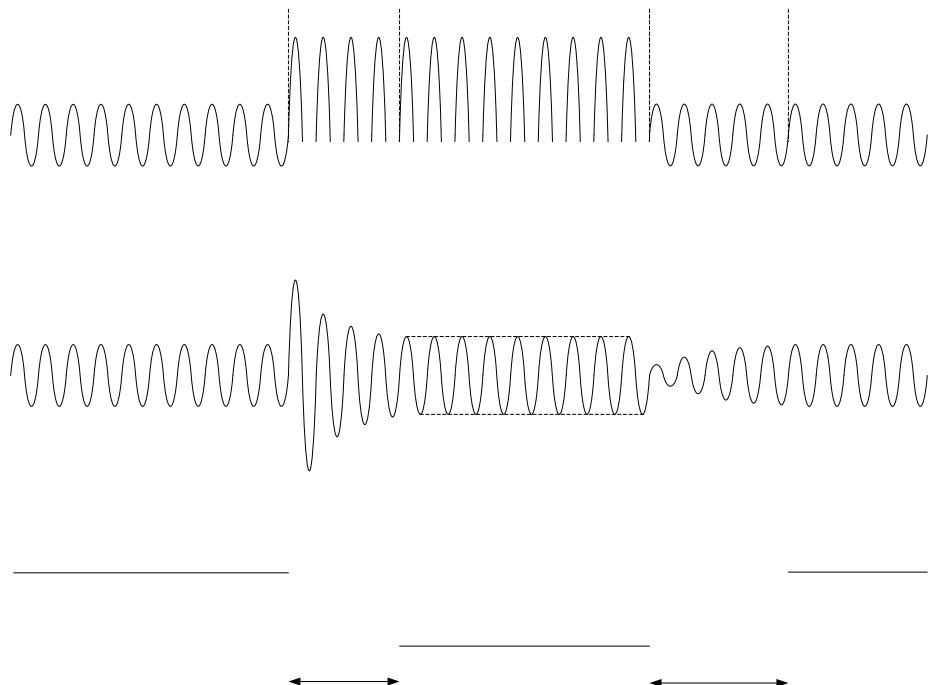
Therefore,  $Kc = 0.5 = 0000\ 0000.1000\ 0000\ 0000\ 0000\ 0000 = 0x0080\ 0000$  in 8.24 format.

- **Gain Smooth Filter**

The Gain Smooth Filter is to smooth the gain and control the ratio of gain increase and decrease. The decay time and attack are shown in Figure 8-36. The structure of the Gain Smooth filter is also the Alpha

filter, so the rise time computation is the same as the Energy filter which is  $\alpha = 1 - e^{-2.2Ts/ta}$ .

**Figure 7-31 Gain Smooth Filter**



#### 7.4.4 Programming Guidelines

##### 7.4.4.1 Playback Process

In playback mode, the audio data are transferred from the DRAM via DMA, processed by the DAC, and finally output via the analog interface.

**Step 1** Codec initialization: configure [AUDIO\\_CODEC\\_BGR\\_REG](#) to open the audio codec bus clock gating and de-assert bus reset; configure [AUDIO\\_CODEC\\_DAC\\_CLK\\_REG](#) and [PLL\\_AUDIO1\\_CTRL\\_REG](#) to configure PLL\_Audio1 frequency and enable PLL\_Audio1. For details, refer to section 3.3 “[CCU](#)”.

**Step 2** Configure the sample rate and data transfer format, then open the DAC.

**Step 3** Configure the DMA and DMA request.

**Step 4** Enable the DAC DRQ and DMA.

#### 7.4.5 Register List

Module Name	Base Address
Audio Codec	0x02030000

Register Name	Offset	Description
AC_DAC_DPC	0x0000	DAC Digital Part Control Register
DAC_VOL_CTRL	0x0004	DAC Volume Control Register
AC_DAC_FIFOC	0x0010	DAC FIFO Control Register
AC_DAC_FIFOS	0x0014	DAC FIFO Status Register
AC_DAC_TXDATA	0x0020	DAC TX DATA Register
AC_DAC_CNT	0x0024	DAC TX FIFO Counter Register
AC_DAC_DG	0x0028	DAC Debug Register
VRA1SPEEDUP_DOWN_CTRL	0x0054	VRA1 Speedup Down Control Register
AC_DAC_DAP_CTRL	0x00F0	DAC DAP Control Register
AC_DAC_DRC_HHPFC	0x0100	DAC DRC High HPF Coef Register
AC_DAC_DRC_LHPFC	0x0104	DAC DRC Low HPF Coef Register
AC_DAC_DRC_CTRL	0x0108	DAC DRC Control Register
AC_DAC_DRC_LPFHAT	0x010C	DAC DRC Left Peak Filter High Attack Time Coef Register
AC_DAC_DRC_LPFLAT	0x0110	DAC DRC Left Peak Filter Low Attack Time Coef Register
AC_DAC_DRC_RPFHAT	0x0114	DAC DRC Right Peak Filter High Attack Time Coef Register
AC_DAC_DRC_RPFLAT	0x0118	DAC DRC Peak Filter Low Attack Time Coef Register
AC_DAC_DRC_LPFHRT	0x011C	DAC DRC Left Peak Filter High Release Time Coef Register
AC_DAC_DRC_LPFLRT	0x0120	DAC DRC Left Peak Filter Low Release Time Coef Register
AC_DAC_DRC_RPFHRT	0x0124	DAC DRC Right Peak filter High Release Time Coef Register

Register Name	Offset	Description
AC_DAC_DRC_RPFLRT	0x0128	DAC DRC Right Peak filter Low Release Time Coef Register
AC_DAC_DRC_LRMSHAT	0x012C	DAC DRC Left RMS Filter High Coef Register
AC_DAC_DRC_LRMSLAT	0x0130	DAC DRC Left RMS Filter Low Coef Register
AC_DAC_DRC_RRMSHAT	0x0134	DAC DRC Right RMS Filter High Coef Register
AC_DAC_DRC_RRMSLAT	0x0138	DAC DRC Right RMS Filter Low Coef Register
AC_DAC_DRC_HCT	0x013C	DAC DRC Compressor Threshold High Setting Register
AC_DAC_DRC_LCT	0x0140	DAC DRC Compressor Slope High Setting Register
AC_DAC_DRC_HKC	0x0144	DAC DRC Compressor Slope High Setting Register
AC_DAC_DRC_LKC	0x0148	DAC DRC Compressor Slope Low Setting Register
AC_DAC_DRC_HOPC	0x014C	DAC DRC Compressor High Output at Compressor Threshold Register
AC_DAC_DRC_LOPC	0x0150	DAC DRC Compressor Low Output at Compressor Threshold Register
AC_DAC_DRC_HLT	0x0154	DAC DRC Limiter Threshold High Setting Register
AC_DAC_DRC_LLTD	0x0158	DAC DRC Limiter Threshold Low Setting Register
AC_DAC_DRC_HKI	0x015C	DAC DRC Limiter Slope High Setting Register
AC_DAC_DRC_LKI	0x0160	DAC DRC Limiter Slope Low Setting Register
AC_DAC_DRC_HOPL	0x0164	DAC DRC Limiter High Output at Limiter Threshold
AC_DAC_DRC_LOPL	0x0168	DAC DRC Limiter Low Output at Limiter Threshold
AC_DAC_DRC_HET	0x016C	DAC DRC Expander Threshold High Setting Register
AC_DAC_DRC_LET	0x0170	DAC DRC Expander Threshold Low Setting Register
AC_DAC_DRC_HKE	0x0174	DAC DRC Expander Slope High Setting Register
AC_DAC_DRC_LKE	0x0178	DAC DRC Expander Slope Low Setting Register
AC_DAC_DRC_HOPE	0x017C	DAC DRC Expander High Output at Expander Threshold
AC_DAC_DRC_LOPE	0x0180	DAC DRC Expander Low Output at Expander Threshold
AC_DAC_DRC_HKN	0x0184	DAC DRC Linear Slope High Setting Register
AC_DAC_DRC_LKN	0x0188	DAC DRC Linear Slope Low Setting Register
AC_DAC_DRC_SFHAT	0x018C	DAC DRC Smooth filter Gain High Attack Time Coef Register
AC_DAC_DRC_SFLAT	0x0190	DAC DRC Smooth filter Gain Low Attack Time Coef Register
AC_DAC_DRC_SFVRT	0x0194	DAC DRC Smooth filter Gain High Release Time Coef Register

Register Name	Offset	Description
AC_DAC_DRC_SFLRT	0x0198	DAC DRC Smooth filter Gain Low Release Time Coef Register
AC_DAC_DRC_MXGHS	0x019C	DAC DRC MAX Gain High Setting Register
AC_DAC_DRC_MXGLS	0x01A0	DAC DRC MAX Gain Low Setting Register
AC_DAC_DRC_MNGHS	0x01A4	DAC DRC MIN Gain High Setting Register
AC_DAC_DRC_MNGLS	0x01A8	DAC DRC MIN Gain Low Setting Register
AC_DAC_DRC_EPSHC	0x01AC	DAC DRC Expander Smooth Time High Coef Register
AC_DAC_DRC_EPSLC	0x01B0	DAC DRC Expander Smooth Time Low Coef Register
AC_DAC_DRC_HPFHGAIN	0x01B8	DAC DRC HPF Gain High Coef Register
AC_DAC_DRC_HPFLGAIN	0x01BC	DAC DRC HPF Gain Low Coef Register
Analog Domain Register		
DAC_REG	0x0310	DAC Analog Control Register
RAMP_REG	0x031C	BIAS Analog Control Register
BIAS_REG	0x0320	BIAS Analog Control Register

#### 7.4.6 Register Description

##### 7.4.6.1 0x0000 DAC Digital Part Control Register (Default Value: 0x0000\_0000)

Offset: 0x0000			Register Name: AC_DAC_DPC
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	EN_DA DAC Digital Part Enable 0: Disabled 1: Enabled
30:29	/	/	/
28:25	R/W	0x0	MODQU Internal DAC Quantization Levels Levels = [7*(21 + MODQU[3:0])] / 128 Default levels = 7*21/128 = 1.15
24	R/W	0x0	DWA DWA Function Disable 0: Enabled 1: Disabled
23:19	/	/	/

Offset: 0x0000			Register Name: AC_DAC_DPC
Bit	Read/Write	Default/Hex	Description
18	R/W	0x0	<p>HPF_EN High Pass Filter Enable 0: Disabled 1: Enabled</p>
17:12	R/W	0x0	<p>DVOL Digital volume control: DVC, ATT = DVC[5:0]*(-1.16 dB) 64 steps, -1.16 dB/step</p>
11:1	/	/	/
0	R/W	0x0	<p>HUB_EN Audio Hub Enable The bit takes effect only when the EN_DA is set to 1. System Domain: Audio Codec/I2S1/I2S2/OWA TXFIFO Hub Enable. 0: Disabled 1: Enabled</p>

#### 7.4.6.2 0x0004 DAC Volume Control Register (Default Value: 0x0000\_A0A0)

Offset: 0x0004			Register Name: DAC_VOL_CTRL
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	<p>DAC_VOL_SEL DAC Volume Control Selection Enable 0: Disabled 1: Enabled</p>
15:8	R/W	0xA0	<p>DAC_VOL_L DAC left channel volume (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB ... 0x9F = -0.75 dB 0xA0 = 0 dB 0xA1 = 0.75 dB ... 0xFF = 71.25 dB</p>

Offset: 0x0004			Register Name: DAC_VOL_CTRL
Bit	Read/Write	Default/Hex	Description
7:0	R/W	0xA0	<p>DAC_VOL_R DAC right channel volume (-119.25 dB To 71.25 dB, 0.75 dB/Step)</p> <p>0x00: Mute 0x01: -119.25 dB 0x9F = -0.75 dB 0xA0 = 0 dB 0xA1 = 0.75 dB ... 0xFF = 71.25 dB</p>

#### 7.4.6.3 0x0010 DAC FIFO Control Register (Default Value: 0x0000\_4000)

Offset: 0x0010			Register Name: AC_DAC_FIFOC
Bit	Read/Write	Default/Hex	Description
31:29	R/W	0x0	<p>DAC_FS Sample Rate of DAC 000: 48 kHz 010: 24 kHz 100: 12 kHz 110: 192 kHz 001: 32 kHz 011: 16 kHz 101: 8 kHz 111: 96 kHz 44.1 kHz/22.05 kHz/11.025 kHz can be supported by Audio PLL Configure Bit</p>
28	R/W	0x0	FIR_VER FIR Version 0: 64-Tap FIR 1: 32-Tap FIR
27	/	/	/

Offset: 0x0010			Register Name: AC_DAC_FIFOC
Bit	Read/Write	Default/Hex	Description
26	R/W	0x0	<p>SEND_LASAT Audio sample select when TX FIFO underrun 0: Sending zero 1: Sending the last audio sample</p>
25:24	R/W	0x0	<p>FIFO_MODE For 20-bit transmitted audio sample: 00/10: FIFO_I[19:0] = {TXDATA[31:12]} 01/11: FIFO_I[19:0] = {TXDATA[19:0]} For 16-bit transmitted audio sample: 00/10: FIFO_I[19:0] = {TXDATA[31:16], 4'b0} 01/11: FIFO_I[19:0] = {TXDATA[15:0], 4'b0}</p>
23	/	/	/
22:21	R/W	0x0	<p>DAC_DRQ_CLR_CNT When TX FIFO available room is less than or equal N, the DRQ request will be de-asserted. N is defined here: 00: IRQ/DRQ de-asserted when WLEVEL &gt; TXTL 01: 4 10: 8 11: 16</p>
20:15	/	/	/
14:8	R/W	0x40	<p>TX_TRIG_LEVEL TX FIFO Empty Trigger Level (TXTL[12:0]) Interrupt and DMA request trigger level for TX FIFO normal condition. IRQ/DRQ generated when WLEVEL ≤ TXTL <b>Note: WLEVEL represents the number of valid samples in the TX FIFO. Only TXTL[6:0] valid when TXMODE = 0</b></p>
7	/	/	/
6	R/W	0x0	<p>DAC_MONO_EN DAC Mono Enable 0: Stereo, 64 levels FIFO 1: Mono, 128 levels FIFO When enabled, L &amp; R channel send the same data.</p>

Offset: 0x0010			Register Name: AC_DAC_FIFOC
Bit	Read/Write	Default/Hex	Description
5	R/W	0x0	<p>TX_SAMPLE_BITS Transmitting Audio Sample Resolution 0: 16 bits 1: 20 bits</p>
4	R/W	0x0	<p>DAC_DRO_EN DAC FIFO Empty DRQ Enable 0: Disabled 1: Enabled</p>
3	R/W	0x0	<p>DAC_IRQ_EN DAC FIFO Empty IRQ Enable 0: Disabled 1: Enabled</p>
2	R/W	0x0	<p>FIFO_UNDERRUN_IRQ_EN DAC FIFO Underrun IRQ Enable 0: Disabled 1: Enabled</p>
1	R/W	0x0	<p>FIFO_OVERRUN_IRQ_EN DAC FIFO Overrun IRQ Enable 0: Disabled 1: Enabled</p>
0	R/WC	0x0	<p>FIFO_FLUSH DAC FIFO Flush Write '1' to flush TX FIFO, self clear to '0'</p>

#### 7.4.6.4 0x0014 DAC FIFO Status Register (Default Value: 0x0080\_8008)

Offset: 0x0014			Register Name: AC_DAC_FIFOS
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23	R	0x1	<p>TX_EMPTY TX FIFO Empty 0: No room for new sample in TX FIFO 1: More than one room for new sample in TX FIFO (&gt;= 1 word)</p>
22:8	R	0x80	<p>TXE_CNT TX FIFO Empty Space Word Counter</p>

Offset: 0x0014			Register Name: AC_DAC_FIFOS
Bit	Read/Write	Default/Hex	Description
7:4	/	/	/
3	R/W1C	0x1	<p>TXE_INT TX FIFO Empty Pending Interrupt 0: No Pending IRQ 1: FIFO Empty Pending Interrupt Write '1' to clear this interrupt or automatically clear if the interrupt condition fails.</p>
2	R/W1C	0x0	<p>TXU_INT TX FIFO Underrun Pending Interrupt 0: No Pending Interrupt 1: FIFO Underrun Pending Interrupt Write '1' to clear this interrupt</p>
1	R/W1C	0x0	<p>TXO_INT TX FIFO Overrun Pending Interrupt 0: No Pending Interrupt 1: FIFO Overrun Pending Interrupt Write '1' to clear this interrupt</p>
0	/	/	/

#### 7.4.6.5 0x0020 DAC TX DATA Register (Default Value: 0x0000\_0000)

Offset: 0x0020			Register Name: AC_DAC_TXDATA
Bit	Read/Write	Default/Hex	Description
31:0	W	0x0	<p>TX_DATA Write the transmitting left and right channel sample data to this register one by one. Write the left channel sample data first and then the right channel sample.</p>

#### 7.4.6.6 0x0024 DAC TX Counter Register (Default Value: 0x0000\_0000)

Offset: 0x0024			Register Name: AC_DAC_CNT
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<p>TX_CNT TX Sample Counter The audio sample number of sending into TXFIFO. When one sample is put into TXFIFO by DMA or by host IO, the TX sample counter register increases by one. The TX sample counter register can be set to any initial value at any time. After being updated by the initial value, the counter register should count from this initial value.</p> <p><b>Note:</b> It is used for Audio/Video Synchronization.</p>

#### 7.4.6.7 0x0028 DAC Debug Register (Default Value: 0x0000\_0000)

Offset: 0x0028			Register Name: AC_DAC_DG
Bit	Read/Write	Default/Hex	Description
31:12	/	/	/
11	R/W	0x0	<p>DAC_MODU_SELECT DAC Modulator Debug 0: DAC Modulator Normal Mode 1: DAC Modulator Debug Mode</p>
10:9	R/W	0x0	<p>DAC_PATTERN_SELECT DAC Pattern Select 00: Normal (Audio sample from TX FIFO) 01: -6 dB Sin wave 10: -60 dB Sin wave 11: Silent wave</p>
8	R/W	0x0	<p>CODEC_CLK_SELECT CODEC Clock Source Select 0: CODEC clock from PLL 1: CODEC clock from OSC (for Debug)</p>
7	/	/	/
6	R/W	0x0	<p>DA_SWP DAC Output Channel Swap Enable 0: Disabled 1: Enabled</p>

Offset: 0x0028			Register Name: AC_DAC_DG
Bit	Read/Write	Default/Hex	Description
5:3	/	/	/
2:0	R/W	0x0	ADDA_LOOP_MODE ADDA Loop Mode Select 000: Disabled Others: Reserved

#### 7.4.6.8 0x0054 VRA1 Speedup Down Control Register (Default Value: 0x0000\_0000)

Offset: 0x0054			Register Name: VRA1SPEEDUP_DOWN_CTRL
Bit	Read/Write	Default/Hex	Description
31:5	/	/	/
4	R	0x0	VRA1SPEEDUP_DOWN_STATE Only if VAR1SPEEDUP_DOWN_Further_CTRL (0x310[22]) is set 0, VAR1Speedup Down State is valid. 0: VAR1Speedup_Down does not work. 1: VAR1Speedup_Down works.
3:2	/	/	/
1	R/W	0x0	VRA1SPEEDUP_DOWN_CTRL VAR1Speedup Down Manual Control Enable 0: Disabled. VAR1Speedup Down converts to 1 after the bus rst releases 32 ms. 1: Enabled. VAR1Speedup Down converts to 1 immediately.
0	R/W	0x0	VRA1SPEEDUP_DOWN_RST_CTRL VAR1Speedup Down RST Manual Control Enable 0: Disabled. VAR1Speedup Down converts to 1 after the bus rst releases 32 ms. 1: Enabled. VAR1Speedup Down reset 0 immediately.

#### 7.4.6.9 0x00F0 DAC DAP Control Register (Default Value: 0x0000\_0000)

Offset: 0x00F0			Register Name: AC_DAC_DAP_CTR
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	DDAP_EN DAP for DRC enable 0: Bypassed 1: Enabled
30	/	/	/
29	R/W	0x0	DDAP_DRC_EN DRC enable control 0: Disabled 1: Enabled
28	R/W	0x0	DDAP_HPF_EN HPF enable control 0: Disabled 1: Enabled
27:0	/	/	/

#### 7.4.6.10 0x0100 DAC DRC High HPF Coef Register (Default Value: 0x0000\_00FF)

Offset: 0x0100			Register Name: AC_DAC_DRC_HHPFC
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0xFF	HPF coefficient setting and the data is 3.24 format.

#### 7.4.6.11 0x0104 DAC DRC Low HPF Coef Register (Default Value: 0x0000\_FAC1)

Offset: 0x0104			Register Name: AC_DAC_DRC_LHPFC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xFAC1	HPF coefficient setting and the data is 3.24 format.

#### 7.4.6.12 0x0108 DAC DRC Control Register (Default Value: 0x0000\_0080)

Offset: 0x0108			Register Name: AC_DAC_DRC_CTRL
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15	R	0x0	DRC delay buffer data output state when The DRC delay function is enabled and the DRC function is disabled. After disabling the DRC function and this bit goes to 0, write the DRC delay function bit to 0. 0: Not completed 1: Completed
14:10	/	/	/
13:8	R/W	0x0	Signal delay time setting 6'h00: (8x1) fs 6'h01: (8x2) fs 6'h02: (8x3) fs ----- 6'h2e: (8*47) fs 6'h2f: (8*48) fs 6'h30 -- 6'h3f: (8*48) fs Delay time = 8*(n + 1) fs, n<6'h30; When the delay function is disabled, the signal delay time is unused.
7	R/W	0x1	DAC_DRC_DELAY_BUF_EN The delay buffer use or not when the DRC is disabled and the DRC buffer data output completely. 0: Do not use the buffer. 1: Use the buffer.
6	R/W	0x0	DAC_DRC_GAIN_MAX_LIMIT_EN DRC gain max limit enable 0: Disabled 1: Enabled
5	R/W	0x0	DAC_DRC_GAIN_MIN_LIMIT_EN DRC gain min limit enable When this function is enabled, it will overwrite the noise detect function. 0: Disabled 1: Enabled

Offset: 0x0108			Register Name: AC_DAC_DRC_CTRL
Bit	Read/Write	Default/Hex	Description
4	R/W	0x0	<p>DAC_DRC_DETECT_NOISE_EN Control the DRC to detect noise when ET is enabled. 0: Disabled 1: Enabled</p>
3	R/W	0x0	<p>DAC_DRC_SIGNAL_FUNC_SEL Signal function select 0: RMS filter 1: Peak filter When the signal function selects the Peak filter, the RMS parameter is unused. (AC_DRC_LRMESHAT, AC_DRC_LRMSLAT, AC_DRC_LRMESHAT, AC_DRC_LRMSLAT) When the signal function selects the RMS filter, the Peak filter parameter is unused. (AC_DRC_LPFHAT, AC_DRC_LPFLAT, AC_DRC_RPFHAT, AC_DRC_RPFLAT, AC_DRC_LPFHRT, AC_DRC_LPFLRT, AC_DRC_RPFHRT, and AC_DRC_RPFLRT)</p>
2	R/W	0x0	<p>DAC_DRC_DELAY_FUNC_EN Delay function enable 0: Disabled 1: Enabled When the bit is disabled, the signal delay time is unused.</p>
1	R/W	0x0	<p>DAC_DRC_LT_EN DRC LT enable 0: Disabled 1: Enabled When the bit is disabled, KI and OPL parameter is unused.</p>
0	R/W	0x0	<p>DAC_DRC_ET_EN DRC ET enable 0: Disabled 1: Enabled When the bit is disabled, Ke and OPE parameter is unused.</p>

#### 7.4.6.13 0x010C DAC DRC Left Peak Filter High Attack Time Coef Register (Default Value: 0x0000\_000B)

Offset: 0x010C			Register Name: AC_DAC_DRC_LPFHAT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/

Offset: 0x010C			Register Name: AC_DAC_DRC_LPFHAT
Bit	Read/Write	Default/Hex	Description
10:0	R/W	0x000B	DAC_DRC_LPFHAT The left peak filter attack time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/ta). The format is 3.24. (The default value is 1 ms)

#### 7.4.6.14 0x0110 DAC DRC Left Peak Filter Low Attack Time Coef Register (Default Value: 0x0000\_77BF)

Offset: 0x0110			Register Name: AC_DAC_DRC_LPFLAT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x77BF	DAC_DRC_LPFLAT The left peak filter attack time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/ta). The format is 3.24. (The default value is 1 ms)

#### 7.4.6.15 0x0114 DAC DRC Right Peak Filter High Attack Time Coef Register (Default Value: 0x0000\_000B)

Offset: 0x0114			Register Name: AC_DAC_DRC_RPFHAT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xB	DAC_DRC_RPFHAT The right peak filter attack time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/ta). The format is 3.24. (The default value is 1 ms)

#### 7.4.6.16 0x0118 DAC DRC Left Peak Filter Low Attack Time Coef Register (Default Value: 0x0000\_77BF)

Offset: 0x0118			Register Name: AC_DAC_DRC_RPFLAT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/

Offset: 0x0118			Register Name: AC_DAC_DRC_RPFLAT
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0x77BF	DAC_DRC_RPFLAT The left peak filter attack time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/ta). The format is 3.24. (The default value is 1 ms)

#### 7.4.6.17 0x011C DAC DRC Left Peak Filter High Release Time Coef Register (Default Value: 0x0000\_00FF)

Offset: 0x011C			Register Name: AC_DAC_DRC_LPFHRT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x00FF	DAC_DRC_LPFHRT The left peak filter release time parameter setting, which is determined by the equation that RT = exp (-2.2Ts/tr). The format is 3.24. (The default value is 100 ms)

#### 7.4.6.18 0x0120 DAC DRC Left Peak Filter Low Release Time Coef Register (Default Value: 0x0000\_E1F8)

Offset: 0x0120			Register Name: AC_DAC_DRC_LPFLRT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xE1F8	DAC_DRC_LPFLRT The left peak filter release time parameter setting, which is determined by the equation that RT = exp (-2.2Ts/tr). The format is 3.24. (The default value is 100 ms)

#### 7.4.6.19 0x0124 DAC DRC Right Peak Filter High Release Time Coef Register (Default Value: 0x0000\_00FF)

Offset: 0x0124			Register Name: AC_DAC_DRC_RPFHRT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/

Offset: 0x0124			Register Name: AC_DAC_DRC_RPFHRT
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0xFF	DAC_DRC_RPFHRT The right peak filter release time parameter setting, which is determined by the equation that RT = exp (-2.2Ts/tr). The format is 3.24. (The default value is 100 ms)

#### 7.4.6.20 0x0128 DAC DRC Right Peak filter Low Release Time Coef Register(Default Value: 0x0000\_E1F8)

Offset: 0x0128			Register Name: AC_DAC_DRC_RPFLRT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xE1F8	DAC_DRC_RPFLRT The right peak filter release time parameter setting, which is determined by the equation that AT = exp(-2.2Ts/tr). The format is 3.24. (The default value is 100 ms)

#### 7.4.6.21 0x012C DAC DRC Left RMS Filter High Coef Register (Default Value: 0x0000\_0001)

Offset: 0x012C			Register Name: AC_DAC_DRC_LRMSHAT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x0001	DAC_DRC_LRMSHAT The left RMS filter average time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/tav). The format is 3.24. (The default value is 10 ms)

#### 7.4.6.22 0x0130 DAC DRC Left RMS Filter Low Coef Register (Default Value: 0x0000\_2BAF)

Offset: 0x0130			Register Name: AC_DAC_DRC_LRMSLAT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/

Offset: 0x0130			Register Name: AC_DAC_DRC_LRMSLAT
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0x2BAF	DAC_DRC_LRMSLAT The left RMS filter average time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/tav). The format is 3.24. (The default value is 10 ms)

#### 7.4.6.23 0x0134 DAC DRC Right RMS Filter High Coef Register (Default Value: 0x0000\_0001)

Offset: 0x0134			Register Name: AC_DAC_DRC_RRMSHAT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x2BAF	DAC_DRC_RRMSHAT The right RMS filter average time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/tav). The format is 3.24. (The default value is 10 ms)

#### 7.4.6.24 0x0138 DAC DRC Right RMS Filter Low Coef Register (Default Value: 0x0000\_2BAF)

Offset: 0x0138			Register Name: AC_DAC_DRC_RRMSLAT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x2BAF	DAC_DRC_RRMSLAT The right RMS filter average time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/tav). The format is 3.24. (The default value is 10 ms)

#### 7.4.6.25 0x013C DAC DRC Compressor Threshold High Setting Register (Default Value: 0x0000\_06A4)

Offset: 0x013C			Register Name: AC_DAC_DRC_HCT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/

Offset: 0x013C			Register Name: AC_DAC_DRC_HCT
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0x06A4	DAC_DRC_HCT The compressor threshold setting, which is set by the equation that $CTin = -CT/6.0206$ . The format is 8.24. (The default value is -40 dB)

#### 7.4.6.26 0x0140 DAC DRC Compressor Slope High Setting Register (Default Value: 0x0000\_D3C0)

Offset: 0x0140			Register Name: AC_DAC_DRC_LCT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xD3C0	DAC_DRC_LCT The compressor threshold setting, which is set by the equation that $CTin = -CT/6.0206$ . The format is 8.24. (The default value is -40 dB)

#### 7.4.6.27 0x0144 DAC DRC Compressor Slope High Setting Register (Default Value: 0x0000\_0080)

Offset: 0x0144			Register Name: AC_DAC_DRC_HKC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0080	DAC_DRC_HKC The slope of the compressor, which is determined by the equation that $Kc = 1/R$ . R is the ratio of the compressor, which is always an integer. The format is 8.24. (The default value is 2:1)

#### 7.4.6.28 0x0148 DAC DRC Compressor Slope Low Setting Register (Default Value: 0x0000\_0000)

Offset: 0x0148			Register Name: AC_DAC_DRC_LKC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/

Offset: 0x0148			Register Name: AC_DAC_DRC_LKC
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0x0000	DAC_DRC_LKC The slope of the compressor, which is determined by the equation that $K_c = 1/R$ . R is the ratio of the compressor, which is always an integer. The format is 8.24. (The default value is 2:1)

#### 7.4.6.29 0x014C DAC DRC Compressor High Output at Compressor Threshold Register (Default Value: 0x0000\_F95B)

Offset: 0x014C			Register Name: AC_DAC_DRC_HOPC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xF95B	DAC_DRC_HOPC The output of the compressor, which is determined by the equation $-OPC/6.0206$ . The format is 8.24 (The default value is -40 dB)

#### 7.4.6.30 0x0150 DAC DRC Compressor Low Output at Compressor Threshold Register (Default Value: 0x0000\_2C3F)

Offset: 0x0150			Register Name: AC_DAC_DRC_LOPC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x2C3F	DAC_DRC_LOPC The output of the compressor, which is determined by the equation $OPC/6.0206$ . The format is 8.24. (The default value is -40 dB)

#### 7.4.6.31 0x0154 DAC DRC Limiter Threshold High Setting Register (Default Value: 0x0000\_01A9)

Offset: 0x0154			Register Name: AC_DAC_DRC_HLT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x01A9	DAC_DRC_HLT The limiter threshold setting, which is set by the equation that $L_{Tin} = -LT/6.0206$ , The format is 8.24. (The default value is -10 dB)

#### 7.4.6.32 0x0158 DAC DRC Limiter Threshold Low Setting Register (Default Value: 0x0000\_34F0)

Offset: 0x0158			Register Name: AC_DAC_DRC_LLT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x34F0	DAC_DRC_LLT The limiter threshold setting, which is set by the equation that LTin = -LT/6.0206. The format is 8.24. (The default value is -10 dB)

#### 7.4.6.33 0x015C DAC DRC Limiter Slope High Setting Register (Default Value: 0x0000\_0005)

Offset: 0x015C			Register Name: AC_DAC_DRC_HKI
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0005	DAC_DRC_HKI The slope of the limiter which is determined by the equation that KI = 1/R. R is the ratio of the limiter, which is always an integer. The format is 8.24. (The default value is <50:1>)

#### 7.4.6.34 0x0160 DAC DRC Limiter Slope Low Setting Register (Default Value: 0x0000\_1EB8)

Offset: 0x0160			Register Name: AC_DAC_DRC_LKI
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x1EB8	DAC_DRC_LKI The slope of the limiter, which is determined by the equation that KI = 1/R. R is the ratio of the limiter, which is always an integer. The format is 8.24. (The default value is <50:1>)

**7.4.6.35 0x0164 DAC DRC Limiter High Output at Limiter Threshold Register (Default Value: 0x0000\_FBD8)**

Offset: 0x0164			Register Name: AC_DAC_DRC_HOPL
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xFBD8	DAC_DRC_HOPL The output of the limiter, which is determined by equation OPT/6.0206. The format is 8.24. (The default value is -25 dB)

**7.4.6.36 0x0168 DAC DRC Limiter Low Output at Limiter Threshold Register (Default Value: 0x0000\_FBA7)**

Offset: 0x0168			Register Name: AC_DAC_DRC_LOPL
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xFBA7	DAC_DRC_LOPL The output of the limiter, which is determined by equation OPT/6.0206. The format is 8.24. (The default value is -25 dB)

**7.4.6.37 0x016C DAC DRC Expander Threshold High Setting Register (Default Value: 0x0000\_0BA0)**

Offset: 0x016C			Register Name: AC_DAC_DRC_HET
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0BA0	DAC_DRC_HET The expander threshold setting, which is set by the equation that ETin = -ET/6.0206. The format is 8.24. (The default value is -70 dB)

**7.4.6.38 0x0170 DAC DRC Expander Threshold Low Setting Register (Default Value: 0x0000\_7291)**

Offset: 0x0170			Register Name: AC_DAC_DRC LET
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x7291	DAC_DRC LET The expander threshold setting, which is set by the equation that ETin = -ET/6.0206. The format is 8.24. (The default value is -70 dB)

**7.4.6.39 0x0174 DAC DRC Expander Slope High Setting Register (Default Value: 0x0000\_0500)**

Offset: 0x0174			Register Name: AC_DAC_DRC_HKE
Bit	Read/Write	Default/Hex	Description
31:14	/	/	/
13:0	R/W	0x0500	DAC_DRC_HKE The slope of the expander, which is determined by the equation that $Ke = 1/R$ . R is the ratio of the expander, which is always an integer and the ke must larger than 50. The format is 8.24. (The default value is <1:5>)

**7.4.6.40 0x0178 DAC DRC Expander Slope Low Setting Register (Default Value: 0x0000\_0000)**

Offset: 0x0178			Register Name: AC_DAC_DRC_LKE
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0000	DAC_DRC_LKE The slope of the expander, which is determined by the equation that $Ke = 1/R$ . R is the ratio of the expander, which is always an integer and the ke must larger than 50. The format is 8.24. (The default value is <1:5>)

**7.4.6.41 0x017C DAC DRC Expander High Output at Expander Threshold Register (Default Value: 0x0000\_F45F)**

Offset: 0x017C			Register Name: AC_DAC_DRC_HOPE
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xF45F	DAC_DRC_HOPE The output of the expander, which is determined by equation $OPE/6.0206$ . The format is 8.24. (The default value is -70 dB)

**7.4.6.42 0x0180 DAC DRC Expander Low Output at Expander Threshold Register (Default Value: 0x0000\_8D6E)**

Offset: 0x0180			Register Name: AC_DAC_DRC_LOPE
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x8D6E	DAC_DRC_LOPE The output of the expander which is determined by equation OPE/6.0206. The format is 8.24. (The default value is -70 dB)

**7.4.6.43 0x0184 DAC DRC Linear Slope High Setting Register (Default Value: 0x0000\_0100)**

Offset: 0x0184			Register Name: AC_DAC_DRC_HKN
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0100	DAC_DRC_HKN The slope of the linear, which is determined by the equation that $K_n = 1/R$ . R is the ratio of the linear, which is always an integer. The format is 8.24. (The default value is <1:1>)

**7.4.6.44 0x0188 DAC DRC Linear Slope Low Setting Register (Default Value: 0x0000\_0000)**

Offset: 0x0188			Register Name: AC_DAC_DRC_LKN
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0000	DAC_DRC_LKN The slope of the linear, which is determined by the equation that $K_n = 1/R$ . R is the ratio of the linear, which is always an integer. The format is 8.24. (The default value is <1:1>)

**7.4.6.45 0x018C DAC DRC Smooth Filter Gain High Attack Time Coef Register (Default Value: 0x0000\_0002)**

Offset: 0x018C			Register Name: AC_DAC_DRC_SFHAT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/

Offset: 0x018C			Register Name: AC_DAC_DRC_SFHAT
Bit	Read/Write	Default/Hex	Description
10:0	R/W	0x0002	DAC_DRC_SFHAT The smooth filter attack time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/tr). The format is 3.24. (The default value is 5 ms)

#### 7.4.6.46 0x0190 DAC DRC Smooth Filter Gain Low Attack Time Coef Register (Default Value: 0x0000\_5600)

Offset: 0x0190			Register Name: AC_DAC_DRC_SFLAT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x5600	DAC_DRC_SFLAT The smooth filter attack time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/tr). The format is 3.24. (The default value is 5 ms)

#### 7.4.6.47 0x0194 DAC DRC Smooth Filter Gain High Release Time Coef Register (Default Value: 0x0000\_0000)

Offset: 0x0194			Register Name: AC_DAC_DRC_SFVRT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x0000	DAC_DRC_SFVRT The gain smooth filter release time parameter setting, which is determined by the equation that RT = 1-exp (-2.2Ts/tr). The format is 3.24. (The default value is 200 ms)

#### 7.4.6.48 0x0198 DAC DRC Smooth Filter Gain Low Release Time Coef Register (Default Value: 0x0000\_0F04)

Offset: 0x0198			Register Name: AC_DAC_DRC_SFLRT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/

Offset: 0x0198			Register Name: AC_DAC_DRC_SFLRT
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0x0F04	DAC_DRC_SFLRT The gain smooth filter release time parameter setting, which is determined by the equation that $RT = 1 - \exp(-2.2Ts/tr)$ . The format is 3.24. (The default value is 200 ms)

#### 7.4.6.49 0x019C DAC DRC MAX Gain High Setting Register (Default Value: 0x0000\_FE56)

Offset: 0x019C			Register Name: AC_DAC_DRC_MXGHS
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xFE56	DAC_DRC_MXGHS The max gain setting, which is determined by equation $MXG_{in} = MXG / 6.0206$ . The format is 8.24 and must $-20 \text{ dB} < MXG < 30 \text{ dB}$ (The default value is -10 dB)

#### 7.4.6.50 0x01A0 DAC DRC MAX Gain Low Setting Register (Default Value: 0x0000\_CBOF)

Offset: 0x01A0			Register Name: AC_DAC_DRC_MXGLS
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xCBOF	DAC_DRC_MXGLS The max gain setting, which is determined by equation $MXG_{in} = MXG / 6.0206$ . The format is 8.24 and must $-20 \text{ dB} < MXG < 30 \text{ dB}$ (The default value is -10 dB)

#### 7.4.6.51 0x01A4 DAC DRC MIN Gain High Setting Register (Default Value: 0x0000\_F95B)

Offset: 0x01A4			Register Name: AC_DAC_DRC_MNGHS
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/

Offset: 0x01A4			Register Name: AC_DAC_DRC_MNGHS
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0xF95B	DAC_DRC_MNGHS The min gain setting, which is determined by equation $MXG_{in}=MXG/6.0206$ . The format is 8.24 and must $-60 \text{ dB} \leq MNG \leq -40 \text{ dB}$ (The default value is -40 dB)

#### 7.4.6.52 0x01A8 DAC DRC MIN Gain Low Setting Register (Default Value: 0x0000\_2C3F)

Offset: 0x01A8			Register Name: AC_DAC_DRC_MNGLS
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x2C3F	DAC_DRC_MNGLS The min gain setting, which is determined by equation $MXG_{in}=MNG/6.0206$ . The format is 8.24 and must $-60 \text{ dB} \leq MNG \leq -40 \text{ dB}$ (The default value is -40 dB)

#### 7.4.6.53 0x01AC DAC DRC Expander Smooth Time High Coef Register (Default Value: 0x0000\_0000)

Offset: 0x01AC			Register Name: AC_DAC_DRC_EPSHC
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x0000	DAC_DRC_EPSHC The gain smooth filter release and attack time parameter setting in expander region, which are determined by the equation that $RT = 1-\exp(-2.2Ts/tr)$ . The format is 3.24. (The default value is 30 ms)

#### 7.4.6.54 0x01B0 DAC DRC Expander Smooth Time Low Coef Register (Default Value: 0x0000\_640C)

Offset: 0x01B0			Register Name: AC_DAC_DRC_EPSLC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/

Offset: 0x01B0			Register Name: AC_DAC_DRC_EPSLC
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0x640C	DAC_DRC_EPSLC The gain smooth filter release and attack time parameter setting in expander region, which are determined by the equation that $RT = 1 - \exp(-2.2Ts/tr)$ . The format is 3.24. (The default value is 30 ms)

#### 7.4.6.55 0x01B8 DAC DRC HPF Gain High Coef Register (Default Value: 0x0000\_0100)

Offset: 0x01B8			Register Name: AC_DAC_DRC_HPFHGAIN
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x100	DAC_DRC_HPFHGAIN The gain of HPF coefficient. The format is 3.24. (gain = 1)

#### 7.4.6.56 0x01BC DAC DRC HPF Gain Low Coef Register (Default Value: 0x0000\_0000)

Offset: 0x01BC			Register Name: AC_DAC_DRC_HPFLGAIN
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0000	DAC_DRC_HPFLGAIN The gain of HPF coefficient. The format is 3.24. (gain = 1)

#### 7.4.6.57 0x0310 DAC Analog Control Register (Default Value: 0x0015\_0000)

Offset: 0x0310			Register Name: DAC_REG
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23	R/W	0x0	CURRENT_TEST_SELECT Internal Current Sink Test Enable (from MICIN3P pin) 0: Normal 1: For Debug
22	/	/	/

Offset: 0x0310			Register Name: DAC_REG
Bit	Read/Write	Default/Hex	Description
21:20	R/W	0x1	IOPVRS VRA2 Buffer OP and Headphone Feedback Buffer OP Bias Current Select 00: 6 uA 01: 7 uA 10: 8 uA 11: 9 uA
19:18	R/W	0x1	ILINEOUTAMPS LINEOUTLL/R AMP Bias Current Select 00: 6 uA 01: 7 uA 10: 8 uA 11: 9 uA
17:16	R/W	0x1	IOPDACS OPDACL/R Bias Current Select 00: 6 uA 01: 7 uA 10: 8 uA 11: 9 uA
15	R/W	0x0	DACL_EN DACL Enable 0: Disabled 1: Enabled
14	R/W	0x0	DACR_EN DACR Enable 0: Disabled 1: Enabled
13	R/W	0x0	LINEOUTLEN Left Channel LINEOUT Enable 0: Disable 1: Enable
12	R/W	0x0	LMUTE DACL to Left Channel LINEOUT Mute Control 0: Mute 1: Not mute

Offset: 0x0310			Register Name: DAC_REG
Bit	Read/Write	Default/Hex	Description
11	R/W	0x0	<p>LINEOUTREN Right Channel LINEOUT Enable 0: Disable 1: Enable</p>
10	R/W	0x0	<p>RMUTE DACR to Right Channel LINEOUT Mute Control 0: Mute 1: Not mute</p>
9:7	/	/	/
6	R/W	0x0	<p>LINEOUTL_DIFFEN Left Channel LINEOUT Output Control 0: Single-End 1: Differential</p>
5	R/W	0x0	<p>LINEOUTR_DIFFEN Right Channel LINEOUT Output Control 0: Single-End 1: Differential</p>
4:0	R/W	0x0	<p>LINEOUT_VOL_CTRL LINEOUT Volume Control. Total 30 level from 0x1F to 0x02 with the volume 0 dB to -43.5 dB, -1.5 dB/step, mute when 00000 &amp; 00001.</p>

#### 7.4.6.58 0x031C Ramp Control Register (Default Value: 0x0018\_0000)

Offset: 0x031C			Register Name: RAMP_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<p>RAMP_RISE_INT_EN RAMP Rise Interrupt Enable 0: Enabled 1: Disabled</p>
30	R/W1C	0x0	<p>RAMP_RISE_INT RK Increase Upward Finish and Rampen Pull Down Instruction 0: No Pending IRQ 1: Ramp Rise Finish Pending Interrupt Write '1' to clear this interrupt.</p>

Offset: 0x031C			Register Name: RAMP_REG
Bit	Read/Write	Default/Hex	Description
29	R/W	0x0	<p>RAMP_FALL_INT_EN RAMP Fall Int Enable 0: Enabled 1: Disabled</p>
28	R/W1C	0x0	<p>RAMP_FALL_INT RK Downward Decrease Finish and Rampen Pull Down Instruction 0: No Pending IRQ 1: Ramp Fall Finish Pending Interrupt Write '1' to clear this interrupt.</p>
27:25	/	/	/
24	R/W	0x0	<p>RAMP_SRST Ramp Soft Reset 0: Disabled 1: Enabled</p>
23:21	/	/	/
20:16	R/W	0x18	<p>RAMP_CLK_DIV_M Analog Ramp Clk Div Freq Value : M (from 0 to 31, Default: 24). Ana_Ramp_Clk= 24MHz/(M+1) Default Ramp Clk Freq: 24MHz/(24+1)=960 kHz</p>
15	R/W	0x0	<p>HP_PULL_OUT_EN Headphone Pullout Enable 0: Disabled 1: Enabled</p>

Offset: 0x031C			Register Name: RAMP_REG
Bit	Read/Write	Default/Hex	Description
14:12	R/W	0x0	<p>RAMP_HOLD_STEP RAMP HOLD STEP</p> <p>000: 9600 001: 19200 010: 38400 011: 76800 100: 96000 101: 115200 110: 153600 111: 192000</p> <p>Ramp Hold Time = Ramp Hold Step/Ramp Clk Freq When Ramp Clk Freq is equal to 960 kHz, the corresponding Ramp Hold time of each gear is:</p> <p>000: 9600/960 kHz=10 ms 001: 19200/960 kHz=20 ms 010: 38400/960 kHz=40 ms 011: 76800/960 kHz=80 ms 100: 96000/960 kHz=100 ms 101: 115200/960 kHz=120 ms 110: 153600/960 kHz=160 ms 111: 192000/960 kHz=200 ms</p>
11:10	/	/	/
9:8	R/W	0x0	<p>GAP_STEP Gap Step</p> <p>00: ramp step 01: ramp step*2 10: ramp step*3 11: ramp step*4</p>
7	/	/	/

Offset: 0x031C			Register Name: RAMP_REG
Bit	Read/Write	Default/Hex	Description
6:4	R/W	0x0	<p>RAMP_STEP RK Frequency Gear, Control Ramp Rise/Fall Total Time 000: 20 001: 30 010: 40 011: 60 100: 80 101: 120 110: 160 111: 240</p> <p>Ramp Rise/Fall Total Time =(Ramp Step/Ramp Clk Freq)*4096 When Default Ramp Clk Freq is equal to 960 kHz, the corresponding time of each gear is: 000: <math>(20/960\text{kHz})*4096=85.3\text{ ms}</math> 001: <math>(30/960\text{kHz})*4096=128\text{ ms}</math> 010: <math>(40/960\text{kHz})*4096=170.6\text{ ms}</math> 011: <math>(60/960\text{kHz})*4096=256\text{ ms}</math> 100: <math>(80/960\text{kHz})*4096=341.3\text{ ms}</math> 101: <math>(120/960\text{kHz})*4096=512\text{ ms}</math> 110: <math>(160/960\text{kHz})*4096=682.6\text{ ms}</math> 111: <math>(240/960\text{kHz})*4096=1024\text{ ms}</math></p>
3	R/W	0x0	<p>RMD_EN Ramp Manual Down Enable 0: Disabled 1: Enabled</p>
2	R/W	0x0	<p>RMU_EN Ramp Manual Up Enable 0: Disabled 1: Enabled</p>
1	R/W	0x0	<p>RMC_EN Ramp Manual Control Enable 0: Disabled 1: Enabled</p>

Offset: 0x031C			Register Name: RAMP_REG
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	<p>RD_EN</p> <p>Ramp Digital Enable</p> <p>0: Disabled</p> <p>1: Enabled</p>

#### 7.4.6.59 0x0320 BIAS Analog Control Register (Default Value: 0x0000\_0080)

The register is not controlled by the clock and reset of Audio Codec, only controlled by the clock and reset of system bus.

Offset: 0x0320			Register Name: BIAS_REG
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R/W	0x80	<p>BIASDATA</p> <p>Bias Current Register Setting Data</p>

#### 7.4.6.60 0x0328 HMIC Control Register (Default Value: 0x0000\_0008)

Offset: 0x0328			Register Name: HMIC_CTRL
Bit	Read/Write	Default/Hex	Description
31:23	/	/	/
22:21	R/W	0x0	<p>HMIC_SAMPLE_SELECT</p> <p>Down Sample Setting Select</p> <p>00: Down by 1, 128 Hz</p> <p>01: Down by 2, 64 Hz</p> <p>10: Down by 4, 32 Hz</p> <p>11: Down by 8, 16 Hz</p>
20:16	R/W	0x0	<p>MDATA_THRESHOLD</p> <p>MIC DET EN Threshold Value</p>

Offset: 0x0328			Register Name: HMIC_CTRL
Bit	Read/Write	Default/Hex	Description
15:14	R/W	0x0	<p>HMIC_SF HMIC Smooth Filter setting 00: by pass 01: (x1+x2)/2 10: (x1+x2+x3+x4)/4 11: (x1+x2+x3+x4+ x5+x6+x7+x8)/8</p>
13:10	R/W	0x0	<p>HMIC_M Debounce when the MIC Key down or up. 0000:1 samlpe data 0001:2 samlpe data ... 1111:16 samlpe data</p>
9:6	R/W	0x0	<p>HMIC_N Debounce when earphone plug in or pull out 125 ms to 2 s 0000: 125 ms 0001: 250 ms ... 1111: 2 s</p>
5:3	R/W	0x1	<p>MDATA_THRESHOLD_DEBOUNCE MDATA Threshold Debounce 000: 0 001: 1 010: 2 011: 3 100: 4 101: 5 110: 6 111: 7</p>
2	R/W	0x0	<p>JACK_OUT_IRQ_EN MIC Detect Interrupt Set 0: Disabled 1: Enabled</p>

Offset: 0x0328			Register Name: HMIC_CTRL
Bit	Read/Write	Default/Hex	Description
1	R/W	0x0	<p>JACK_IN_IRQ_EN MIC Detect Interrupt Set 0: Disabled 1: Enabled</p>
0	R/W	0x0	<p>MIC_DET_IRQ_EN MIC Detect Interrupt Set 0: Disabled 1: Enabled</p>

#### 7.4.6.61 0x032C HMIC Status Register (Default Value: 0x0000\_6000)

Offset: 0x032C			Register Name: HMIC_STS
Bit	Read/Write	Default/Hex	Description
31:15	/	/	/
14:13	R/W	0x3	<p>MDATA_DISCARD After MIC DATA data is received, the first N-data will be discarded. N defined as follows: 00: None discarded 01: 1-data discarded 10: 2-data discarded 11: 4-data discarded</p>
12:8	R	0x0	<p>HMIC_DATA HMIC Average Data</p>
7:5	/	/	/
4	R/W1C	0x0	<p>JACK_DET_OIRQ Jack output detect pending interrupt 0: No Pending IRQ 1: Pending IRQ Writing 1 clear pending.</p>
3	R/W1C	0x0	<p>JACK_DET_IIRQ Jack input detect pending interrupt 0: No Pending IRQ 1: Pending IRQ Writing 1 clear pending.</p>
2:1	/	/	/

Offset: 0x032C			Register Name: HMIC_STS
Bit	Read/Write	Default/Hex	Description
0	R/W1C	0x0	<p>MIC_DET_ST MIC detect pending interrupt 0: No pending IRQ 1: Pending IRQ Writing 1 clear pending.</p>

#### 7.4.6.62 0x0340 Headphone2 Analog Control Register (Default Value: 0x0640\_4000)

Offset: 0x0340			Register Name: HP2_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<p>HPFB_BUF_EN Headphone Feedback Buffer OP Enable 0: Disable 1: Enable</p>
30:28	R/W	0x0	<p>HEADPHONE_GAIN HeadPhone Gain 000: 0 dB 001: -6 dB 010: -12 dB 011: -18 dB 100: -24 dB 101: -30 dB 110: -36 dB 111: -42 dB</p>
27:26	R/W	0x1	<p>HPFB_RES Headphone Feedback Big Resistor Control 00: 0.88 MΩ 01: 1.00 MΩ 10: 1.08 MΩ 11: 1.20 MΩ</p>
25:24	R/W	0x2	<p>OPDRV_CUR Headphone OP Output Stage Current Setting 00: Min 11: Max</p>

Offset: 0x0340			Register Name: HP2_REG
Bit	Read/Write	Default/Hex	Description
23:22	R/W	0x1	IOPHP Headphone L/R OP Bias Current Select 00: 6 uA 01: 7 uA 10: 8 uA 11: 9 uA
21	R/W	0x0	HP_DRVEN Headphone Driver Enable 0: Disable 1: Enable
20	R/W	0x0	HP_DRVOUTEN Headphone Driver Output Enable 0: Disabled 1: Enabled
19	R/W	0x0	RSWITCH RSwitch 0: HPOUT OUTPUT VCM of RAMP_DAC 1: VRA1
18	R/W	0x0	RAMPEN Ramp DAC Enable 0: Disabled 1: Enabled
17	R/W	0x0	HPFB_IN_EN Headphone Feedback PAD IN Switch Enable 0: Disabled 1: Enabled
16	R/W	0x0	RAMP_FINAL_CONTROL Headphone Ramp Final Step Control 0: Ramp Output Select Ramp 1: Ramp Output Select HPFB buffer Output
15	R/W	0x0	RAMP_OUT_EN Ramp Output Switch Enable 0: Disable 1: Enable

Offset: 0x0340			Register Name: HP2_REG
Bit	Read/Write	Default/Hex	Description
14:13	R/W	0x2	RAMP_FINAL_STATE_RES Ramp Final State Resistor 00: 2.5k 01: 5.0k 10: 10k 11: 20k
9:8	R/W	0x0	HPFB_BUF_OUTPUT_CURRENT Headphone Feedback Buffer Output Current Select 00: 35I 01: 28I 10: 45I 11: 38I I=7 uA
7:0	/	/	/

#### 7.4.6.63 0x0348 POWER Analog Control Register (Default Value: 0x8000\_3325)

The register is not controlled by the clock and reset of Audio Codec, only controlled by the clock and reset of system bus.

Offset: 0x0348			Register Name: POWER_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x1	ALDO_EN ALDO Enable 0: Disabled 1: Enabled
30	R/W	0x0	HPLDO_EN HPLDO Enable 0: Disabled 1: Enabled
29	R/W	0x0	VAR1SPEEDUP_DOWN_Further_CTRL VAR1 Speedup Down Further Control In Adda Analog 0: The digital logic signal input by the digital-analog interface pin controls the var1_speedup_down function (that is, the var1 signal is rapidly pulled up/down) 1: Writing 1 can finish the var1_speedup_down function (ignore the control of the digital-analog interface pin)

Offset: 0x0348			Register Name: POWER_REG
Bit	Read/Write	Default/Hex	Description
28:17	/	/	/
16	R	0x0	AVCCPOR Avccpor Monitor
15	/	/	/
14:12	R/W	0x3	ALDO_OUTPUT_VOLTAGE ALDO Output Voltage Control 000: 2.03 V 001: 1.95 V 010: 1.87 V 011: 1.80 V 100: 1.73 V 101: 1.67 V 110: 1.61 V 111: 1.56 V
11	/	/	/
10:8	R/W	0x3	HPLDO_OUTPUT_VOLTAGE HPLDO Output Voltage Control 000: 2.03 V 001: 1.95 V 010: 1.87 V 011: 1.80 V 100: 1.73 V 101: 1.67 V 110: 1.61 V 111: 1.56 V
7:0	R/W	0x25	BG_TRIM BG Output Voltage Trimming Only low 6-bit is used. The BG output voltage range is from 0.7 V to 1.208 V.

# Contents

8	Interfaces .....	639
8.1	TWI.....	639
8.1.1	Overview .....	639
8.1.2	Block Diagram .....	639
8.1.3	Functional Description .....	641
8.1.4	Programming Guidelines.....	646
8.1.5	Register List.....	649
8.1.6	Register Description .....	650
8.2	UART.....	665
8.2.1	Overview .....	665
8.2.2	Block Diagram .....	665
8.2.3	Functional Description .....	666
8.2.4	Programming Guidelines.....	672
8.2.5	Register List.....	675
8.2.6	Register Description .....	677
8.3	SPI.....	706
8.3.1	Overview .....	706
8.3.2	Block Diagram .....	706
8.3.3	Functional Description .....	708
8.3.4	Programming Guidelines.....	716
8.3.5	Register List.....	719
8.3.6	Register Description .....	720
8.4	SPI_DBI.....	741
8.4.1	Overview .....	741
8.4.2	Block Diagram .....	742
8.4.3	Functional Description .....	743
8.4.4	Programming Guidelines.....	763
8.4.5	Register List.....	771
8.4.6	Register Description .....	772
8.5	USB2.0 DRD .....	804

8.5.1 Overview .....	804
8.5.2 Block Diagram .....	805
8.5.3 Functional Description .....	805
8.6 USB2.0 HOST .....	807
8.6.1 Overview .....	807
8.6.2 Block Diagram .....	807
8.6.3 Functional Description .....	808
8.6.4 Register List .....	808
8.6.5 EHCI Register Description.....	810
8.6.6 OHCI Register Description.....	827
8.7 GPIO .....	849
8.7.1 Overview .....	849
8.7.2 Block Diagram .....	849
8.7.3 Functional Description .....	850
8.7.4 Register List .....	858
8.7.5 Register Description .....	860
8.8 GPADC.....	934
8.8.1 Overview .....	934
8.8.2 Block Diagram .....	935
8.8.3 Functional Description .....	935
8.8.4 Programming Guidelines.....	937
8.8.5 Register List .....	938
8.8.6 Register Description .....	939
8.9 PWM .....	946
8.9.1 Overview .....	946
8.9.2 Block Diagram .....	946
8.9.3 Functional Description .....	947
8.9.4 Programming Guidelines.....	957
8.9.5 Register List .....	958
8.9.6 Register Description .....	960
8.10 LEDC .....	987
8.10.1 Overview .....	987
8.10.2 Block Diagram .....	987

---

8.10.3 Functional Description .....	988
8.10.4 Programming Guidelines.....	993
8.10.5 Register List.....	998
8.10.6 Register Description.....	998
8.11 EMAC.....	1009
8.11.1 Overview .....	1009
8.11.2 Block Diagram .....	1009
8.11.3 Functional Description .....	1010
8.11.4 Programming Guidelines.....	1017
8.11.5 Register List.....	1018
8.11.6 Register Description .....	1019
8.12 CIR Receiver .....	1037
8.12.1 Overview .....	1037
8.12.2 Block Diagram .....	1037
8.12.3 Functional Description .....	1038
8.12.4 Programming Guidelines.....	1042
8.12.5 Register List.....	1043
8.12.6 Register Description .....	1043
8.13 CIR Transmitter .....	1049
8.13.1 Overview .....	1049
8.13.2 Block Diagram .....	1049
8.13.3 Functional Description .....	1050
8.13.4 Programming Guidelines.....	1052
8.13.5 Register List.....	1053
8.13.6 Register Description .....	1053

# Figures

Figure 8-1 TWI Block Diagram.....	640
Figure 8-2 Write Timing in 7-bit Standard Addressing Mode .....	642
Figure 8-3 Read Timing in 7-bit Standard Addressing Mode .....	642
Figure 8-4 Write Timing in 10-bit Extended Addressing Mode.....	642
Figure 8-5 Read Timing in 10-bit Extended Addressing Mode.....	643
Figure 8-6 TWI Driver Write Packet Transmission .....	643
Figure 8-7 TWI Driver Read Packet Transmission .....	644
Figure 8-8 TWI Programming State Diagram .....	646
Figure 8-9 UART Block Diagram .....	666
Figure 8-10 UART Serial Data Format .....	667
Figure 8-11 Application Diagram for RTS/CTS Autoflow Control.....	668
Figure 8-12 RTS/CTS Autoflow Control Data Format .....	668
Figure 8-13 Application Diagram for IrDA Transceiver .....	668
Figure 8-14 Serial IrDA Data Format .....	669
Figure 8-15 Application Diagram for RS-485 Transceiver .....	669
Figure 8-16 RS-485 Data Format.....	669
Figure 8-17 UART Mode Baud and Error Rates.....	670
Figure 8-18 IrDA Mode Baud and Error Rates.....	671
Figure 8-19 RS485 Mode Baud and Error Rates.....	671
Figure 8-20 SPI Block Diagram .....	707
Figure 8-21 SPI Application Block Diagram .....	709
Figure 8-22 SPI Phase 0 Timing Diagram.....	710
Figure 8-23 SPI Phase 1 Timing Diagram.....	710
Figure 8-24 SPI 3-Wire Mode .....	711
Figure 8-25 SPI Dual-Input/Dual-Output Mode .....	712
Figure 8-26 SPI Dual I/O Mode.....	712
Figure 8-27 SPI Quad-Input/Quad-Output Mode .....	713
Figure 8-28 SPI Write/Read Data in CPU Mode .....	717
Figure 8-29 SPI Write/Read Data in DMA Mode.....	718
Figure 8-30 SPI_DBI Block Diagram.....	742

---

Figure 8-31 SPI Application Block Diagram .....	745
Figure 8-32 DBI Application Block Diagram .....	745
Figure 8-33 SPI Phase 0 Timing Diagram.....	746
Figure 8-34 SPI Phase 1 Timing Diagram.....	746
Figure 8-35 SPI 3-Wire Mode .....	747
Figure 8-36 SPI Dual-Input/Dual-Output Mode .....	748
Figure 8-37 SPI Dual I/O Mode.....	748
Figure 8-38 SPI Quad-Input/Quad-Output Mode .....	749
Figure 8-39 DBI 3-Line Display Bus Serial Interface Writing Operation Format .....	751
Figure 8-40 DBI 3-Line Display Bus Serial Interface 8-bit Reading Operation Format.....	752
Figure 8-41 DBI 3-Line Display Bus Serial Interface 24-bit Reading Operation Format.....	752
Figure 8-42 DBI 3-Line Display Bus Serial Interface 32-bit Reading Operation Format.....	753
Figure 8-43 DBI 4-Line Display Bus Serial Interface Writing Operation Format .....	754
Figure 8-44 DBI 4-Line Display Bus Serial Interface 8-bit Reading Operation Format.....	754
Figure 8-45 DBI 4-Line Display Bus Serial Interface 24-bit Reading Operation Format.....	755
Figure 8-46 DBI 4-Line Display Bus Serial Interface 32-bit Reading Operation Format.....	755
Figure 8-47 RGB111 3-Line Interface Transmit Video Format .....	756
Figure 8-48 RGB444 3-Line Interface Transmit Video Format .....	756
Figure 8-49 RGB565 3-Line Interface Transmit Video Format .....	757
Figure 8-50 RGB666 3-Line Interface Transmit Video Format .....	757
Figure 8-51 RGB111 4-Line Interface Transmit Video Format .....	758
Figure 8-52 RGB444 4-Line Interface Transmit Video Format .....	758
Figure 8-53 RGB565 4-Line Interface Transmit Video Format .....	759
Figure 8-54 RGB666 4-Line Interface Transmit Video Format .....	759
Figure 8-55 RGB444 2 Data Lane Interface Transmit Video Format .....	760
Figure 8-56 RGB565 2 Data Lane Interface Transmit Video Format .....	761
Figure 8-57 RGB666 2 Data Lane Interface Transmit Video Format 0 .....	762
Figure 8-58 RGB666 2 Data Lane Interface Transmit Video Format 1 (ilitek) .....	762
Figure 8-59 RGB666 2 Data Lane Interface Transmit Video Format 2 (New vision) .....	762
Figure 8-60 RGB888 2 Data Lane Interface Transmit Video Format .....	763
Figure 8-61 SPI Write/Read Data in CPU Mode .....	764
Figure 8-62 SPI Write/Read Data in DMA Mode.....	765
Figure 8-63 USB2.0 DRD Controller Block Diagram .....	805

---

---

Figure 8-64 USB2.0 DRD Controller and PHY Connection Diagram .....	806
Figure 8-65 USB2.0 Host Controller Block Diagram .....	807
Figure 8-66 USB2.0 Host Controller and PHY Connection Diagram .....	808
Figure 8-67 GPIO Block Diagram .....	850
Figure 8-68 Pull up/down Logic .....	855
Figure 8-69 IO Buffer Strength Diagram .....	856
Figure 8-70 GPADC Block Diagram .....	935
Figure 8-71 GPADC Clock and Timing Requirement .....	936
Figure 8-72 GPADC Initial Process .....	937
Figure 8-73 PWM Block Diagram .....	947
Figure 8-74 PWM01 Clock Controller Diagram .....	948
Figure 8-75 PWM01 Output Logic Module Diagram .....	949
Figure 8-76 PWM0 High Level Active State .....	950
Figure 8-77 PWM0 Low Level Active State .....	951
Figure 9-78 Phase of PWM0 High Level Active State .....	951
Figure 8-79 PWM0 Output Waveform in Pulse Mode and Cycle Mode .....	952
Figure 8-80 PWM01 Complementary Pair Output .....	953
Figure 8-81 Dead-time Output Waveform .....	954
Figure 8-82 Group 0~3 PWM Signal Output .....	955
Figure 8-83 PWM01 Capture Logic Module Diagram .....	955
Figure 8-84 PWM0 Channel Capture Timing .....	956
Figure 8-85 LEDC Block Diagram .....	987
Figure 8-86 LEDC Package Output Timing Diagram .....	988
Figure 8-87 LEDC 1-frame Output Timing Diagram .....	989
Figure 8-88 LEDC Input Data Structure .....	989
Figure 8-89 LEDC Typical Circuit .....	989
Figure 8-90 LEDC Data Input Code .....	990
Figure 8-91 LEDC Data Transfer Mode .....	990
Figure 8-92 LEDC Normal Configuration Process .....	994
Figure 8-93 LEDC Timeout Abnormal Processing Flow .....	996
Figure 8-94 FIFO Overflow Abnormal Processing Flow .....	997
Figure 8-95 EMAC Block Diagram .....	1010
Figure 8-96 EMAC Typical Application .....	1012

---

---

Figure 8-97 EMAC RX/TX Descriptor List.....	1013
Figure 8-98 CIR Receiver Block Diagram .....	1037
Figure 8-99 CIR Receiver Clock.....	1038
Figure 8-100 CIR Receiver Application Diagram .....	1038
Figure 8-101 NEC Protocol .....	1039
Figure 8-102 Logical ‘0’ and Logical ‘1’ of NEC Protocol .....	1039
Figure 8-103 ATHR Definition .....	1040
Figure 8-104 ITHR Definition.....	1040
Figure 8-105 NTHR Definition .....	1040
Figure 8-106 APAM Definition .....	1041
Figure 8-107 CIR Receiver Process .....	1042
Figure 8-108 CIR_TX Block Diagram .....	1049
Figure 8-109 CIR_TX Clock Description .....	1050
Figure 8-110 Definitions of Logical “1” and Logical “0” .....	1051
Figure 8-111 CIR Message Timing Diagram.....	1051
Figure 8-112 CIR Transmitter Process.....	1052

# Tables

Table 8-1 TWI Sub-blocks.....	640
Table 8-2 TWI External Signals.....	641
Table 8-3 TWI Clock Sources.....	641
Table 8-4 UART External Signals .....	666
Table 8-5 UART Clock Sources.....	667
Table 8-6 SPI Sub-blocks .....	707
Table 8-7 SPI External Signals .....	708
Table 8-8 SPI Clock Sources.....	708
Table 8-9 SPI Transmit Format.....	710
Table 8-10 SPI Old Sample Mode and Run Clock .....	714
Table 8-11 SPI New Sample Mode .....	714
Table 8-12 SPI_DBI Sub-blocks.....	742
Table 8-13 SPI_DBI External Signals.....	743
Table 8-14 SPI_DBI Clock Sources.....	744
Table 8-15 SPI Transmit Format.....	746
Table 8-16 SPI Old Sample Mode and Run Clock .....	750
Table 8-17 SPI New Sample Mode .....	750
Table 8-18 USB2.0 DRD External Signals.....	805
Table 8-19 USB2.0 Host External Signals .....	808
Table 8-20 Multi-function Port .....	850
Table 8-21 PB Multiplex Function .....	852
Table 8-22 PC Multiplex Function .....	852
Table 8-23 PD Multiplex Function.....	852
Table 8-24 PF Multiplex Function .....	853
Table 8-25 PG Multiplex Function.....	853
Table 8-26 Port Function.....	855
Table 8-27 GPADC External Signals.....	935
Table 8-28 GPADC Clock Sources .....	936
Table 8-29 PWM External Signals .....	947
Table 8-30 LEDC Sub-blocks .....	987

---

Table 8-31 LEDC External Signals .....	988
Table 8-32 LEDC Clock Sources .....	988
Table 8-33 Time Parameters of Typical LED Specification .....	990
Table 8-34 EMAC Pin Mapping.....	1010
Table 8-35 EMAC RGMII Pin List .....	1011
Table 8-36 EMAC RMII Pin List .....	1011
Table 8-37 EMAC Clock Characteristics.....	1012
Table 8-38 EMAC_EPHY_CLK_REG0 Configuration Value.....	1017
Table 8-39 CIR Receiver External Signals .....	1038
Table 8-40 CIR_TX External Signals .....	1050

# 8 Interfaces

## 8.1 TWI

### 8.1.1 Overview

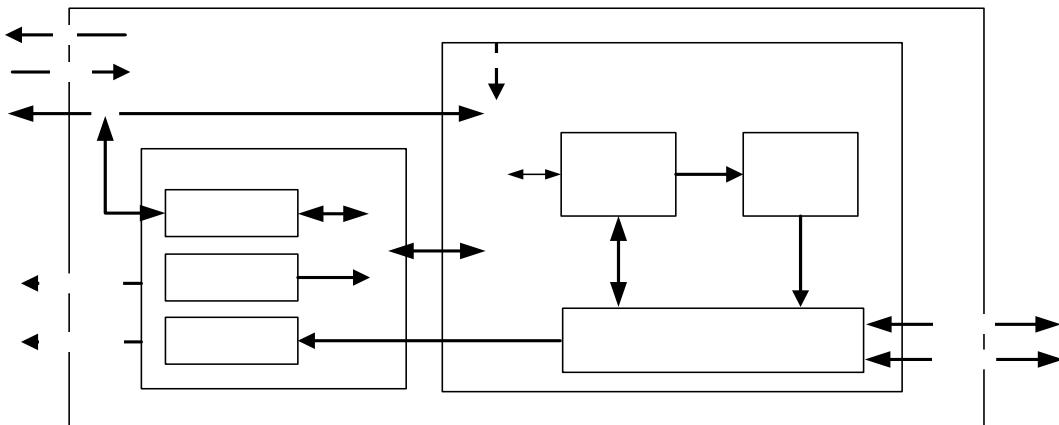
The Two Wire Interface (TWI) provides an interface between a CPU and any TWI-bus-compatible device that connects via the TWI bus. The TWI is designed to be compatible with the standard I2C bus protocol. The communication of the TWI is carried out by a byte-wise mode based on interrupt polled handshaking. Each device on the TWI bus is recognized by a unique address and can operate as either transmitter or receiver, a device connected to the TWI bus can be considered as master or slave when performing data transfers. Note that a master device is a device that initiates a data transfer on the bus and generates the clock signals to permit the transfer. During this transfer, any device addressed by this master is considered a slave.

The TWI has the following features:

- Compliant with I2C bus standard
- Supports standard mode (up to 100 kbit/s) and fast mode (up to 400 kbit/s)
- Supports 7-bit and 10-bit device addressing modes
- Supports master mode or slave mode
- Master mode features:
  - Supports the bus arbitration in the case of multiple master devices
  - Supports clock synchronization and bit and byte waiting
  - Supports packet transmission and DMA
- Slave mode features:
  - Interrupt on address detection
- The TWI controller includes one TWI engine and one TWI driver. And the TWI driver supports packet transmission and DMA mode when TWI works in master mode

### 8.1.2 Block Diagram

Figure 8-1 shows the block diagram of TWI.

**Figure 8-1 TWI Block Diagram**

TWI contains the following sub-blocks:

**Table 8-1 TWI Sub-blocks**

Sub-block	Description
RESET	Module reset signal
INT	Module output interrupt signal
CFG_REG	Module configuration register in TWI
PE	Packet encoding/decoding
CCU	Module clock controller unit
SEND_FIFO	The register address bytes and the written data bytes are buffered in SEND_FIFO
RECV_FIFO	The read data bytes are buffered in RECV_FIFO

The controller includes TWI engine and TWI driver. Each time the TWI engine sends a START signal, a STOP signal, or a BYTE data, or a corresponding ACK, the TWI engine will generate an interrupt, and wait for the CPU to process and clear the interrupt before the next START, STOP, or BYTE, ACK transmission can be performed. Therefore, when a device communication is completed, many interrupts will be generated, and the CPU needs to wait for the previous interrupt before it can configure the next one. The TWI driver defines each communication with the device as a packet transmission. The CPU can directly configure the slave address, register address and data transmission for one or more package transmissions without waiting for interruption, then start the TWI driver, and the TWI driver can control the TWI engine to complete a pre-configured communication, and report an interrupt to the CPU after completion.

### 8.1.3 Functional Description

#### 8.1.3.1 External Signals

The TWI controller has 4 TWI modules called TWI0, TWI1, TWI2, and TWI3. The following table describes the external signals of the TWI. The TWIn-SCK and TWIn-SDA are bidirectional I/O, when the TWI is configured as a master device, the TWIn-SCK is an output pin; when the TWI is configurable as a slave device, the TWIn-SCK is an input pin. When using TWI, the corresponding PADs are selected as TWI function via section 8.7 “[GPIO](#)”.

**Table 8-2 TWI External Signals**

Signal	Description	Type
TWI0-SCK	TWI0 Clock Signal	I/O, OD
TWI0-SDA	TWI0 Serial Data	I/O, OD
TWI1-SCK	TWI1 Clock Signal	I/O, OD
TWI1-SDA	TWI1 Serial Data	I/O, OD
TWI2-SCK	TWI2 Clock Signal	I/O, OD
TWI2-SDA	TWI2 Serial Data	I/O, OD
TWI3-SCK	TWI3 Clock Signal	I/O, OD
TWI3-SDA	TWI3 Serial Data	I/O, OD

#### 8.1.3.2 Clock Sources

Each TWI controller has an input clock source. The following table describes the clock sources for TWI. After selecting a proper clock, users must open the gating of TWI and release the corresponding reset bit.

For more details on the clock setting, configuration, and gating information, see section 3.3 “[CCU](#)”.

**Table 8-3 TWI Clock Sources**

Clock Sources	Description
APB1 Bus	TWI clock source. Refer to CCU for details on APB1.

#### 8.1.3.3 Write/Read Timing in Standard and Extended Addressing Mode

This section is the 7-bit/10-bit addressing mode of the entire TWI protocol to read and write device registers. It can be achieved by directly using the TWI engine or using the TWI driver to control the TWI engine.

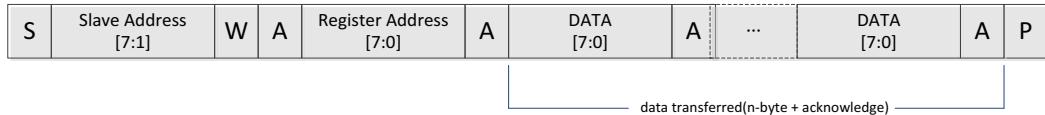
Figure 8-2 describes the write timing in 7-bit standard addressing mode.

**Figure 8-2 Write Timing in 7-bit Standard Addressing Mode**

Slave addr = 7-bit , register addr = 8-bit, data = 8-bit



Slave addr = 7-bit , register addr = 8-bit, data = n-byte



from master to slave

S: START condition

A: acknowledge(SDA LOW)



from slave to master

P: STOP condition

Ā: not acknowledge(SDA HIGH)

Figure 8-3 describes the read timing in 7-bit standard address mode.

**Figure 8-3 Read Timing in 7-bit Standard Addressing Mode**

Slave addr = 7-bit , register addr = 8-bit, data = 8-bit



Slave addr = 7-bit , register addr = 8-bit, data = n-byte



from master to slave

S: START condition

A: acknowledge(SDA LOW)

Sr: RE-START condition



from slave to master

P: STOP condition

Ā: not acknowledge(SDA HIGH)

Figure 8-4 describes the write timing in 10-bit extended address mode.

**Figure 8-4 Write Timing in 10-bit Extended Addressing Mode**

Slave addr = 10-bit , register addr = 8-bit, data = 8-bit



Slave addr = 10-bit , register addr = 8-bit, data = n-byte



from master to slave

S: START condition

A: acknowledge(SDA LOW)



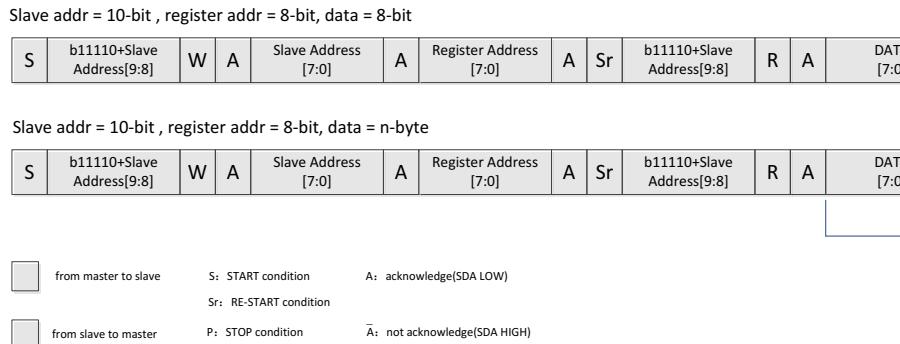
from slave to master

P: STOP condition

Ā: not acknowledge(SDA HIGH)

Figure 8-5 describes the read timing in 10-bit extended address mode.

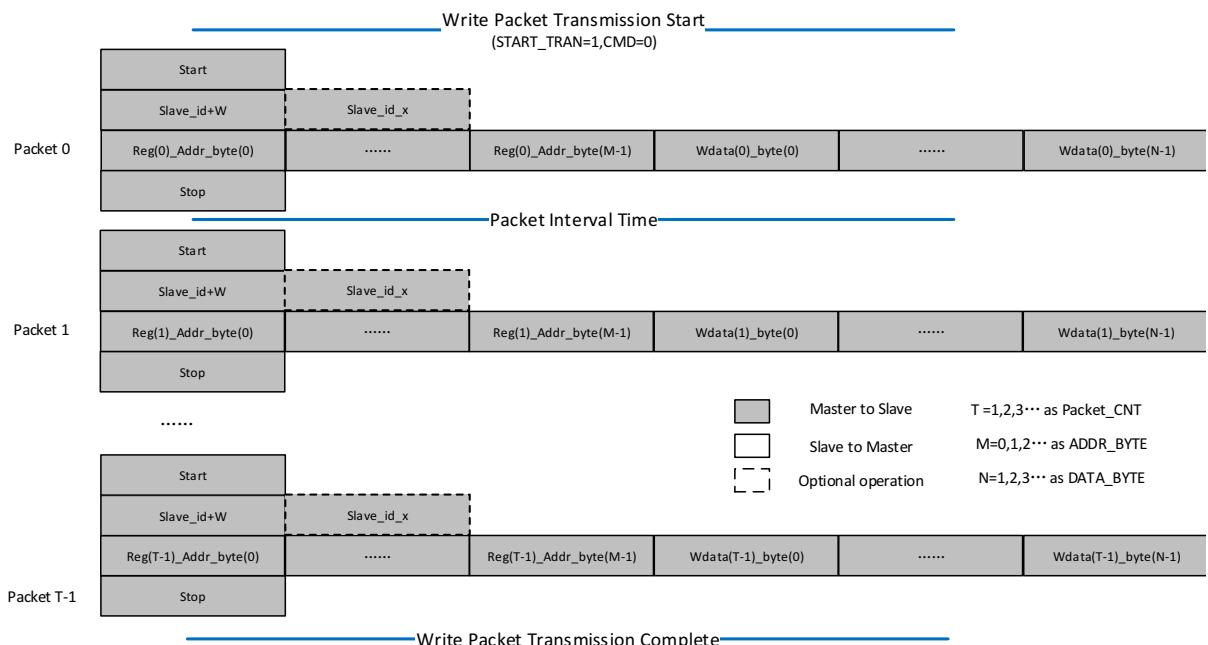
**Figure 8-5 Read Timing in 10-bit Extended Addressing Mode**

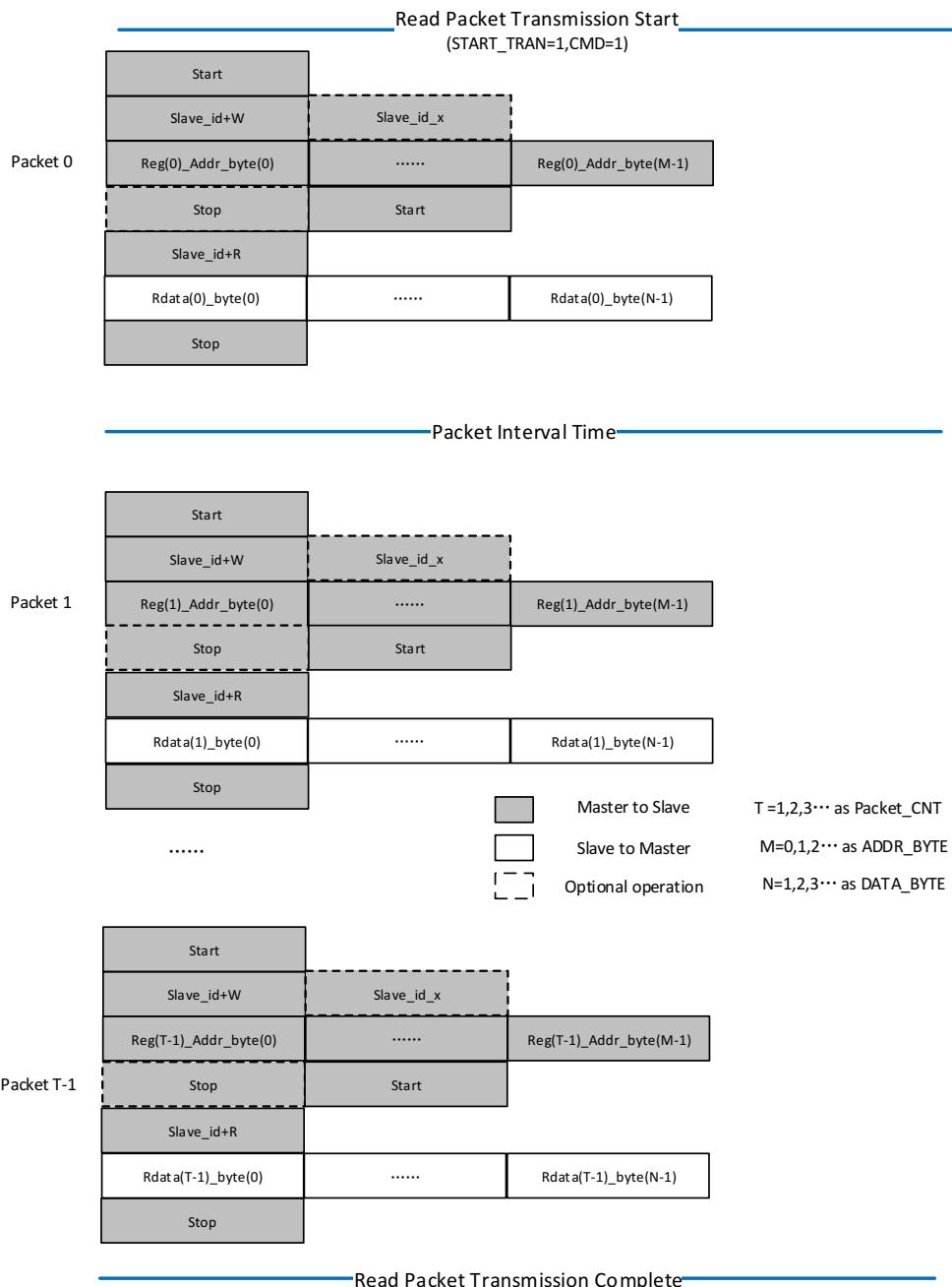


#### 8.1.3.4 Write/Read Packet Transmission of TWI Driver

The TWI driver is only supported for master mode. When the TWI works in master mode, the TWI driver drives the TWI engine for one or more packet transmission instead of the CPU host. Packet transmission is defined in the following figures. The register address bytes and the written data bytes are buffered in SEND\_FIFO, the read data bytes are buffered in RECV\_FIFO.

**Figure 8-6 TWI Driver Write Packet Transmission**



**Figure 8-7 TWI Driver Read Packet Transmission**

### 8.1.3.5 Master and Slave Mode of TWI Engine

In Master mode, the CPU host controls the TWI engine by writing command and data to its registers. The TWI engine transmits an interrupt to CPU when each time a byte transfer is done or a START/STOP command is detected. The CPU host can poll the status register if the interrupt mechanism is not disabled by the CPU host.

When the CPU host wants to start a bus transfer, it initiates a bus START to enter the master mode by setting [TWI\\_CNT\[INT\\_FLAG\]](#) to high. The TWI engine will assert the INT line and [TWI\\_CNT\[INT\\_FLAG\]](#) to indicate a completion for the START command and each consequent byte transfer. At each interrupt, the CPU host needs

to check the current state by the [TWI\\_STAT](#) register. A transfer must conclude with the STOP command by setting [TWI\\_CNTR\[M\\_STP\]](#) to high.

In Slave mode, the TWI engine also constantly samples the bus and look for its own slave address during addressing cycles. Once a match is found, it is addressed, and the TWI engine interrupts the CPU host with the corresponding status. Upon request, the CPU host should read the status, read/write the [TWI\\_DATA](#) register, and set the [TWI\\_CNTR](#) register. After each byte transfer, a slave device always stops the operation of the remote master by holding the next low pulse on the SCL line until the CPU host responds to the status of the previous byte transfer or START command.

#### 8.1.3.6 Generation of Repeated Start

After the data transfer, if the master still requires the bus, it can signal another Start followed by another slave address without signaling a Stop.

#### 8.1.3.7 Programming State Diagram

Figure 8-8 shows the TWI programming state diagram. For the value between two states, see the [TWI\\_STAT](#) register in section 8.1.6.5.

M\_SEND\_S: master sends START signal;

M\_SEND\_ADDR: master sends slave address;

M\_SEND\_XADD: master sends slave extended address;

M\_SEND\_SR: master repeated start;

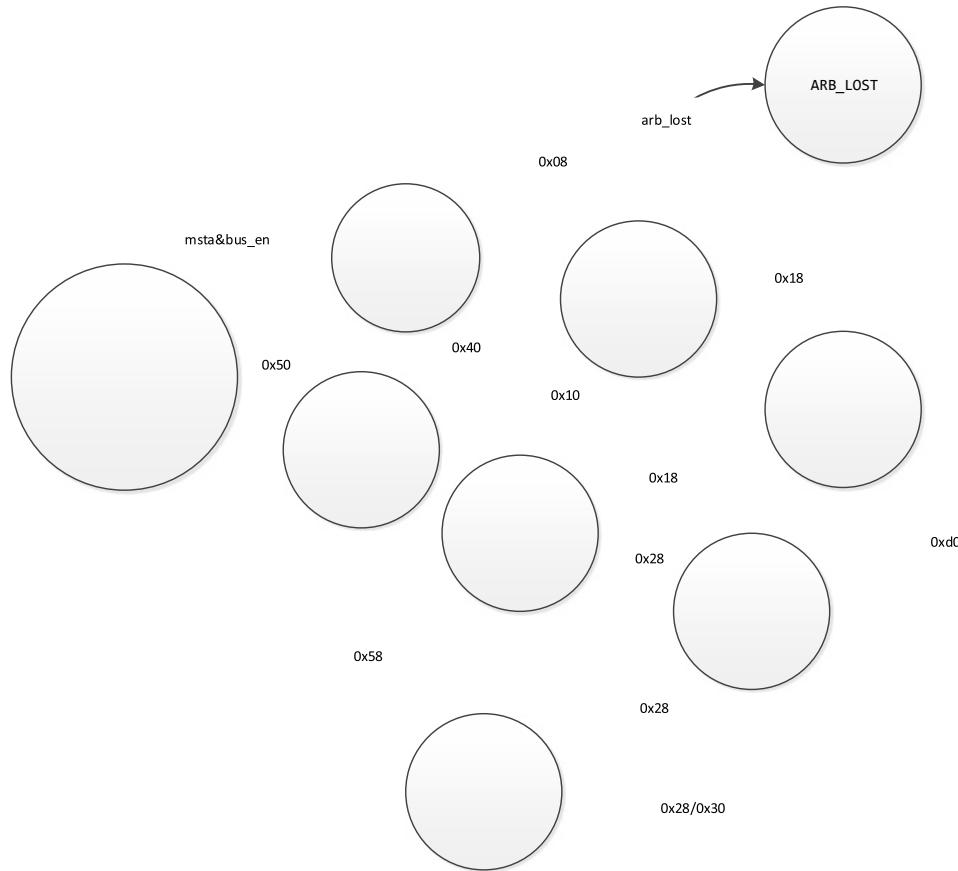
M\_SEND\_DATA: master sends data;

M\_SEND\_P: master sends STOP signal;

M\_RECV\_DATA: master receives data;

ARB\_LOST: Arbitration lost;

C\_IDLE: Idle.

**Figure 8-8 TWI Programming State Diagram**

### 8.1.4 Programming Guidelines

The TWI controller operates in an 8-bit data format. The data on the TWI\_SDA line is always 8 bits long. At first, the TWI controller sends a start condition. When in the addressing formats of 7-bit, the TWI sends out an 8-bit message which includes 7 MSB slave address and 1 LSB read/write flag. The least significant of the slave address indicates the direction of transmission. When the TWI works in 10-bit slave address mode, the operation will be divided into two steps, for details on the operation, refer to register description in Section 8.1.6.1 and 8.1.6.2.

The following takes the TWI module in the CPUX domain as an example.

#### 8.1.4.1 Initialization for TWI Engine

To initialize the TWI engine, perform the following steps:

- Step 1** Configure corresponding GPIO multiplex function as TWI mode.
- Step 2** For TWIn, set [TWI\\_BGR\\_REG\[TWIn\\_GATING\]](#) in CCU module to 0 to close TWIn clock.
- Step 3** For TWIn, set [TWI\\_BGR\\_REG\[TWIn\\_RST\]](#) in CCU module to 0, then set to 1 to reset TWIn.

- Step 4** For TWIn, set [TWI\\_BGR\\_REG](#)[TWIn\_GATING] in CCU module to 1 to open TWIn clock.
- Step 5** Configure [TWI\\_CCR](#)[CLK\_M] and [TWI\\_CCR](#)[CLK\_N] to get the needed rate (The clock source of TWI is from APB1).
- Step 6** Configure [TWI\\_CNTR](#)[BUS\_EN] and [TWI\\_CNTR](#)[A\_ACK], when using interrupt mode, set [TWI\\_CNTR](#)[INT\_EN] to 1, and register the system interrupt through GIC module. In slave mode, configure [TWI\\_ADDR](#) and [TWI\\_XADDR](#) registers to finish TWI initialization configuration.

#### 8.1.4.2 Writing Data Operation for TWI Engine

To write data to the device, perform the following steps:

- Step 1** Clear [TWI\\_EFR](#) register, and configure [TWI\\_CNTR](#)[M\_STA] to 1 to transmit the START signal.
- Step 2** After the START signal is transmitted, the first interrupt is triggered, then write device ID to [TWI\\_DATA](#) (For a 10-bit device ID, firstly write the first byte ID, secondly write the second byte ID in the next interrupt).
- Step 3** The Interrupt is triggered again after device ID transmission completes, write device data address to be read to [TWI\\_DATA](#) (For a 16-bit address, firstly write the first-byte address, secondly write the second-byte address).
- Step 4** Interrupt is triggered after data address transmission completes, write data to be transmitted to [TWI\\_DATA](#) (For consecutive write data operation, every byte transmission completion triggers interrupt, during interrupt write the next byte data to [TWI\\_DATA](#)).
- Step 5** After transmission completes, write [TWI\\_CNTR](#)[M\_STP] to 1 to transmit the STOP signal and end this write-operation.

#### 8.1.4.3 Reading Data Operation for TWI Engine

To read data from the device, perform the following steps:

- Step 1** Clear [TWI\\_EFR](#) register, and set [TWI\\_CNTR](#)[A\_ACK] to 1, and configure [TWI\\_CNTR](#)[M\_STA] to 1 to transmit the START signal.
- Step 2** After the START signal is transmitted, the first interrupt is triggered, then write device ID to [TWI\\_DATA](#) (For a 10-bit device ID, firstly write the first-byte ID, secondly write the second-byte ID in the next interrupt).
- Step 3** The Interrupt is triggered again after device ID transmission completes, write device data address to be read to [TWI\\_DATA](#) (For a 16-bit address, firstly write the first-byte address, secondly write the second-byte address).

- Step 4** The Interrupt is triggered after data address transmission completes, write [TWI\\_CNTR\[M\\_STA\]](#) to 1 to transmit new START signal, and after interrupt triggers, write device ID to [TWI\\_DATA](#) to start read-operation.
- Step 5** After device address transmission completes, each receive completion will trigger an interrupt, in turn, read [TWI\\_DATA](#) to get data, when receiving the previous interrupt of the last byte data, clear [C:\Users\chenxiaofang\AppData\Roaming\Microsoft\Word\ Hlk50046436 - Hlk50051279\[A\\_ACK\]](#) to stop acknowledge signal of the last byte.
- Step 6** Write [TWI\\_CNTR\[M\\_STP\]](#) to 1 to transmit the STOP signal and end this read-operation.

#### 8.1.4.4 Initialization for TWI Driver

To initialize the TWI driver, perform the following steps:

- Step 1** Configure corresponding GPIO multiplex function as TWI mode.
- Step 2** For TWIn, set [TWI\\_BGR\\_REG\[TWIN\\_GATING\]](#) in CCU module to 0 to close TWIn clock.
- Step 3** For TWIn, set [TWI\\_BGR\\_REG\[TWIN\\_RST\]](#) in CCU module to 0, then set to 1 to reset TWIn.
- Step 4** For TWIn, set [TWI\\_BGR\\_REG\[TWIN\\_GATING\]](#) in CCU module to 1 to open TWIn clock.
- Step 5** Set [TWI\\_DRV\\_CTRL\[TWI\\_DRV\\_EN\]](#) to 1 to enable the TWI driver.
- Step 6** Configure [TWI\\_DRV\\_BUS\\_CTRL\[CLK\\_M\]](#) and [TWI\\_DRV\\_BUS\\_CTRL\[CLK\\_N\]](#) to get the needed rate (The clock source of TWI is from APB1).
- Step 7** Set [TWI\\_DRV\\_CTRL\[RESTART\\_MODE\]](#) to 0 and [C:\Users\chenxiaofang\AppData\Roaming\Microsoft\Word\ Hlk50051373 - Hlk50051463\[READ\\_TRAN\\_MODE\]](#) to 1, set [TWI\\_DRV\\_INT\\_CTRL\[TRAN\\_COM\\_INT\\_EN\]](#) to 1.
- Step 8** When using DMA for data transmission, set [TWI\\_DRV\\_DMA\\_CFG\[DMA\\_RX\\_EN\]](#) and [TWI\\_DRV\\_DMA\\_CFG\[DMA\\_TX\\_EN\]](#) to 1, and configure [TWI\\_DRV\\_DMA\\_CFG\[RX\\_TRIG\]](#) and [TWI\\_DRV\\_DMA\\_CFG\[TX\\_TRIG\]](#) to set the thresholds of RXFIFO and TXFIFO.

#### 8.1.4.5 Writing Packet Transmission for TWI Driver

To write package to the device, perform the following steps:

- Step 1** Configure [TWI\\_DRV\\_SLV\[SLV\\_ID\]](#) to set the device ID, and configure [TWI\\_DRV\\_SLV\[CMD\]](#) to 0 to set the write operation.
- Step 2** Configure [TWI\\_DRV\\_FMT\[ADDR\\_BYTE\]](#) according to the address width of the device register, and [TWI\\_DRV\\_FMT\[DATA\\_BYTE\]](#) according to the written data count in a packet.

**Step 3** Configure [TWI\\_DRV\\_CFG](#)[PACKET\_CNT] to set the written packet number.

**Step 4** Configure DMA channel, including TWI TXFIFO, device register address, and the written data.

**Step 5** Set [C:\Users\chenxiaofang\AppData\Roaming\Microsoft\Word\ Hlk50051463 - Hlk50051932](#)[START\_TRAN] to 1 to start TWI Driver transmission.

**Step 6** When TWI driver transmission completes, the interrupt is triggered, it indicates that the write packet transmission ends.

#### 8.1.4.6 Reading Packet Transmission for TWI Driver

To read package from the device, perform the following steps:

**Step 1** Configure [TWI\\_DRV\\_SLV](#)[SLV\_ID] to set the device ID, and configure [TWI\\_DRV\\_SLV](#)[CMD] to 1 to set the read operation.

**Step 2** Configure [TWI\\_DRV\\_FMT](#)[ADDR\_BYTE] according to the address width of the device register, and [TWI\\_DRV\\_FMT](#)[DATA\_BYTE] according to the read data count in a packet.

**Step 3** Configure [TWI\\_DRV\\_CFG](#)[PACKET\_CNT] to set the read packet number.

**Step 4** Configure DMA channel, including TWI TXFIFO, TWI RXFIFO, device register address and the read data.

**Step 5** Set [C:\Users\chenxiaofang\AppData\Roaming\Microsoft\Word\ Hlk50051463 - Hlk50051932](#)[START\_TRAN] to 1 to start TWI Driver transmission.

**Step 6** When TWI driver transmission completes, the interrupt is triggered, it indicates that the read packet transmission ends.

#### 8.1.5 Register List

Module Name	Base Address
TWI0	0x02502000
TWI1	0x02502400
TWI2	0x02502800
TWI3	0x02502C00

Register Name	Offset	Description
TWI_ADDR	0x0000	TWI Slave Address Register
TWI_XADDR	0x0004	TWI Extended Slave Address Register
TWI_DATA	0x0008	TWI Data Byte Register

Register Name	Offset	Description
TWI_CNTR	0x000C	TWI Control Register
TWI_STAT	0x0010	TWI Status Register
TWI_CCR	0x0014	TWI Clock Control Register
TWI_SRST	0x0018	TWI Software Reset Register
TWI_EFR	0x001C	TWI Enhance Feature Register
TWI_LCR	0x0020	TWI Line Control Register
TWI_DRV_CTRL	0x0200	TWI_DRV Control Register
TWI_DRV_CFG	0x0204	TWI_DRV Transmission Configuration Register
TWI_DRV_SLV	0x0208	TWI_DRV Slave ID Register
TWI_DRV_FMT	0x020C	TWI_DRV Packet Format Register
TWI_DRV_BUS_CTRL	0x0210	TWI_DRV Bus Control Register
TWI_DRV_INT_CTRL	0x0214	TWI_DRV Interrupt Control Register
TWI_DRV_DMA_CFG	0x0218	TWI_DRV DMA Configure Register
TWI_DRV_FIFO_CON	0x021C	TWI_DRV FIFO Content Register
TWI_DRV_SEND_FIFO_ACC	0x0300	TWI_DRV Send Data FIFO Access Register
TWI_DRV_RECV_FIFO_ACC	0x0304	TWI_DRV Receive Data FIFO Access Register

### 8.1.6 Register Description

#### 8.1.6.1 0x0000 TWI Slave Address Register (Default Value: 0x0000\_0000)

Offset: 0x0000			Register Name: TWI_ADDR
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:1	R/W	0x0	SLA Slave Address For 7-bit addressing, the bit[7:1] indicates: SLA6, SLA5, SLA4, SLA3, SLA2, SLA1, SLA0 For 10-bit addressing, the bit[7:1] indicates: 1, 1, 1, 1, 0, SLAX[9:8]
0	R/W	0x0	GCE General Call Address Enable 0: Disable 1: Enable



For 7-bit addressing:

SLA6 – SLA0 is the 7-bit address of the TWI in slave mode. When the TWI receives this address after a START condition, it will generate an interrupt and enter slave mode. (SLA6 corresponds to the first bit received from the TWI bus.) If GCE is set to '1', the TWI will also recognize the general call address (00h).

For 10-bit addressing:

When the address received starts with b'11110, the TWI recognizes b'11110 as the first part of a 10-bit address and if the next two bits match ADDR[2:1] (For example, SLAX9 and SLAX8 for the extended address of the device), it sends an ACK. (The device does not generate an interrupt at this point.) If the next byte of the address matches the XADDR register (SLAX7 – SLAX0), the TWI generates an interrupt and goes into slave mode.

#### 8.1.6.2 0x0004 TWI Extend Address Register (Default Value: 0x0000\_0000)

Offset: 0x0004			Register Name: TWI_XADDR
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R/W	0x0	SLAX Extend Slave Address SLAX[7:0]

#### 8.1.6.3 0x0008 TWI Data Register (Default Value: 0x0000\_0000)

Offset: 0x0008			Register Name: TWI_DATA
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R/W	0x0	TWI_DATA Data byte transmitted or received

#### 8.1.6.4 0x000C TWI Control Register (Default Value: 0x0000\_0000)

Offset: 0x000C			Register Name: TWI_CNTR
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R/W	0x0	<p>INT_EN Interrupt Enable 0: The interrupt line always low 1: The interrupt line will go high when INT_FLAG is set.</p>
6	R/W	0x0	<p>BUS_EN TWI Bus Enable 0: The TWI bus SDA/SCL is ignored and the TWI controller will not respond to any address on the bus. 1: The TWI will respond to call to its slave address – and to the general call address if the GCE bit in the ADDR register is set.</p> <p><b>Note: In master operation mode, this bit should be set to '1'.</b></p>
5	R/WAC	0x0	<p>M_STA Master Mode Start When the M_STA is set to '1', the TWI controller enters master mode and will transmit a START condition on the bus when the bus is free. If the M_STA bit is set to '1' when the TWI controller is already in master mode and one or more bytes have been transmitted, then a repeated START condition will be sent. If the M_STA bit is set to '1' when the TWI is accessed in slave mode, the TWI will complete the data transfer in slave mode then enter master mode when the bus has been released. The M_STA bit is cleared automatically after a START condition has been sent. Writing a '0' to this bit has no effect.</p>
4	R/W1C	0x0	<p>M_STP Master Mode Stop If the M_STP is set to '1' in master mode, a STOP condition is transmitted on the TWI bus. If the M_STP bit is set to '1' in slave mode, the TWI will indicate if a STOP condition has been received, but no STOP condition will be transmitted on the TWI bus. If both M_STA and M_STP bits are set, the TWI will first transmit the STOP condition (if in master mode), then transmit the START condition. The M_STP bit is cleared automatically. Writing a '0' to this bit has no effect.</p>

Offset: 0x000C			Register Name: TWI_CNTR
Bit	Read/Write	Default/Hex	Description
3	R/W1C	0x0	<p>INT_FLAG Interrupt Flag</p> <p>The INT_FLAG is automatically set to '1' when any of the 28 (out of the possible 29) states is entered (see 'STAT Register' below). The state that does not set INT_FLAG is state F8h. If the INT_EN bit is set, the interrupt line goes high when INT_FLAG is set to '1'. If the TWI is operating in slave mode, the data transfer is suspended when INT_FLAG is set and the low period of the TWI bus clock line (SCL) is stretched until '1' is written to INT_FLAG. The TWI clock line is then released and the interrupt line goes low.</p>
2	R/W	0x0	<p>A_ACK Assert Acknowledge</p> <p>When A_ACK is set to '1', an Acknowledge (low level on SDA) will be sent during the acknowledge clock pulse on the TWI bus if:</p> <ul style="list-style-type: none"> <li>(1). Either the whole of a matching 7-bit slave address or the first or the second byte of a matching 10-bit slave address has been received.</li> <li>(2). The general call address has been received and the GCE bit in the ADDR register is set to '1'.</li> <li>(3). A data byte has been received in master or slave mode.</li> </ul> <p>When A_ACK is '0', a Not Acknowledge (high level on SDA) will be sent when a data byte is received in master or slave mode.</p> <p>If A_ACK is cleared to '0' in slave transmitter mode, the byte in the DATA register is assumed to be the 'last byte'. After this byte has been transmitted, the TWI will enter state C8h then return to the idle state (status code F8h) when INT_FLAG is cleared.</p> <p>The TWI will not respond as a slave unless A_ACK is set.</p>
1	/	/	/
0	R/W	0x0	<p>CLK_COUNT_MODE</p> <p>0: scl clock high period count on oscl 1: scl clock high period count on iscl</p>

#### 8.1.6.5 0x0010 TWI Status Register (Default Value: 0x0000\_00F8)

Offset: 0x0010			Register Name: TWI_STAT
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/

Offset: 0x0010			Register Name: TWI_STAT
Bit	Read/Write	Default/Hex	Description
7:0	R	0xF8	<p>STA</p> <p>Status Information Byte</p> <p>Code Status</p> <p>0x00: Bus error</p> <p>0x08: START condition transmitted</p> <p>0x10: Repeated START condition transmitted</p> <p>0x18: Address + Write bit transmitted, ACK received</p> <p>0x20: Address + Write bit transmitted, ACK not received</p> <p>0x28: Data byte transmitted in master mode, ACK received</p> <p>0x30: Data byte transmitted in master mode, ACK not received</p> <p>0x38: Arbitration lost in address or data byte</p> <p>0x40: Address + Read bit transmitted, ACK received</p> <p>0x48: Address + Read bit transmitted, ACK not received</p> <p>0x50: Data byte received in master mode, ACK transmitted</p> <p>0x58: Data byte received in master mode, not ACK transmitted</p> <p>0x60: Slave address + Write bit received, ACK transmitted</p> <p>0x68: Arbitration lost in the address as master, slave address + Write bit received, ACK transmitted</p> <p>0x70: General Call address received, ACK transmitted</p> <p>0x78: Arbitration lost in the address as master, General Call address received, ACK transmitted</p> <p>0x80: Data byte received after slave address received, ACK transmitted</p> <p>0x88: Data byte received after slave address received, not ACK transmitted</p> <p>0x90: Data byte received after General Call received, ACK transmitted</p> <p>0x98: Data byte received after General Call received, not ACK transmitted</p> <p>0xA0: STOP or repeated START condition received in slave mode</p> <p>0xA8: Slave address + Read bit received, ACK transmitted</p> <p>0xB0: Arbitration lost in the address as master, slave address + Read bit received, ACK transmitted</p> <p>0xB8: Data byte transmitted in slave mode, ACK received</p> <p>0xC0: Data byte transmitted in slave mode, ACK not received</p> <p>0xC8: The Last byte transmitted in slave mode, ACK received</p> <p>0xD0: Second Address byte + Write bit transmitted, ACK received</p>

Offset: 0x0010			Register Name: TWI_STAT
Bit	Read/Write	Default/Hex	Description
			0xD8: Second Address byte + Write bit transmitted, ACK not received 0xF8: No relevant status information, INT_FLAG=0 Others: Reserved

#### 8.1.6.6 0x0014 TWI Clock Register (Default Value: 0x0000\_0080)

Offset: 0x0014			Register Name: TWI_CCR
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R/W	0x1	CLK_DUTY Setting duty cycle of clock as master 0: 50% 1: 40%
6:3	R/W	0x0	CLK_M The TWI SCL output frequency, in master mode, is F1/10: $F1 = F0/(CLK\_M + 1)$ $F_{SCL} = F1/10 = F_{in}/(2^{CLK\_N} * (CLK\_M + 1)*10)$ Specially, $F_{SCL} = F1/11$ when $CLK\_M=0$ and $CLK\_DUTY=40\%$ due to the delay of SCL sample debounce.
2:0	R/W	0x0	CLK_N The TWI bus is sampled by the TWI at the frequency defined by F0: $F_{SAMP} = F0 = F_{in}/2^{CLK\_N}$ The TWI SCL output frequency, in master mode, is F1/10: $F1 = F0/(CLK\_M + 1)$ $F_{SCL} = F1/10 = F_{in}/(2^{CLK\_N} * (CLK\_M + 1)*10)$ Specially, $F_{SCL} = F1/11$ when $CLK\_M=0$ and $CLK\_DUTY=40\%$ due to the delay of SCL sample debounce. For Example: $F_{in} = 24 \text{ MHz (APB clock input)}$ For 400 kHz full speed 2-wire, $CLK\_N = 1$ , $CLK\_M = 2$ $F0 = 24 \text{ MHz}/2^1 = 12 \text{ MHz}$ , $F1 = F0/(10*(2+1)) = 0.4 \text{ MHz}$ For 100 kHz standard speed 2-wire, $CLK\_N = 1$ , $CLK\_M = 11$ $F0 = 24 \text{ MHz}/2^1 = 12 \text{ MHz}$ , $F1 = F0/(10*(11+1)) = 0.1 \text{ MHz}$

#### 8.1.6.7 0x0018 TWI Soft Reset Register (Default Value: 0x0000\_0000)

Offset: 0x0018			Register Name: TWI_SRST
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/WAC	0x0	SOFT_RST Soft Reset Write '1' to this bit to reset the TWI and clear to '0' when completing Soft Reset operation.

#### 8.1.6.8 0x001C TWI Enhance Feature Register (Default Value: 0x0000\_0000)

Offset: 0x001C			Register Name: TWI_EFR
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1:0	R/W	0x0	DBN Data Byte Number Follow Read Command Control 00: No data byte can be written after the read command 01: Only 1-byte data can be written after the read command 10: 2-bytes data can be written after the read command 11: 3-bytes data can be written after the read command

#### 8.1.6.9 0x0020 TWI Line Control Register (Default Value: 0x0000\_003A)

Offset: 0x0020			Register Name: TWI_LCR
Bit	Read/Write	Default/Hex	Description
31:6	/	/	/
5	R	0x1	SCL_STATE Current State of TWI_SCL 0: Low 1: High
4	R	0x1	SDA_STATE Current State of TWI_SDA 0: Low 1: High

Offset: 0x0020			Register Name: TWI_LCR
Bit	Read/Write	Default/Hex	Description
3	R/W	0x1	<p>SCL_CTL TWI_SCL Line State Control Bit When the line control mode is enabled (bit[2] is set), this bit decides the output level of TWI_SCL. 0: Output low level 1: Output high level</p>
2	R/W	0x0	<p>SCL_CTL_EN TWI_SCL Line State Control Enable When this bit is set, the state of TWI_SCL is controlled by the value of bit[3]. 0: Disable TWI_SCL line control mode 1: Enable TWI_SCL line control mode</p>
1	R/W	0x1	<p>SDA_CTL TWI_SDA Line State Control Bit When the line control mode is enabled (bit[0] is set), this bit decides the output level of TWI_SDA. 0: Output low level 1: Output high level</p>
0	R/W	0x0	<p>SDA_CTL_EN TWI_SDA Line State Control Enable When this bit is set, the state of TWI_SDA is controlled by the value of bit[1]. 0: Disable TWI_SDA line control mode 1: Enable TWI_SDA line control mode</p>

**8.1.6.10 0x0200 TWI\_DRV Control Register (Default Value: 0x00F8\_1000)**

Offset: 0x0200			Register Name: TWI_DRV_CTRL
Bit	Read/Write	Default/Hex	Description
31	R/WAC	0x0	<p>START_TRAN Start transmission 0: Transmission idle 1: Start transmission Automatically cleared to '0' when finished. If the slave is not responding for the expected status over the time defined by TIMEOUT, the current transmission will stop. All setting formats and data will be loaded from registers and FIFO when the transmission starts.</p>
30	/	/	/
29	R/W	0x0	<p>RESTART_MODE Restart mode 0: RESTART 1: STOP+START Define the TWI_DRV action after sending the register address.</p>
28	R/W	0x0	<p>READ_TRAN_MODE Read transition mode 0: Send slave_id+W 1: Not send slave_id+W Setting this bit to 1 if reading from a slave in which the register width is equal to 0.</p>
27:24	R	0x0	<p>TRAN_RESULT Transition result 000: OK 001: FAIL Other: Reserved</p>

Offset: 0x0200			Register Name: TWI_DRV_CTRL
Bit	Read/Write	Default/Hex	Description
23:16	R	0xf8	<p>TWI_STA TWI status 0x00: bus error 0x08: START condition transmitted 0x10: Repeated START condition transmitted 0x18: Address + Write bit transmitted, ACK received 0x20: Address + Write bit transmitted, ACK not received 0x28: Data byte transmitted in master mode, ACK received 0x30: Data byte transmitted in master mode, ACK not received 0x38: Arbitration lost in address or data byte 0x40: Address + Read bit transmitted, ACK received 0x48: Address + Read bit transmitted, ACK not received 0x50: Data byte received in master mode, ACK received 0x58: Data byte received in master mode, ACK not received 0x01: Timeout when sending the 9<sup>th</sup> SCL clock Other: Reserved</p>
15:8	R/W	0x10	<p>TIMEOUT_N Timeout number When sending the 9<sup>th</sup> clock, assert fail signal when the slave device does not respond after N*F<sub>SCL</sub> cycles. And the software must do a reset to the TWI_DRV module and send a stop condition to slave.</p>
7:2	/	/	/
1	R/W	0x0	<p>SOFT_RESET Software reset 0: Normal 1: Reset</p>
0	R/W	0x0	<p>TWI_DRV_EN TWI driver enable 0: Module disable 1: Module enable (only use in TWI Master Mode)</p>

#### 8.1.6.11 0x0204 TWI\_DRV Transmission Configuration Register (Default Value: 0x0000\_0001)

Offset: 0x0204			Register Name: TWI_DRV_CFG
Bit	Read/Write	Default/Hex	Description
31:16	R/W	0x0	<b>PKT_INTERVAL</b> Define the interval between each packet for PKT_INTERVAL $F_{SCL}$ cycles.
15:0	R/W	0x1	<b>PACKET_CNT</b> The FIFO data is transmitted as PACKET_CNT packets in current format.

#### 8.1.6.12 0x0208 TWI\_DRV Slave ID Register (Default Value: 0x0000\_0000)

Offset: 0x0208			Register Name: TWI_DRV_SLV
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:9	R/W	0x0	<b>SLV_ID</b> Slave device ID For 7-bit addressing, the bit[7:1] indicates: SLA6, SLA5, SLA4, SLA3, SLA2, SLA1, SLA0 For 10-bit addressing, the bit[7:1] indicates: 1, 1, 1, 1, 0, SLAX[9:8]
8	R/W	0x0	<b>CMD</b> R/W operation to slave device 0: Write 1: Read
7:0	R/W	0x0	<b>SLV_ID_X</b> SLAX[7:0] The low 8 bits for slave device ID with 10-bit addressing.

#### 8.1.6.13 0x020C TWI\_DRV Packet Format Register (Default Value: 0x0001\_0001)

Offset: 0x020C			Register Name: TWI_DRV_FMT
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/

Offset: 0x020C			Register Name: TWI_DRV_FMT
Bit	Read/Write	Default/Hex	Description
23:16	R/W	0x1	ADDR_BYTE How many bytes be sent as slave device reg address 0~255
15:0	R/W	0x1	DATA_BYTE How many bytes be sent/received as data 1~65535

#### 8.1.6.14 0x0210 TWI\_DRV Bus Control Register (Default Value: 0x0000\_80C0)

Offset: 0x0210			Register Name: TWI_DRV_BUS_CTRL
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	W	0x0	CLK_COUNT_MODE Clock count mode 0: scl clock high period count on oscl 1: scl clock high period count on iscl
15	R/W	0x1	CLK_DUTY Setting duty cycle of clock as Master 0: 50% 1: 40%
14:12	R/W	0x0	CLK_N TWI_DRV bus sampling clock F0=24MHz/2^CLK_N
11:8	R/W	0x0	CLK_M TWI_DRV output SCL frequency is $F_{SCL}=F1/10=(F0/(CLK\_M+1))/10$ Specially, Foscl = F1/11 when CLK_M=0 and CLK_DUTY=40% due to the delay of SCL sample debounce.
7	R	0x1	SCL_STA SCL current status
6	R	0x1	SDA_STA SDA current status
5:4	/	/	/
3	R/W	0x0	SCL_MOV SCL manual output value

Offset: 0x0210			Register Name: TWI_DRV_BUS_CTRL
Bit	Read/Write	Default/Hex	Description
2	R/W	0x0	SDA_MOV SDA manual output value
1	R/W	0x0	SCL_MOE SCL manual output enable
0	R/W	0x0	SDA_MOE SDA manual output enable

#### 8.1.6.15 0x0214 TWI\_DRV Interrupt Control Register (Default Value: 0x0000\_0000)

Offset: 0x0214			Register Name: TWI_DRV_INT_CTRL
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
19	R/W	0x0	RX_REQ_INT_EN If set, an interrupt is sent when RX_REQ_PD sets.
18	R/W	0x0	TX_REQ_INT_EN If set, an interrupt is sent when TX_REQ_PD sets.
17	R/W	0x0	TRAN_ERR_INT_EN If set, an interrupt is sent when TRAN_ERR_PD sets.
16	R/W	0x0	TRAN_COM_INT_EN If set, an interrupt is sent when TRAN_COM_PD sets.
15:4	/	/	/
3	R/W1C	0x0	RX_REQ_PD Set when the data byte number in RECV_FIFO reaches RX_TRIG.
2	R/W1C	0x0	TX_REQ_PD Set when there is no less than DMA_TX_TRIG empty byte number in SEND_FIFO.
1	R/W1C	0x0	TRAN_ERR_PD Packet transmission failure pending
0	R/W1C	0x0	TRAN_COM_PD Packet transmission completion pending

**8.1.6.16 0x0218 TWI\_DRV DMA Configure Register (Default Value: 0x0010\_0010)**

Offset: 0x0218			Register Name: TWI_DRV_DMA_CFG
Bit	Read/Write	Default/Hex	Description
31:25	/	/	/
24	R/W	0x0	DMA_RX_EN DMA RX Enable
23:22	/	/	/
21:16	R/W	0x10	RX_TRIG RX trigger When DMA_RX_EN is set, send DMA RX Req when the data byte number in RECV_FIFO reaches RX_TRIG, or the read transmission is completed, the data of RECV_FIFO does not reach RX_TRIG but as long as the RECV_FIFO is not empty.
15:9	/	/	/
8	R/W	0x0	DMA_TX_EN DMA TX Enable
7:6	/	/	/
5:0	R/W	0x10	TX_TRIG TX trigger When DMA_TX_EN is set, send DMA TX Req when the space of SEND_FIFO (FIFO Level – data volume) reaches TX_TRIG.

**8.1.6.17 0x021C TWI\_DRV FIFO Content Register (Default Value: 0x0000\_0000)**

Offset: 0x021C			Register Name: TWI_DRV_FIFO_CON
Bit	Read/Write	Default/Hex	Description
31:23	/	/	/
22	R/WAC	0x0	RECV_FIFO_CLEAR Set this bit to clear RECV_FIFO pointer, and this bit is cleared automatically.
21:16	R	0x0	RECV_FIFO_CONTENT The number of data in RECV_FIFO
15:7	/	/	/
6	R/WAC	0x0	SEND_FIFO_CLEAR Set this bit to clear SEND_FIFO pointer, and this bit is cleared automatically.

Offset: 0x021C			Register Name: TWI_DRV_FIFO_CON
Bit	Read/Write	Default/Hex	Description
5:0	R	0x0	SEND_FIFO_CONTENT The number of data in SEND_FIFO

#### 8.1.6.18 0x0300 TWI\_DRV Send Data FIFO Access Register (Default Value: 0x0000\_0000)

Offset: 0x0300			Register Name: TWI_DRV_SEND_FIFO_ACC
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	W	0x0	SEND_DATA_FIFO Address of a 32x8 SEND_FIFO, which stores reg address and data sending to the slave device.

#### 8.1.6.19 0x0304 TWI\_DRV Receive Data FIFO Access Register (Default Value: 0x0000\_0000)

Offset: 0x0304			Register Name: TWI_DRV_RECV_FIFO_ACC
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R	0x0	RECV_DATA_FIFO Address of a 32x8 RECV_FIFO, which stores data received from the slave device.

## 8.2 UART

### 8.2.1 Overview

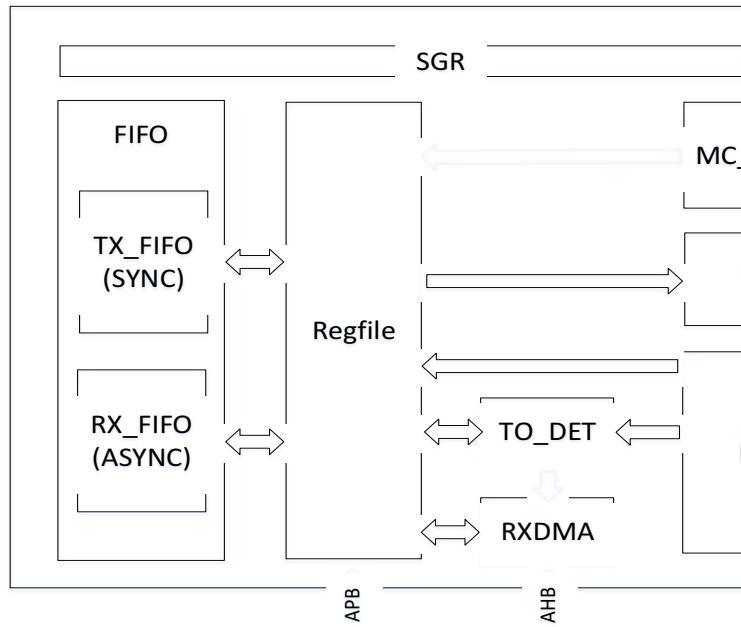
The universal asynchronous receiver transmitter (UART) provides an asynchronous serial communication with external devices, modem (data carrier equipment, DCE). It performs serial-to-parallel conversion on the data received from peripherals and transmits the converted data to the internal bus. It also performs parallel-to-serial conversion on the data that is transmitted to peripherals.

The UART has the following features:

- Compatible with industry-standard 16450/16550 UARTs
- Supports IrDA-compatible slow infrared (SIR) format
- Two separate FIFOs: one is RX FIFO, and the other is TX FIFO
  - Each of them is 64 bytes (For UART0)
  - Each of them is 256 bytes (For UART1, UART2, UART3, UART4, and UART5)
- The working reference clock is from the APB bus clock
  - Speed up to 4 Mbit/s with 64 MHz APB clock
  - Speed up to 1.5 Mbit/s with 24 MHz APB clock
- 5 to 8 data bits for RS-232 characters, or 9 bits RS-485 format
- 1, 1.5 or 2 stop bits
- Programmable parity (even, odd, or no parity)
- Supports TX/RX DMA slave controller interface
- Supports software/hardware flow control
- Supports RX DMA Master interface (Only for UART1)
- Supports auto-flow by using CTS & RTS (Only for UART1/2/3)

### 8.2.2 Block Diagram

Figure 8-9 shows a block diagram of the UART.

**Figure 8-9 UART Block Diagram**

### 8.2.3 Functional Description

#### 8.2.3.1 External Signals

The following table describes the external signals of UART.

**Table 8-4 UART External Signals**

Signal	Description	Type
UART0-TX	UART0 Data Transmit	O
UART0-RX	UART0 Data Receive	I
UART1-TX	UART1 Data Transmit	O
UART1-RX	UART1 Data Receive	I
UART1-CTS	UART1 Data Clear to Send	I
UART1-RTS	UART1 Data Request to Send	O
UART2-TX	UART2 Data Transmit	O
UART2-RX	UART2 Data Receive	I
UART2-CTS	UART2 Data Clear to Send	I
UART2-RTS	UART2 Data Request to Send	O
UART3-TX	UART3 Data Transmit	O
UART3-RX	UART3 Data Receive	I

Signal	Description	Type
UART3-CTS	UART3 Data Clear to Send	I
UART3-RTS	UART3 Data Request to Send	O
UART4-TX	UART4 Data Transmit	O
UART4-RX	UART4 Data Receive	I
UART5-TX	UART5 Data Transmit	O
UART5-RX	UART5 Data Receive	I

### 8.2.3.2 Clock Sources

The following table describes the clock sources of UART.

**Table 8-5 UART Clock Sources**

Clock Sources	Description
APB1 Bus	UART clock source. Refer to CCU for details on APB1.

### 8.2.3.3 Typical Applications and Timing Diagram

#### UART Serial Data Format

The following figure shows the UART serial data format. The start bit, data bit, parity bit, and stop bit can be configured.

**Figure 8-10 UART Serial Data Format**

#### Using UART for RTS/CTS Autoflow Control

Figure 8-11 shows the typical application diagram for RTS/CTS autoflow control. Figure 8-12 shows the data format of the RTS/CTS autoflow control.

Figure 8-11 Application Diagram for RTS/CTS Autoflow Control

Figure 8-12 RTS/CTS Autoflow Control Data Format

### Using UART for Serial IrDA

Figure 8-13 shows the application diagram for the IrDA transceiver. Figure 8-14 shows the data format of the serial IrDA.

Figure 8-13 Application Diagram for IrDA Transceiver

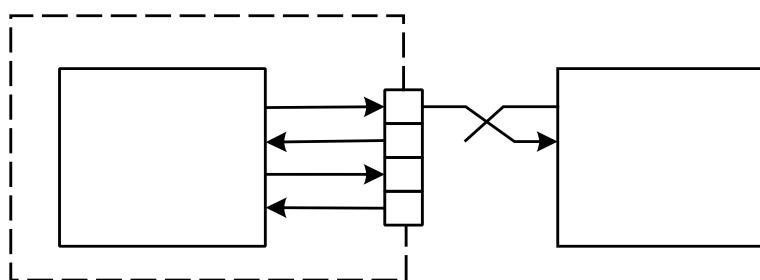


Figure 8-14 Serial IrDA Data Format

### Using UART for RS-485

Figure 8-15 shows the application diagram for the RS-485 transceiver. Figure 8-16 shows the data format of the RS-485.

Figure 8-15 Application Diagram for RS-485 Transceiver

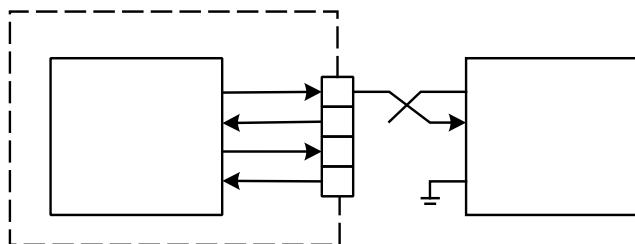


Figure 8-16 RS-485 Data Format

#### 8.2.3.4 UART Operating Mode

##### Data Frame Format

The [UART\\_LCR](#) register can set the basic parameter of a data frame: data width (5 to 8 bits), stop bit number (1/1.5/2), parity type.

A frame transfer of the UART includes the start signal, data signal, parity bit, and stop signal. The LSB is transmitted first.

- Start signal (start bit): It is the start flag of a data frame. According to the UART protocol, the low level of the TXD signal indicates the start of a data frame. When the UART transmits data, the level needs to hold high.
- Data signal (data bit): The data bit width can be configured as 5-bit, 6-bit, 7-bit, and 8-bit through different applications.
- Parity bit: It is a 1-bit error correction signal. Parity bit includes odd parity, even parity. The UART can enable and disable the parity bit by setting the [UART\\_LCR](#) register.
- Stop Signal (stop bit): It is the stop bit of a data frame. The stop bit can be set to 1-bit, 1.5-bit, and 2-bit by the [UART\\_LCR](#) register. The high level of the TXD signal indicates the end of a data frame.

## Baud and Error Rates

The baud rate is calculated as follows: Baud rate = SCLK/(16 \* divisor).

The SCLK is usually APB1 and can be set in section 3.3 "[CCU](#)".

The divisor is frequency divider of UART. The frequency divider has 16-bit, the low 8-bit is in the [UART\\_DLL](#) register, the high 8-bit is in the [UART\\_DLH](#) register.

The relationship between the different UART mode and the error rate is as follows.

**Figure 8-17 UART Mode Baud and Error Rates**

Clock source	Divisor	Baud rate	Over sampling	Error (%)
24000000	5000	300	16	0
24000000	2500	600	16	0
24000000	1250	1200	16	0
24000000	625	2400	16	0
24000000	313	4800	16	-0.16
24000000	156	9600	16	0.16
24000000	78	19200	16	0.16
24000000	39	38400	16	0.16
24000000	26	57600	16	0.16
24000000	13	115200	16	0.16
48000000	13	230400	16	0.16
64000000	7	576000	16	-0.794
75000000	5	921600	16	1.725

Clock source	Divisor	Baud rate	Over sampling	Error (%)
48000000	3	1000000	16	0
24000000	1	1500000	16	0
48000000	1	3000000	16	0
64000000	1	4000000	16	0

**Figure 8-18 IrDA Mode Baud and Error Rates**

Clock source	Divisor	Baud rate	Encoding	Error (%)
24000000	5000	300	3/16	0
24000000	2500	600	3/16	0
24000000	1250	1200	3/16	0
24000000	625	2400	3/16	0
24000000	313	4800	3/16	-0.16
24000000	156	9600	3/16	0.16
24000000	78	19200	3/16	0.16
24000000	39	38400	3/16	0.16
24000000	26	57600	3/16	0.16
24000000	13	115200	3/16	0.16

**Figure 8-19 RS485 Mode Baud and Error Rates**

Clock source	Divisor	Baud rate	Encoding	Error (%)
24000000	5000	300	16	0
24000000	2500	600	16	0
24000000	1250	1200	16	0
24000000	625	2400	16	0
24000000	313	4800	16	-0.16
24000000	156	9600	16	0.16
24000000	78	19200	16	0.16
24000000	39	38400	16	0.16
24000000	26	57600	16	0.16
24000000	13	115200	16	0.16

## DLAB Definition

The DLAB control bit ([UART\\_LCR](#)[7]) is the access control bit of the divisor Latch register.

If DLAB is 0, then the 0x00 offset address is the [UART\\_RBR/UART\\_THR](#) (RX/TX FIFO) register, and the 0x04 offset address is the [UART\\_IER](#) register.

If DLAB is 1, then the 0x00 offset address is the [UART\\_DLL](#) register, and the 0x04 offset address is the [UART\\_DLH](#) register.

When the UART initializes, the divisor needs to be set. That is, writing 1 to DLAB can access the [UART\\_DLL](#) and [UART\\_DLH](#) register, after finished the configuration, writing 0 to DLAB can access the [UART\\_RBR/UART\\_THR](#) register.

## CHCFG\_AT\_BUSY Definition

The function of the CHCFG\_AT\_BUSY ([UART\\_HALT](#)[1]) and CHANGE\_UPDATE ([UART\\_HALT](#)[2]) are as follows.

**CHCFG\_AT\_BUSY:** Enable the bit, the software can also set the UART controller when UART is busy, such as the [UART\\_LCR](#), [UART\\_DLH](#), [UART\\_DLL](#) register.

**CHANGE\_UPDATE:** If CHCFG\_AT\_BUSY is enabled, and CHANGE\_UPDATE is written to 1, the configuration of the UART controller can be updated. After completed the update, the bit is cleared to 0 automatically.

Setting divisor performs the following steps:

**Step 1** Write 1 to CHCFG\_AT\_BUSY to enable “configure at busy”.

**Step 2** Write 1 to DLAB ([UART\\_LCR](#)[7]) and set the [UART\\_DLH](#) and [UART\\_DLL](#) registers.

**Step 3** Write 1 to CHANGE\_UPDATE to update the configuration. The bit is cleared to 0 automatically after completing the update.

## UART Busy Flag

The [UART\\_USR](#)[0] is a busy flag of the UART controller.

When the TX transmits data, or the RX receives data, or the TX FIFO is not empty, or the RX FIFO is not empty, then the busy flag bit can be set to 1 by hardware, which indicates the UART controller is busy.

### 8.2.4 Programming Guidelines

The following takes the UART module in the CPUX domain as an example.

### 8.2.4.1 Initialization

#### Step 1 System Initialization

- Configure [APB1\\_CFG\\_REG](#) in the CCU module to set the APB1 bus clock (The clock is 24MHz by default).
- Set [UART\\_BGR\\_REG](#)[UARTx\_GATING] to 1 to enable the module clock, and set [UART\\_BGR\\_REG](#)[UARTx\_RST] to 1 to de-assert the module.

#### Step 2 UART Controller Initialization

- IO configuration: Configure GPIO multiplex as UART function, and set UART pins to internal pull-up mode (For detail, see the description in section 9.7 “[GPIO](#)”).
- Baud-rate configuration:
  - Set UART baud-rate (refer to section 9.2.3.4);
  - Write [UART\\_FCR](#)[FIFOE] to 1 to enable TX/RX FIFO;
  - Write [UART\\_HALT](#)[HALT\_TX] to 1 to disable TX transfer;
  - Set [UART\\_LCR](#)[DLAB] to 1, remain default configuration for other bits; set 0x00 offset address to the [UART\\_DLL](#) register, set 0x04 offset address to the [UART\\_DLH](#) register;
  - Write the high 8-bit of divisor to the [UART\\_DLH](#) register, and write the low 8-bit of divisor to the [UART\\_DLL](#) register;
  - Set [UART\\_LCR](#)[DLAB] to 0, remain default configuration for other bits; set 0x00 offset address to the [UART\\_RBR/UART\\_THR](#) register, set 0x04 offset address to the [UART\\_IER](#) register;
  - Set [UART\\_HALT](#)[HALT\_TX] to 0 to enable TX transfer.

#### Step 3 Controller Parameter Configuration

- Set data width, stop bits, and even/odd parity type by writing the [UART\\_LCR](#) register.
- Reset, enable FIFO and set FIFO trigger condition by writing the [UART\\_FCR](#) register.
- Set the flow control parameter by writing the [UART\\_MCR](#) register.

#### Step 4 Interrupt Configuration

- Configure UART interrupt vector number to request UART interrupt (Refer to section 3.8 “GIC” module for interrupt vector number).
- In DMA mode, write [UART\\_IER](#) to 0 to disable interrupt; write [UART\\_HSK](#)[Handshake configuration] to 0xE5 to set DMA handshake mode; write [UART\\_FCR](#)[DMAM] to 1 to set DMA transmission/reception mode; set DMA parameter and request DMA interrupt according to DMA configuration process.

- In Interrupt mode, configure [UART\\_IER](#) to enable the corresponding interrupt according to requirements: such as transmit (TX) interrupt, receive (RX) interrupt, receive line status interrupt, RS48 interrupt, etc. (Here TX/RX interrupt is usually used).

#### 8.2.4.2 Transferring/Receiving Data in Query Mode

##### Data transfer

**Step 1** Write data to [UART\\_THR](#) to start data transfer.

**Step 2** Check TX\_FIFO status by reading [UART\\_USR](#)[TFNF]. If the bit is 1, data can continue to be written; if the bit is 0, wait for data transfer, and data cannot continue to write until FIFO is not full.

##### Data receive

**Step 1** Check RX\_FIFO status by reading [UART\\_USR](#)[RFNE].

**Step 2** Read data from [UART\\_RBR](#) if RX\_FIFO is not empty.

**Step 3** If [UART\\_USR](#)[RFNE] is 0, data is received completely.

#### 8.2.4.3 Transferring/Receiving Data in Interrupt Mode

##### Data transfer

**Step 1** Set [UART\\_IER](#)[ETBEI] to 1 to enable the *UART transmission interrupt*.

**Step 2** Write the data to be transmitted to [UART\\_THR](#).

**Step 3** When the data of TX\_FIFO meets trigger condition (such as FIFO/2, FIFO/4), the UART transfer interrupt is generated.

**Step 4** Check [UART\\_USR](#)[TFE] and determine whether TX\_FIFO is empty. If [UART\\_USR](#)[TFE] is 1, it indicates that the data in TX\_FIFO is transmitted completely.

**Step 5** Clear [UART\\_IER](#)[ETBEI] to 0 to disable transfer interrupt.

##### Data receive

**Step 1** Set [UART\\_IER](#)[ERBFI] to 1 to enable the *UART reception interrupt*.

**Step 2** When the received data from RX\_FIFO meets trigger condition (such as FIFO/2, FIFO/4), the UART receive interrupt is generated.

**Step 3** Read data from [UART\\_RBR](#).

**Step 4** Check RX\_FIFO status by reading [UART\\_USR](#)[RFNE] and determine whether to read data. If the bit is 1, continue to read data from [UART\\_RBR](#) until [UART\\_USR](#)[RFNE] is cleared to 0, which indicates data is received completely.

#### 8.2.4.4 Transferring/Receiving Data in DMA Mode

##### Data transfer

**Step 1** Configure the UART DMA interrupt according to the initialization process.

**Step 2** Configure DMA data channel, including the transfer source address, the transfer destination address, the number of data to be transferred, and the transfer type, and so on. (For details, see section 3.9 “[DMAC](#)”).

**Step 3** Enable the DMA transfer function of the UART by setting the register of the DMA module.

**Step 4** Determine whether UART data is transferred completely based on the DMA status. If all data is transferred completely, disable the DMA transfer function of the UART.

##### Data receive

**Step 1** Configure DMA data channel, including the transfer source address, the transfer destination address, the number of data to be transferred, and the transfer type, and so on. (For details, see section 3.9 “[DMAC](#)”).

**Step 2** Enable the DMA receive function of the UART by setting the register of the DMA module.

**Step 3** Determine whether UART data is received completely based on the DMA status. If all data is received completely, disable the DMA receive function of the UART.

#### 8.2.5 Register List

Module Name	Base Address
UART0	0x02500000
UART1	0x02500400
UART2	0x02500800
UART3	0x02500C00
UART4	0x02501000
UART5	0x02501400

<b>Register Name</b>	<b>Offset</b>	<b>Description</b>
UART_RBR	0x0000	UART Receive Buffer Register
UART_THR	0x0000	UART Transmit Holding Register
UART_DLL	0x0000	UART Divisor Latch Low Register
UART_DLH	0x0004	UART Divisor Latch High Register
UART_IER	0x0004	UART Interrupt Enable Register
UART_IIR	0x0008	UART Interrupt Identity Register
UART_FCR	0x0008	UART FIFO Control Register
UART_LCR	0x000C	UART Line Control Register
UART_MCR	0x0010	UART Modem Control Register
UART_LSR	0x0014	UART Line Status Register
UART_MSR	0x0018	UART Modem Status Register
UART_SCH	0x001C	UART Scratch Register
UART_USR	0x007C	UART Status Register
UART_TFL	0x0080	UART Transmit FIFO Level Register
UART_RFL	0x0084	UART Receive FIFO Level Register
UART_HSK	0x0088	UART DMA Handshake Configuration Register
UART_DMA_REQ_EN	0x008C	UART DMA Request Enable Register
UART_HALT	0x00A4	UART Halt TX Register
UART_DBG_DLL	0x00B0	UART Debug DLL Register
UART_DBG_DLH	0x00B4	UART Debug DLH Register
UART_A_FCC	0x00F0	UART FIFO Clock Control Register
UART_A_RXDMA_CTRL	0x0100	UART RXDMA Control Register
UART_A_RXDMA_STR	0x0104	UART RXDMA Start Register
UART_A_RXDMA_STA	0x0108	UART RXDMA Status Register
UART_A_RXDMA_LMT	0x010C	UART RXDMA Limit Register
UART_A_RXDMA_SADDRL	0x0110	UART RXDMA Buffer Start Address Low Register
UART_A_RXDMA_SADDRH	0x0114	UART RXDMA Buffer Start Address High Register
UART_A_RXDMA_BL	0x0118	UART RXDMA Buffer Length Register
UART_A_RXDMA_IE	0x0120	UART RXDMA Interrupt Enable Register
UART_A_RXDMA_IS	0x0124	UART RXDMA Interrupt Status Register
UART_A_RXDMA_WADDRL	0x0128	UART RXDMA Write Address Low Register
UART_A_RXDMA_WADDRH	0x012C	UART RXDMA Write Address high Register

Register Name	Offset	Description
UART_A_RXDMA_RADDRL	0x0130	UART RXDMA Read Address Low Register
UART_A_RXDMA_RADDRH	0x0134	UART RXDMA Read Address high Register
UART_A_RXDMA_DCNT	0x0138	UART RADMA Data Count Register

## 8.2.6 Register Description

### 8.2.6.1 0x0000 UART Receiver Buffer Register (Default Value: 0x0000\_0000)

Offset: 0x0000			Register Name: UART_RBR
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R	0x0	<p>RBR Receiver Buffer Register Data byte received on the serial input port (sin) in UART mode, or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in <a href="#">UART_LCR</a> is set. If in FIFO mode and FIFOs are enabled (The <a href="#">UART_FCR</a>[0] is set to 1), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register can not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data are lost and an overrun error occurs.</p>

### 8.2.6.2 0x0000 UART Transmit Holding Register (Default Value: 0x0000\_0000)

Offset: 0x0000			Register Name: UART_THR
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/

Offset: 0x0000			Register Name: UART_THR
Bit	Read/Write	Default/Hex	Description
7:0	W	0x0	<p>THR</p> <p>Transmit Holding Register</p> <p>Data is transmitted on the serial output port (SOUT) in UART mode or the serial infrared output (sir_out_n) in infrared mode.</p> <p>Data should only be written to the <a href="#">UART_THR</a> when the THRE bit (<a href="#">UART_LSR[5]</a>) is set.</p> <p>If in FIFO mode and FIFOs are enabled (<a href="#">UART_FCR[0]</a> = 1) and THRE is set, the 16 number of characters data may be written to the <a href="#">UART_THR</a> before the FIFO is full. When the FIFO is full, any written data results in the written data being lost.</p>

#### 8.2.6.3 0x0000 UART Divisor Latch Low Register (Default Value: 0x0000\_0000)

Offset: 0x0000			Register Name: UART_DLL
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R/W	0x0	<p>DLL</p> <p>Divisor Latch Low</p> <p>Lower 8 bits of 16 bits, read/write, Divisor Latch Register contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (<a href="#">UART_LCR[7]</a>) is set and the UART is not busy (<a href="#">UART_USR[0]</a> is 0).</p> <p>The output baud rate is equal to the serial clock (SCLK) frequency divided by sixteen times the value of the baud rate divisor, as follows: <b>baud rate = (serial clock freq)/(16 * divisor)</b>.</p> <p>Note that when the Divisor Latch Registers (<a href="#">UART_DLL</a> and <a href="#">UART_DLH</a>) are set to 0, the baud clock is disabled and no serial communications occur. Also, once the DLL is set, at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data.</p>

#### 8.2.6.4 0x0004 UART Divisor Latch High Register (Default Value: 0x0000\_0000)

Offset: 0x0004			Register Name: UART_DLH
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/

Offset: 0x0004			Register Name: UART_DLH
Bit	Read/Write	Default/Hex	Description
7:0	R/W	0x0	<p>DLH Divisor Latch High</p> <p>Upper 8 bits of 16 bits, read/write, Divisor Latch Register contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (<a href="#">UART_LCR</a>[7]) is set and the UART is not busy (<a href="#">UART_USR</a>[0] is 0).</p> <p>The output baud rate is equal to the serial clock (SCLK) frequency divided by sixteen times the value of the baud rate divisor, as follows: <b>baud rate = (serial clock freq)/(16 * divisor)</b>.</p> <p>Note that when the Divisor Latch Registers (<a href="#">UART_DLL</a> and <a href="#">UART_DLH</a>) is set to 0, the baud clock is disabled and no serial communications occur. Also, once the DLH is set, at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data.</p>

#### 8.2.6.5 0x0004 UART Interrupt Enable Register (Default Value: 0x0000\_0000)

Offset: 0x0004			Register Name: UART_IER
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R/W	0x0	<p>PTIME</p> <p>Programmable THRE Interrupt Mode Enable</p> <p>This is used to enable/disable the generation of THRE Interrupt.</p> <p>0: Disable 1: Enable</p>
6:5	/	/	/
4	R/W	0x0	<p>RS485_INT_EN</p> <p>RS485 Interrupt Enable</p> <p>0: Disable 1: Enable</p>
3	R/W	0x0	<p>EDSSI</p> <p>Enable Modem Status Interrupt</p> <p>This is used to enable/disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt.</p> <p>0: Disable 1: Enable</p>

Offset: 0x0004			Register Name: UART_IER
Bit	Read/Write	Default/Hex	Description
2	R/W	0x0	<p>ELSI Enable Receiver Line Status Interrupt This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt. 0: Disable 1: Enable</p>
1	R/W	0x0	<p>ETBEI Enable Transmit Holding Register Empty Interrupt This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt. This is the third-highest priority interrupt. 0: Disable 1: Enable</p>
0	R/W	0x0	<p>ERBFI Enable Received Data Available Interrupt This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFOs enabled). These are the second-highest priority interrupt. 0: Disable 1: Enable</p>

#### 8.2.6.6 0x0008 UART Interrupt Identity Register (Default Value: 0x0000\_0001)

Offset: 0x0008			Register Name: UART_IIR
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:6	R	0x0	<p>FEFLAG FIFOs Enable Flag This is used to indicate whether the FIFOs are enabled or disabled. 00: Disable 11: Enable</p>
5:4	/	/	/

Offset: 0x0008			Register Name: UART_IIR
Bit	Read/Write	Default/Hex	Description
3:0	R	0x1	<p>IID Interrupt ID This indicates the highest priority pending interrupt which can be one of the following types.</p> <p>0000: modem status 0001: no interrupt pending 0010: THR empty 0011: RS485 Interrupt 0100: received data available 0110: receiver line status 0111: busy detect 1100: character timeout The bit 3 indicates an interrupt can only occur when the FIFOs are enabled and used to distinguish a Character Timeout condition interrupt.</p>

Interrupt ID	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset
0001	-	None	None	-
0110	Highest	Receiver line status	Overrun/parity/framing errors or break interrupt	Reading the line status register
0011	Second	RS485 Interrupt	In RS485 mode, receives address data and match setting address	Writes 1 to addr flag to reset
0100	Third	Received data available	Receiver data available (non-FIFO mode or FIFOs disabled) or RCVR FIFO trigger level reached (FIFO mode and FIFOs enabled)	Reading the receiver buffer register (non-FIFO mode or FIFOs disabled) or the FIFO drops below the trigger level (FIFO mode and FIFOs enabled)
1100	Fourth	Character timeout indication	No characters in or out of the RCVR FIFO during the last 4 character times and there is at least 1character in it during This time	Reading the receiver buffer register

Interrupt ID	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset
0010	Fifth	Transmit holding register empty	Transmitter holding register empty (Program THRE mode disabled) or XMIT FIFO at or below threshold (Program THRE mode enabled)	Reading the IIR register (if the source of interrupt); or, writing into THR (FIFOs or THRE mode not selected or disabled) or XMIT FIFO above threshold (FIFOs and THRE mode selected and enabled).
0000	Sixth	Modem status	Clear to send or data set ready or ring indicator or data carrier detect. Note that if the autoflow control mode is enabled, a change in CTS (that is, DCTS set) does not cause an interrupt.	Reading the Modem status register
0111	Seventh	Busy detect indication	UART_16550_COMPATIBLE = NO and master has tried to write to the Line Control Register while the UART is busy (USR[0] is set to one).	Reading the UART status register

#### 8.2.6.7 0x0008 UART FIFO Control Register (Default Value: 0x0000\_0000)

Offset: 0x0008			Register Name: UART_FCR
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:6	W	0x0	<p>RT RCVR Trigger</p> <p>This is used to select the trigger level in the receiver FIFO when the Received Data Available Interrupt is generated. In the autoflow control mode, it is used to determine when the rts_n signal is de-asserted. It also determines when the dma_rx_req_n signal is asserted in certain modes of operation.</p> <p>00: 1 character in the FIFO 01: FIFO ¼ full 10: FIFO ½ full 11: FIFO-2 less than full</p>

Offset: 0x0008			Register Name: UART_FCR
Bit	Read/Write	Default/Hex	Description
5:4	W	0x0	<p>TFT TX Empty Trigger</p> <p>This is used to select the empty threshold level when the THRE Interrupts are generated and the mode is active. It also determines when the dma_tx_req_n signal is asserted in certain modes of operation.</p> <p>00: FIFO empty 01: 2 characters in the FIFO 10: FIFO ¼ full 11: FIFO ½ full</p>
3	W	0x0	<p>DMAM DMA Mode 0: Mode 0</p> <p>In this mode, when the PTE in UART_HALT is high and TX FIFO is enabled, the TX DMA request will be set when the TFL in UART_TFL is less than or equal to FIFO Trigger Level (otherwise it will be cleared). When the PTE is high and TX FIFO is disabled, the TX DMA request will be set only if the THR in UART_THR is empty. If the PTE is low, the TX DMA request will be set only if the TX FIFO (TX FIFO enabled) or THR (TX FIFO disabled) is empty.</p> <p>When the DMA_PTE_RX in UART_HALT is high and RX FIFO is enabled, the RX DRQ will be set only if the RFL in UART_RFL is equal to or more than FIFO Trigger Level, otherwise, it will be cleared.</p> <p>1: Mode 1</p> <p>In this mode, TX FIFO should be enabled. If the PTE in <a href="#">UART_HALT</a> is high, the TX DMA request will be set when the TFL in <a href="#">UART_TFL</a> is less than or equal to FIFO Trigger Level; If the PTE is low, the TX DMA request will be set when TX FIFO is empty. Once the request is set, it is cleared only when TX FIFO is full.</p> <p>If the RFL in UART_RFL is equal to or more than FIFO Trigger Level or there is a character timeout, the RX DRQ will be set; Once the RX DRQ is set, it is cleared only when RX FIFO (RX FIFO enabled) or RBR (RX FIFO disabled) is empty.</p>
2	W	0x0	<p>XFIFOR XMIT FIFO Reset</p> <p>The bit resets the control part of the transfer FIFO and treats the FIFO as empty. This also de-asserts the DMA TX request.</p> <p>It is 'self-cleared'. It is not necessary to clear this bit.</p>

Offset: 0x0008			Register Name: UART_FCR
Bit	Read/Write	Default/Hex	Description
1	W	0x0	<p>RFIFOR RCVR FIFO Reset</p> <p>The bit resets the control part of the receive FIFO and treats the FIFO as empty. This also de-asserts the DMA RX request.</p> <p>It is 'self-cleared'. It is not necessary to clear this bit.</p>
0	W	0x0	<p>FIFOE Enable FIFOs</p> <p>The bit enables/disables the transmitting (XMIT) and receiving (RCVR) FIFOs. Whenever the value of this bit is changed, both the XMIT and RCVR controller part of FIFOs is reset.</p>

#### 8.2.6.8 0x000C UART Line Control Register (Default Value: 0x0000\_0000)

Offset: 0x000C			Register Name: UART_LCR
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R/W	0x0	<p>DLAB Divisor Latch Access Bit</p> <p>It is writable only when UART is not busy (UART_USR[0] is 0) and always readable. This bit is used to enable the reading and writing of the Divisor Latch register (UART_DLL and UART_DLH) to set the baud rate of the UART. This bit must be cleared after the initial baud rate setup in order to access other registers.</p> <p>0: Select RX Buffer Register (UART_RBR)/TX Holding Register (UART_THR) and Interrupt Enable Register (UART_IER)</p> <p>1: Select Divisor Latch LS Register (UART_DLL) and Divisor Latch MS Register (UART_DLM)</p>
6	R/W	0x0	<p>BC Break Control Bit</p> <p>This is used to cause a break condition to be transmitted to the receiving device. If set to 0, the serial output is forced to the spacing (logic 0) state. When not in Loopback mode, as determined by UART_MCR[4], the SOUT line is forced low until the Break bit is cleared. If SIR_MODE is enabled and active (UART_MCR[6] is set to 1), the sir_out_n line is continuously pulsed. When in Loopback mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low.</p>

Offset: 0x000C			Register Name: UART_LCR
Bit	Read/Write	Default/Hex	Description
5:4	R/W	0x0	<p>EPS Even Parity Select</p> <p>It is writable only when UART is not busy (UART_USR[0] is 0). This is used to select the even and odd parity when the PEN is enabled (the UART_LCR[3] is set to 1). Setting the UART_LCR[5] is unset to reverse the LCR[4].</p> <p>00: Odd Parity 01: Even Parity 1X: Reverse LCR[4]</p> <p>In RS485 mode, it is the 9th bit--address bit.</p> <p>11: 9th bit = 0, indicates that this is a data byte. 10: 9th bit = 1, indicates that this is an address byte.</p> <p><b>Note: When using this function, the PEN(UART_LCR[3]) must set to 1.</b></p>
3	R/W	0x0	<p>PEN Parity Enable</p> <p>It is writable only when UART is not busy (UART_USR[0] is 0) and always readable. This bit is used to enable and disable parity generation and detection in transmitted and received serial characters respectively.</p> <p>0: Parity disabled 1: Parity enabled</p>
2	R/W	0x0	<p>STOP Number of stop bits</p> <p>It is writable only when UART is not busy (UART_USR[0] is 0) and always readable. This is used to select the number of stop bits per character that the peripheral transmits and receives. If set to 0, one stop bit is transmitted in the serial data. If set to 1 and the data bits are set to 5 (UART_LCR[1:0] is 0), one and a half stop bit is transmitted. Otherwise, two stop bits are transmitted. Note that regardless of the number of stop bits selected, the receiver checks only the first stop bit.</p> <p>0: 1 stop bit 1: 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit</p>

Offset: 0x000C			Register Name: UART_LCR
Bit	Read/Write	Default/Hex	Description
1:0	R/W	0x0	<p>DLS Data Length Select It is writable only when UART is not busy (UART_USR[0] is 0) and always readable. This is used to select the count of bits in a transmitted or received frame.</p> <p>00: 5 bits 01: 6 bits 10: 7 bits 11: 8 bits</p>

#### 8.2.6.9 0x0010 UART Modem Control Register (Default Value: 0x0000\_0000)

Offset: 0x0010			Register Name: UART_MCR
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:6	R/W	0x0	<p>UART_FUNCTION Select IrDA or RS485 00: UART Mode 01: IrDA SIR Mode 10: RS485 Mode 11: Reserved</p>
5	R/W	0x0	<p>AFCE Auto Flow Control Enable When FIFOs are enabled and the AFCE bit is set, the AutoFlow Control is enabled. 0: Auto Flow Control mode disabled 1: Auto Flow Control mode enabled</p>

Offset: 0x0010			Register Name: UART_MCR
Bit	Read/Write	Default/Hex	Description
4	R/W	0x0	<p>LOOP</p> <p>Loop Back Mode</p> <p>0: Normal Mode</p> <p>1: Loop Back Mode</p> <p>This is used to put the UART into a diagnostic mode for test purposes. If operating in UART mode (SIR_MODE != Enabled or not active, UART_MCR[6] is set to 0), the data on the SOUT line is held high, while serial data output is looped back to the sin line, internally. In this mode, all the interrupts are fully functional. Also, in loopback mode, the modem control inputs (dsr_n, cts_n, ri_n, dcd_n) are disconnected and the modem control outputs (dtr_n, rts_n, out1_n, out2_n) are looped back to the inputs, internally. If operating in infrared mode (SIR_MODE == Enabled AND active, UART_MCR[6] is set to 1), the data on the sir_out_n line is held low, while serial data output is inverted and looped back to the sir_in line.</p>
3:2	/	/	/
1	R/W	0x0	<p>RTS</p> <p>Request to Send</p> <p>This is used to directly control the Request to Send (rts_n) output. The RTS (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data. When Auto RTS Flow Control is not enabled (UART_MCR[5] is set to 0), the rts_n signal is set low by programming UART_MCR[1] (RTS) to a high. In Auto Flow Control, AFCE_MODE == Enabled and active (UART_MCR[5] is set to 1) and FIFOs enable (UART_FCR[0] is set to 1), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold). The rts_n signal is de-asserted when UART_MCR[1] is set low.</p> <p>0: rts_n de-asserted (logic 1)</p> <p>1: rts_n asserted (logic 0)</p> <p>Note that in Loopback mode (UART_MCR[4] is set to 1), the rts_n output is held inactive high while the value of this location is internally looped back to an input.</p>

Offset: 0x0010			Register Name: UART_MCR
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	<p>DTR Data Terminal Ready</p> <p>This is used to directly control the Data Terminal Ready (dtr_n) output. The value written to this location is inverted and driven out on dtr_n.</p> <p>0: dtr_n de-asserted (logic 1) 1: dtr_n asserted (logic 0)</p> <p>The DTR output is used to inform the modem or data set that the UART is ready to establish communications.</p> <p>Note that in Loopback mode (UART_MCR[4] is set to 1), the dtr_n output is held inactive high while the value of this location is internally looped back to an input.</p>

#### 8.2.6.10 0x0014 UART Line Status Register (Default Value: 0x0000\_0060)

Offset:0x0014			Register Name: UART_LSR
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R	0x0	<p>FIFOERR RX Data Error in FIFO</p> <p>When FIFOs are disabled, this bit is always 0. When FIFOs are enabled, this bit is set to "1" when there is at least one PE, FE, or BI in the RX FIFO. It is cleared by reading from the UART_LSR register, there are no subsequent errors in the FIFO.</p>
6	R	0x1	<p>TEMT Transmitter Empty</p> <p>If the FIFOs are disabled, this bit is set to "1" whenever the TX Holding Register (UART_THR) and the TX Shift Register are empty. If the FIFOs are enabled, this bit is set whenever the TX FIFO and the TX Shift Register are empty. In both cases, this bit is cleared when a byte is written to the TX data channel.</p>

Offset:0x0014			Register Name: UART_LSR
Bit	Read/Write	Default/Hex	Description
5	R	0x1	<p>THRE TX Holding Register Empty</p> <p>If the FIFOs are disabled, this bit is set to "1" when the TX Holding Register (UART_THR) is empty and ready to accept new data and it is cleared when the CPU writes to the TX Holding Register.</p> <p>If the FIFOs are enabled, this bit is set to "1" whenever the TX FIFO is empty and it is cleared when at least one byte is written to the TX FIFO.</p>
4	R	0x0	<p>BI Break Interrupt</p> <p>This is used to indicate the detection of a break sequence on the serial input data.</p> <p>If in UART mode (SIR_MODE == Disabled), it is set when the serial input, sir_in, is held in a logic '0' state for longer than the sum of <i>start time + data bits + parity + stop bits</i>.</p> <p>If in infrared mode (SIR_MODE == Enabled), it is set when the serial input, sir_in, is continuously pulsed to logic '0' for longer than the sum of <i>start time + data bits + parity + stop bits</i>. A break condition on serial input causes one and only one character, consisting of all zeros, to be received by the UART.</p> <p>In the FIFO mode, the character associated with the break condition is carried through the FIFO and is revealed when the character is at the top of the FIFO. Reading the UART_LSR clears the BI bit. In the non-FIFO mode, the BI indication occurs immediately and persists until the UART_LSR is read.</p>

Offset:0x0014			Register Name: UART_LSR
Bit	Read/Write	Default/Hex	Description
3	RC	0x0	<p>FE Framing Error</p> <p>This is used to indicate the occurrence of a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data.</p> <p>In the FIFO mode, since the framing error is associated with a character received, it is revealed when the character with the framing error is at the top of the FIFO. When a framing error occurs, the UART tries to resynchronize. It does this by assuming that the error was due to the start bit of the next character and then continues receiving the other bit i.e. data, and/or parity and stop. It should be noted that the Framing Error (FE) bit (UART_LSR[3]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (UART_LSR[4]).</p> <p>0: no framing error 1: framing error</p> <p>Reading the UART_LSR clears the FE bit.</p>
2	RC	0x0	<p>PE Parity Error</p> <p>This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (UART_LCR[3]) is set. In the FIFO mode, since the parity error is associated with a character received, it is revealed when the character with the parity error arrives at the top of the FIFO. It should be noted that the Parity Error (PE) bit (UART_LSR[2]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (UART_LSR[4]).</p> <p>0: no parity error 1: parity error</p> <p>Reading the UART_LSR clears the PE bit.</p>

Offset:0x0014			Register Name: UART_LSR
Bit	Read/Write	Default/Hex	Description
1	RC	0x0	<p>OE Overrun Error</p> <p>This occurs if a new data character was received before the previous data was read. In the non-FIFO mode, the OE bit is set when a new character arrives in the receiver before the previous character was read from the UART_RBR. When this happens, the data in the UART_RBR is overwritten. In the FIFO mode, an overrun error occurs when the FIFO is full and a new character arrives at the receiver. The data in the FIFO is retained and the data in the receive shift register is lost.</p> <p>0: no overrun error 1: overrun error</p> <p>Reading the UART_LSR clears the OE bit.</p>
0	R	0x0	<p>DR Data Ready</p> <p>This is used to indicate that the receiver contains at least one character in the UART_RBR or the receiver FIFO.</p> <p>0: no data ready 1: data ready</p> <p>This bit is cleared when the UART_RBR is read in non-FIFO mode, or when the receiver FIFO is empty, in FIFO mode.</p> <p><b>Note:</b> Not use when the RXDMA master is enabled (rxdma_ctrl[0] is set to 1).</p>

#### 8.2.6.11 0x0018 UART Modem Status Register (Default Value: 0x0000\_0000)

Offset: 0x0018			Register Name: UART_MSR
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R	0x0	<p>DCD Line State of Data Carrier Detect</p> <p>This is used to indicate the current state of the modem control line dcd_n. This bit is the complement of dcd_n. When the Data Carrier Detect input (dcd_n) is asserted it is an indication that the carrier has been detected by the modem or data set.</p> <p>0: dcd_n input is de-asserted (logic 1) 1: dcd_n input is asserted (logic 0)</p>

Offset: 0x0018			Register Name: UART_MSR
Bit	Read/Write	Default/Hex	Description
6	R	0x0	<p>RI Line State of Ring Indicator This is used to indicate the current state of the modem control line ri_n. This bit is the complement of ri_n. When the Ring Indicator input (ri_n) is asserted it is an indication that a telephone ringing signal has been received by setting the modem or data. 0: ri_n input is de-asserted (logic 1) 1: ri_n input is asserted (logic 0)</p>
5	R	0x0	<p>DSR Line State of Data Set Ready This is used to indicate the current state of the modem control line dsr_n. This bit is the complement of the dsr_n. When the Data Set Ready input (dsr_n) is asserted, it is an indication that the modem or data set is ready to establish communication with UART. 0: dsr_n input is de-asserted (logic 1) 1: dsr_n input is asserted (logic 0) In Loopback Mode (UART_MCR[4] is set to 1), the DSR is the same as the DTR (UART_MCR[0]).</p>
4	R	0x0	<p>CTS Line State of Clear To Send This is used to indicate the current state of the modem control line cts_n. This bit is the complement of cts_n. When the Clear to Send input (cts_n) is asserted, it is an indication that the modem or data set is ready to exchange data with UART. 0: cts_n input is de-asserted (logic 1) 1: cts_n input is asserted (logic 0) In Loopback Mode (UART_MCR[4] = 1), the CTS is the same as the RTS (UART_MCR[1]).</p>

Offset: 0x0018			Register Name: UART_MSR
Bit	Read/Write	Default/Hex	Description
3	RC	0x0	<p>DDCD Delta Data Carrier Detect This is used to indicate that the modem control line dcd_n has changed since the last time the UART_MSR was read. 0: no change on dcd_n since the last read of UART_MSR 1: change on dcd_n since the last read of UART_MSR Reading the UART_MSR clears the DDCD bit.</p> <p><b>Note:</b> If the DDCD bit is not set and the dcd_n signal is asserted (low) and a reset occurs, then the DDCD bit is set when the reset is removed if the dcd_n signal remains asserted.</p>
2	RC	0x0	<p>TERI Trailing Edge Ring Indicator This is used to indicate that a change in the input ri_n (from an active-low to an inactive-high state) has occurred since the last time the UART_MSR was read. 0: no change on ri_n since the last read of UART_MSR 1: change on ri_n since the last read of UART_MSR Reading the UART_MSR clears the TERI bit.</p>
1	RC	0x0	<p>DDSR Delta Data Set Ready This is used to indicate that the modem control line dsr_n has changed since the last time the UART_MSR was read. 0: no change on dsr_n since the last read of UART_MSR 1: change on dsr_n since the last read of UART_MSR Reading the UART_MSR clears the DDSR bit. In Loopback Mode (UART_MCR[4] = 1), the DDSR reflects changes on the DTR (UART_MCR[0]).</p> <p><b>Note:</b> If the DDSR bit is not set and the dsr_n signal is asserted (low) and a reset occurs, then the DDSR bit is set when the reset is removed if the dsr_n signal remains asserted.</p>

Offset: 0x0018			Register Name: UART_MSR
Bit	Read/Write	Default/Hex	Description
0	RC	0x0	<p>DCTS Delta Clear to Send This is used to indicate that the modem control line cts_n has changed since the last time the UART_MSR was read. 0: no change on ctsdsr_n since the last read of UART_MSR 1: change on ctsdsr_n since the last read of UART_MSR Reading the UART_MSR clears the DCTS bit. In Loopback Mode (UART_MCR[4] = 1), the DCTS reflects changes on the RTS (UART_MCR[1]).</p> <p><b>Note:</b> If the DCTS bit is not set and the cts_n signal is asserted (low) and a reset occurs, then the DCTS bit is set when the reset is removed if the cts_n signal remains asserted.</p>

#### 8.2.6.12 0x001C UART Scratch Register (Default Value: 0x0000\_0000)

Offset: 0x001C			Register Name: UART_SCH
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R/W	0x0	<p>SCRATCH_REG Scratch Register This register is for programmers to use as a temporary storage space. It has no defined purpose in the UART.</p>

#### 8.2.6.13 0x007C UART Status Register (Default Value: 0x0000\_0006)

Offset: 0x007C			Register Name: UART_USR
Bit	Read/Write	Default/Hex	Description
31:5	/	/	/
4	R	0x0	<p>RFF RX FIFO Full This is used to indicate that the RX FIFO is completely full. 0: RX FIFO not full 1: RX FIFO Full This bit is cleared when the RX FIFO is no longer full.</p>

Offset: 0x007C			Register Name: UART_USR
Bit	Read/Write	Default/Hex	Description
3	R	0x0	<p>RFNE RX FIFO Not Empty This is used to indicate that the RX FIFO contains one or more entries. 0: RX FIFO is empty 1: RX FIFO is not empty This bit is cleared when the RX FIFO is empty.</p>
2	R	0x1	<p>TFE TX FIFO Empty This is used to indicate that the TX FIFO is completely empty. 0: TX FIFO is not empty 1: TX FIFO is empty This bit is cleared when the TX FIFO is no longer empty.</p>
1	R	0x1	<p>TFNF TX FIFO Not Full This is used to indicate that the TX FIFO is not full. 0: TX FIFO is full 1: TX FIFO is not full This bit is cleared when the TX FIFO is full.</p>
0	R	0x0	<p>BUSY UART Busy Bit 0: Idle or inactive 1: Busy</p>

#### 8.2.6.14 0x0080 UART Transmit FIFO Level Register (Default Value: 0x0000\_0000)

Offset: 0x0080			Register Name: UART_TFL
Bit	Read/Write	Default/Hex	Description
31:9	/	/	/
8:0	R	0x0	<p>TFL TX FIFO Level The bit indicates the number of data entries in the TX FIFO.</p>

#### 8.2.6.15 0x0084 UART Receive FIFO Level Register (Default Value: 0x0000\_0000)

Offset: 0x0084			Register Name: UART_RFL
Bit	Read/Write	Default/Hex	Description
31:9	/	/	/
8:0	R	0x0	<p>RFL RX FIFO Level The bit indicates the number of data entries in the RX FIFO.</p> <p><b>Note:</b> Not use when the RXDMA master is enabled (UART_RXDMA_CTRL[0] is set to 1).</p>

#### 8.2.6.16 0x0088 UART DMA Handshake Configuration Register (Default Value: 0x0000\_00A5)

Offset: 0x0088			Register Name: UART_HSK
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R/W	0xA5	<p>Handshake configuration 0xA5: DMA wait cycle mode 0xE5: DMA handshake mode</p>

#### 8.2.6.17 0x008C UART DMA Request Enable Register(Default Value: 0x0000\_0003)

Offset: 0x008C			Register Name: UART_DMA_REQ_EN
Bit	Read/Write	Default/Hex	Description
31:3	/	/	/
2	R/W	0x0	<p>DMA Timeout Enable 0: Disable 1: Enable</p>
1	R/W	0x1	<p>DMA TX REQ Enable 0: Disable 1: Enable</p>
0	R/W	0x1	<p>DMA RX REQ Enable 0: Disable 1: Enable</p>

## 8.2.6.18 0x00A4 UART Halt TX Register (Default Value: 0x0000\_0000)

Offset: 0x00A4			Register Name: UART_HALT
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R/W	0x0	<p>PTE The sending of TX_REQ In DMA1 mode (FIFO on), if the PTE is set to 1 when the TFL in UART_TFL is less than or equal to the trigger value, the controller sends the DMA request. If the PTE is set to 0, when FIFO is empty, the controller sends the DMA request. The DMA request will stop when FIFO is full.</p> <p>In DMA0 mode, if the PTE is set to 1 and FIFO is on, when the TFL in UART_TFL is less than or equal to the trigger value, the controller sends DMA request. If the PTE is set to 1 and FIFO off, when the THR in UART_THR is empty, the controller sends DMA request. If the PTE is set to 0, when FIFO(FIFO Enable) or THR(FIFO Enable) is empty, the controller sends DMA request. Otherwise, the DMA request is cleared.</p>
6	R/W	0x0	<p>DMA_PTE_RX The Transmission of RX_DRQ In DMA1 mode, when RFL is more than or equal to the trigger value, or a receive timeout has occurred, the controller sends DRQ.</p> <p>In DMA0 mode, when DMA_PTE_RX = 1 and FIFO is on, if RFL is more than or equal to trig, the controller sends DRQ, else DRQ is cleared. In other cases, once the received data is valid, the controller sends DRQ.</p>
5	R/W	0x0	<p>SIR_RX_INVERT SIR RX Pulse Polarity Invert 0: Not invert receiver signal 1: Invert receiver signal</p>
4	R/W	0x0	<p>SIR_TX_INVERT SIR TX Pulse Polarity Invert 0: Not invert transmit pulse 1: Invert transmit pulse</p>
3	/	/	/

Offset: 0x00A4			Register Name: UART_HALT
Bit	Read/Write	Default/Hex	Description
2	R/WAC	0x0	<p>CHANGE_UPDATE</p> <p>After the user uses UART_HALT[1] to change the baud rate or LCR configuration, write 1 to update the configuration and wait this bit to self-clear to 0 to finish update process. Writing 0 to this bit has no effect.</p> <p>1: Update trigger, self-clear to 0 when finish update.</p>
1	R/W	0x0	<p>CHCFG_AT_BUSY</p> <p>This is an enable bit for the user to change LCR register configuration and baud rate register (UART_DLH and UART_DLL) when the UART is busy.</p> <p>1: Enable change when busy</p>
0	R/W	0x0	<p>HALT_TX</p> <p>Halt TX</p> <p>This register is used to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFOs are implemented and enabled.</p> <p>0 : Halt TX disabled</p> <p>1 : Halt TX enabled</p> <p><b>Note: If FIFOs are not enabled, the setting has no effect on operation.</b></p>

#### 8.2.6.19 0x00B0 UART DBG DLL Register (Default Value: 0x0000\_0000)

Offset: 0x00B0			Register Name: UART_DBG_DLL
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R	0x0	DEBUG DLL

#### 8.2.6.20 0x00B4 UART DBG DLH Register (Default Value: 0x0000\_0000)

Offset: 0x00B4			Register Name: UART_DBG_DLH
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R	0x0	DEBUG DLH

#### 8.2.6.21 0x00F0 UART FIFO Clock Control Register (Default Value: 0x0000\_0003)

Offset: 0x00F0			Register Name: UART_FCC
Bit	Read/Write	Default/Hex	Description
31:8	R	0x0	FIFO Depth Indicates the depth of TX/RX FIFO
7:3	/	/	/
2	R/W	0x0	RXFIFO Clock Mode 0: Sync mode, writing/reading clocks use apb clock 1: Sync mode, writing clock uses apb clock, reading clock uses ahb clock
1	R/W	0x1	TX FIFO Clock Enable 0: Clock disable 1: Clock enable
0	R/W	0x1	RX FIFO Clock Enable 0: Clock disable 1: Clock enable

#### 8.2.6.22 0x0100 UART RXDMA Control Register (Default Value: 0x0000\_0000)

Offset: 0x0100			Register Name: UART_RXDMA_CTRL
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23:8	R/W	0x0	RXDMA Timeout Threshold Unit is 1 UART bit time Note that this field is only configurable when RXDMA Busy is 0.
7	/	/	/
6	R/W	0x0	RXDMA Timeout Enable Once enable, the DMA starts a transfer even the data entries in RX FIFO do not reach BLK_SIZE. Note that this field is only configurable when RXDMA Busy is 0.

Offset: 0x0100			Register Name: UART_RXDMA_CTRL
Bit	Read/Write	Default/Hex	Description
5:4	R/W	0x0	<p>RXDMA AHB Burst Mode Set for AHB port burst supported INCR8 is recommended, while INCR16 may be unsupported due to the system bus.</p> <p>00: SIGNLE 01: INCR4 10: INCR8 11: INCR16</p> <p>Note that this field is only configurable while RXDMA Busy is 0.</p>
3:2	R/W	0x0	<p>RXDMA BLK SIZE Every time when data entries in RX FIFO reach BLK_SIZE, start a DMA block transfer. It is recommended that the block size no more than RX FIFO Depth.</p> <p>00: 8 bytes 01: 16 bytes 10: 32 bytes 11: 64 bytes</p> <p>Note that this field is only configurable while RXDMA Busy is 0.</p>
1	R/W	0x0	<p>RXDMA Mode 0: Continous 1: Limited When data transferred reaches the limited count set in RXDMA LIMIT, the DMA stops and the RXDMA Start bit is cleared automatically.</p> <p>Note that this field is only configurable while RXDMA Busy is 0.</p>
0	R/W	0x0	<p>RXDMA Enable 0: RXDMA Disable 1: RXDMA Enable Note that if the software turns off this bit, the RXDMA will stop after the current block transfer completes, then the software should do a reset to the RX FIFO before re-enable.</p>

#### 8.2.6.23 0x0104 UART RXDMA Start Register (Default Value: 0x0000\_0000)

Offset: 0x0104			Register Name: UART_RXDMA_STR
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
0	R/WAC	0x0	RXDMA Start Only valid when RXDMA mode is set to 1, it is auto cleared when data transferred reaches the RXDMA Limit Size.

#### 8.2.6.24 0x0108 UART RXDMA Status Register (Default Value: 0x0000\_0000)

Offset: 0x0108			Register Name: UART_RXDMA_STA
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
1	R	0x0	Buffer Read Address Updating 0: Buffer Read Address Register is ready for updating 1: Buffer Read Address Register is busy for updating The software should not update Buffer Read Address Register until this bit is 0.
0	R	0x0	RXDMA BUSY 0: RXDMA is idle 1: RXDMA is busy

#### 8.2.6.25 0x010C UART RXDMA Limit Register (Default Value: 0x0000\_0000)

Offset: 0x010C			Register Name: UART_RXDMA_LMT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0	RXDMA Limit Size Only valid when RXDMA Mode is set to 1, and the unit is byte. Note that this field is only configurable while RXDMA Busy is 0.

**8.2.6.26 0x0110 UART RXDMA Buffer Start Address Low Register (Default Value: 0x0000\_0000)**

Offset: 0x0110			Register Name: UART_RXDMA_SADDRL
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	RXDMA Buffer Start Address [31:0] Byte address Note that this field is only configurable while RXDMA Busy is 0.

**8.2.6.27 0x0114 UART RXDMA Buffer Start Address High Register (Default Value: 0x0000\_0000)**

Offset: 0x0114			Register Name: UART_RXDMA_SADDRH
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1:0	R/W	0x0	RXDMA Buffer Start Address [33:32] Byte address Note that this field is only configurable while RXDMA Busy is 0.

**8.2.6.28 0x0118 UART RXDMA Buffer Length Register (Default Value: 0x0000\_0000)**

Offset: 0x0118			Register Name: UART_RXDMA_BL
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0	RXDMA Buffer Length Unit is byte Note that this field is only configurable while RXDMA Busy is 0.

**8.2.6.29 0x0120 UART RXDMA Interrupt Enable Register (Default Value: 0x0000\_0000)**

Offset: 0x0120			Register Name: UART_RXDMA_IE
Bit	Read/Write	Default/Hex	Description
31:4	/	/	/
3	R/W	0x0	RXDMA Buffer Overrun RXDMA Buffer Overrun Interrupt Enable

Offset: 0x0120			Register Name: UART_RXDMA_IE
Bit	Read/Write	Default/Hex	Description
2	R/W	0x0	RXDMA Timeout Done RXDMA Timeout Done Interrupt Enable
1	R/W	0x0	RXDMA BLK Done RXDMA BLK Done Interrupt Enable
0	R/W	0x0	RXDMA Limit Done RXDMA Limit Done Interrupt Enable

#### 8.2.6.30 0x0124 UART RXDMA Interrupt Status Register (Default Value: 0x0000\_0000)

Offset: 0x0124			Register Name: UART_RXDMA_IS
Bit	Read/Write	Default/Hex	Description
31:4	/	/	/
3	R/W1C	0x0	RXDMA Buffer Overrun Asserted when the RXDMA buffer is overflow.
2	R/W1C	0x0	RXDMA Timeout Done Asserted when a DMA transfer caused by timeout is done.
1	R/W1C	0x0	RXDMA BLK Done Asserted when a DMA block transfer is done.
0	R/W1C	0x0	RXDMA Limit Done Asserted when data transferred reaches limit size in RXDMA Limit Mode.

#### 8.2.6.31 0x0128 UART RXDMA Write Address Low Register (Default Value: 0x0000\_0000)

Offset: 0x0128			Register Name: UART_RXDMA_WADDR
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	RXDMA Current Write Address[31:0] Updated when every DMA transfer is done It is byte address.

**8.2.6.32 0x012C UART RXDMA Write Address High Register (Default Value: 0x0000\_0000)**

Offset: 0x012C			Register Name: UART_RXDMA_WADDRH
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1:0	R	0x0	RXDMA Current Write Address[33:32] Updated when every DMA transfer is done It is byte address.

**8.2.6.33 0x0130 UART RXDMA Read Address Low Register (Default Value: 0x0000\_0000)**

Offset: 0x0130			Register Name: UART_RXDMA_RADDRL
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	RXDMA Current Read Address[31:0] Software should update this register after reading data in RXDMA Buffer in time It is byte address. The software should not update Buffer Read Address Register until UART_RXDMA_STA[1] is 0. The software should update Read Address High Register first, and then Read Address Low Register, even there is no change on Read Address High Register.

**8.2.6.34 0x0134 UART RXDMA Read Address High Register (Default Value: 0x0000\_0000)**

Offset: 0x0134			Register Name: UART_RXDMA_RADDRH
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1:0	R/W	0x0	RXDMA Current Read Address[33:32] Software should update this register after reading data in RXDMA Buffer in time. It is byte address. The software should not update Buffer Read Address Register until until UART_RXDMA_STA[1] is 0. The software should update Read Address High Register first and then Read Address Low Register , even there is no change on Read Address High Register.

**8.2.6.35 0x0138 UART RXDMA Data Count Register (Default Value: 0x0000\_0000)**

Offset: 0x0138			Register Name: UART_RXDMA_DCNT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R	0x0	RXDMA Data Count Only valid while RXDMA Mode is set to 1, it is used for counting the data transferred by RXDMA, and is cleared when reaches RXDMA Limit Size. Its unit is byte.

## 8.3 SPI

### 8.3.1 Overview

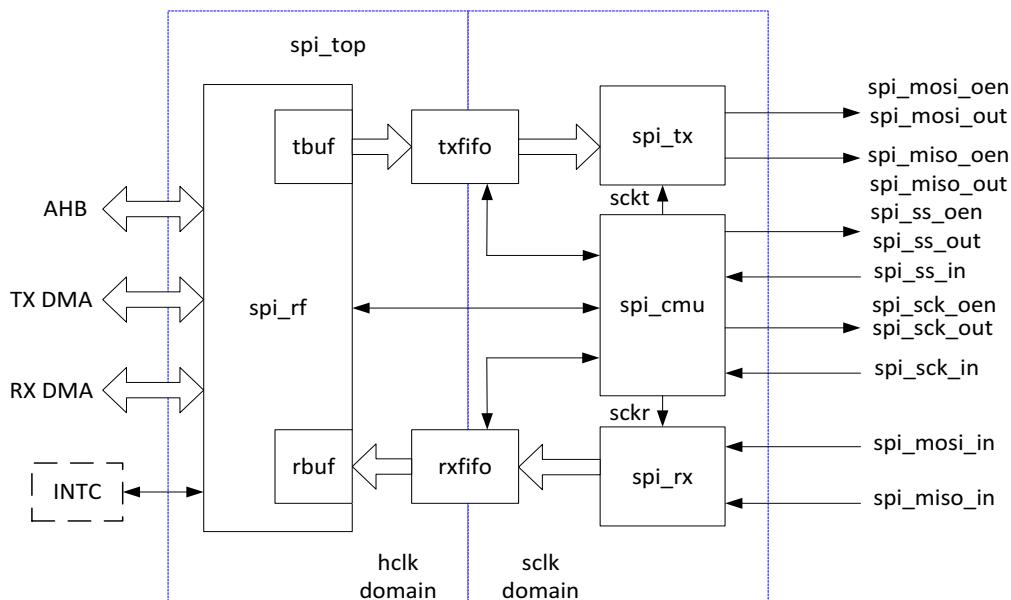
The Serial Peripheral Interface (SPI) is a full-duplex, synchronous, four-wire serial communication interface between a CPU and SPI-compliant external devices. The SPI controller contains a 64 x 8 bits receiver buffer (RXFIFO) and a 64 x 8 bits transmit buffer (TXFIFO). It can work in master mode and slave mode.

The SPI has the following features:

- Full-duplex synchronous serial interface
- Master/slave configurable
- 8-bit wide by 64-entry FIFO for both transmit and receive data
- Polarity and phase of the Chip Select (SPI\_SS) and SPI Clock (SPI\_SCLK) are configurable
- Supports interrupts and DMA
- Supports mode0, mode1, mode2, and mode3
- Supports 3-wire/4-wire SPI
- Supports programmable serial data frame length: 1 bit to 32 bits
- Supports standard SPI, dual-output/dual-input SPI, dual I/O SPI, quad-output/quad-input SPI
- Supports maximum IO rate of the mass production: 100 MHz

### 8.3.2 Block Diagram

Figure 8-20 shows a block diagram of the SPI.

**Figure 8-20 SPI Block Diagram**

SPI contains the following sub-blocks:

**Table 8-6 SPI Sub-blocks**

Sub-block	Description
spi_rf	Responsible for implementing the internal register, interrupt, and DMA Request.
spi_tbuf	The data length transmitted from AHB to TXFIFO is converted into 8 bits, then the data is written into the RXFIFO.
spi_rbuf	The block is used to convert the RXFIFO data into the reading data length of AHB.
txfifo, rxfifo	The data transmitted from the SPI to the external serial device is written into the TXFIFO; the data received from the external serial device into SPI is pushed into the RXFIFO.
spi_cmu	Responsible for implementing SPI bus clock, chip select, internal sample, and the generation of transfer clock.
spi_tx	Responsible for implementing SPI data transfer, the interface of the internal TXFIFO, and status register.
spi_rx	Responsible for implementing SPI data receive, the interface of the internal RXFIFO, and status register.

### 8.3.3 Functional Description

#### 8.3.3.1 External Signals

The following table describes the external signals of SPI. The MOSI and MISO are bidirectional I/O, when SPI is as a master device, the CLK and CS are the output pin; when SPI is as a slave device, the CLK and CS are the input pin. When using SPI, the corresponding PADs are selected as SPI function via section 8.7 “[GPIO](#)”.

**Table 8-7 SPI External Signals**

Signal	Description	Type
SPI0-CS	SPI0 chip select signal, low active When the device is not selected, data will not be accepted via the SI pin, and the SO pin will stop transmission.	I/O
SPI0-CLK	SPI0 clock signal This pin is used to provide a clock to the device and is used to control the flow of data to and from the device.	I/O
SPI0-MOSI	SPI0 master data out, slave data in	I/O
SPI0-MISO	SPI0 master data in, slave data out	I/O
SPI0-WP	Write protection and low active It also can be used for serial data input and output for SPI Quad Input or Quad Output mode.	I/O
SPI0-HOLD	When the device is selected and a serial sequence is underway, the HOLD pin can be used to temporarily pause the serial communication with the master device without deselecting or resetting the serial sequence. While the HOLD pin is asserted, the SO pin is at high impedance, and all transitions on the SCK pin and data on the SI pin are ignored. It also can be used for serial data input and output for SPI Quad Input or Quad Output mode.	I/O

#### 8.3.3.2 Clock Sources

The SPI controller gets 5 different clock sources, users can select one of them to make SPI clock source. The following table describes the clock sources for SPI. For more details on the clock setting, configuration, and gating information, see section 3.3 “[CCU](#)”.

**Table 8-8 SPI Clock Sources**

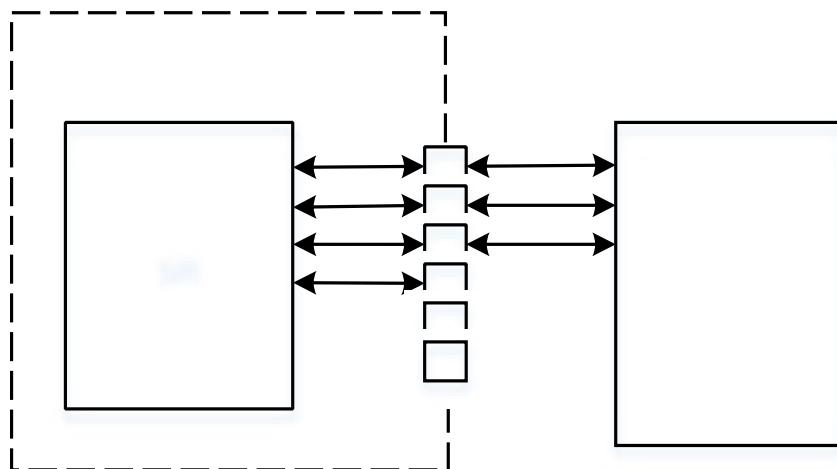
Clock Sources	Description
HOSC	24 MHz Crystal

Clock Sources	Description
PLL_PERI(1X)	Peripheral Clock, default value is 600 MHz
PLL_PERI(2X)	Peripheral Clock, default value is 1200 MHz
PLL_AUDIO0(DIV2)	Audio Clock, the default value is 1536 MHz
PLL_AUDIO0(DIV5)	Audio Clock, the default value is 614.4 MHz

### 8.3.3.3 Typical Application

Figure 8-21 shows the application block diagram when the SPI master device is connected to a slave device.

**Figure 8-21 SPI Application Block Diagram**



### 8.3.3.4 SPI Transmit Format

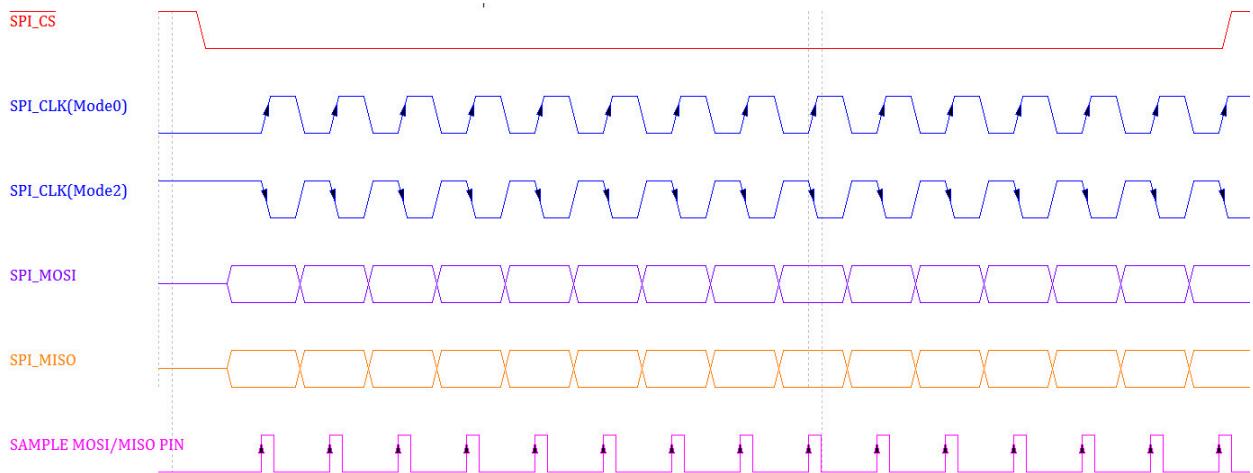
The SPI supports 4 different formats for data transfer. The software can select one of the four modes in which the SPI works by setting the bit1 (Polarity) and bit0 (Phase) of [SPI\\_TCR](#). The SPI controller master uses the SPI\_SCLK signal to transfer data in and out of the shift register. Data is clocked using any one of four programmable clock phase and polarity combinations.

The CPOL ([SPI\\_TCR\[1\]](#)) defines the polarity of the clock signal (SPI\_SCLK). The SPI\_SCLK is a high level when CPOL is '1' and it is a low level when CPOL is '0'. The CPHA ([SPI\\_TCR\[0\]](#)) decides whether the leading edge of SPI\_SCLK is used to setup or sample data. The leading edge is used to setup data when CPHA is '1', and sample data when CPHA is '0'. The following table lists the four modes.

**Table 8-9 SPI Transmit Format**

Mode	Polarity (CPOL)	Phase (CPHA)	Leading Edge	Trailing Edge
Mode0	0	0	Sample on the rising edge	Setup on the falling edge
Mode1	0	1	Setup on the rising edge	Sample on the falling edge
Mode2	1	0	Sample on the falling edge	Setup on the rising edge
Mode3	1	1	Setup on the falling edge	Sample on the rising edge

Figure 8-22 and Figure 8-23 describe four waveforms for SPI\_SCLK.

**Figure 8-22 SPI Phase 0 Timing Diagram****Figure 8-23 SPI Phase 1 Timing Diagram**

### 8.3.3.5 SPI Master and Slave Mode

The SPI controller can be configured to a master or slave device. The master mode is selected by setting the MODE bit ([SPI\\_GCR\[1\]](#)); the slave mode is selected by clearing the MODE bit.

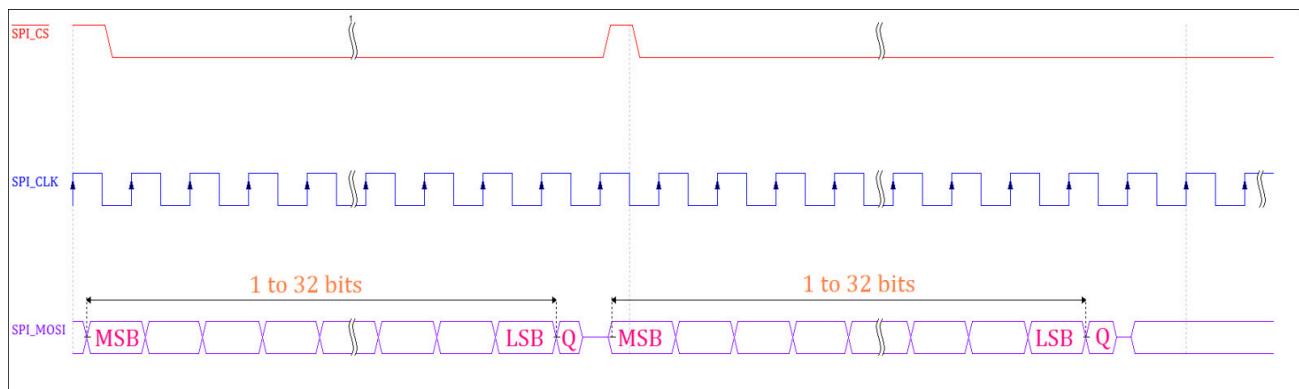
In master mode, the SPI\_CLK is generated and transmitted to the external device, and the data from the TX FIFO is transmitted on the MOSI pin, the data from the slave device is received on the MISO pin and sent to RX FIFO. The Chip Select (SPI\_SS) is an active low signal, and it must be set low before the data are transmitted or received. The SPI\_SS can be selected the auto control mode or software manual control mode. When using the auto control, the SS\_OWNER ([SPI\\_TCR\[6\]](#)) must be cleared (default value is 0); when using the manual control, the SS\_OWNER must be set. And the level of SPI\_SS is controlled by SS\_LEVEL ([SPI\\_TCR\[7\]](#)).

In slave mode, after the software selects the MODE bit ([SPI\\_GCR\[1\]](#)) to '0', it waits for master initiate a transaction. When the master asserts SPI\_SS, then SPI\_CLK is transmitted to the slave device, the slave data is transmitted from TX FIFO on the MISO pin and the data from the MOSI pin is received in RX FIFO.

### 8.3.3.6 SPI 3-Wire Mode

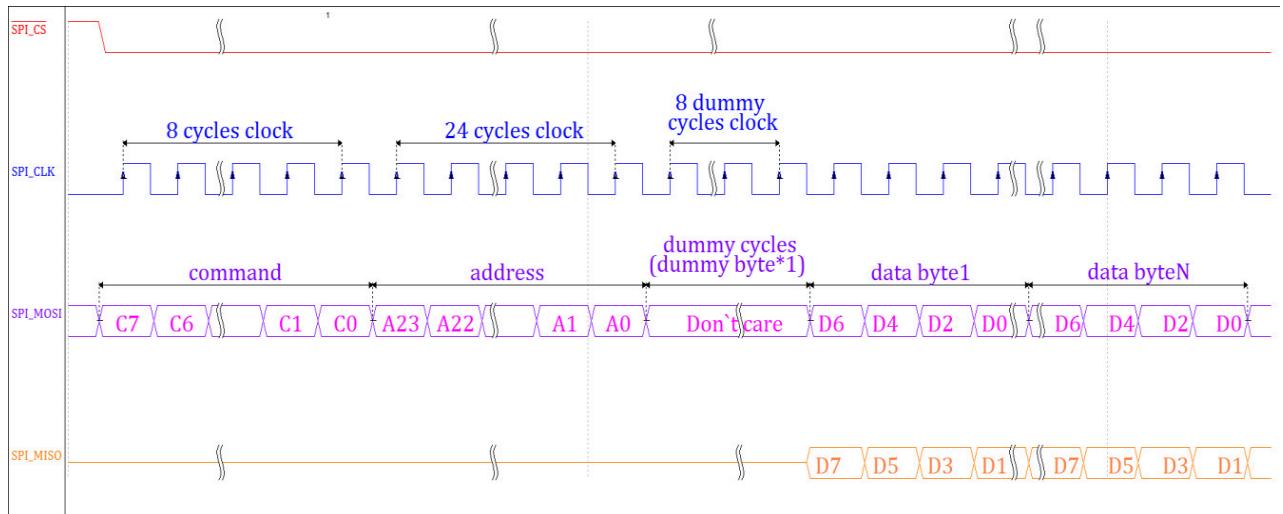
The SPI 3-wire mode is only valid when the SPI controller work in master mode, and is selected when the Work Mode Select bit ([SPI\\_BATC\[1:0\]](#)) is equal to 0x2. And in the 3-wire mode, the input data and the output data use the same single data line. The following figure describes the 3-wire mode.

**Figure 8-24 SPI 3-Wire Mode**

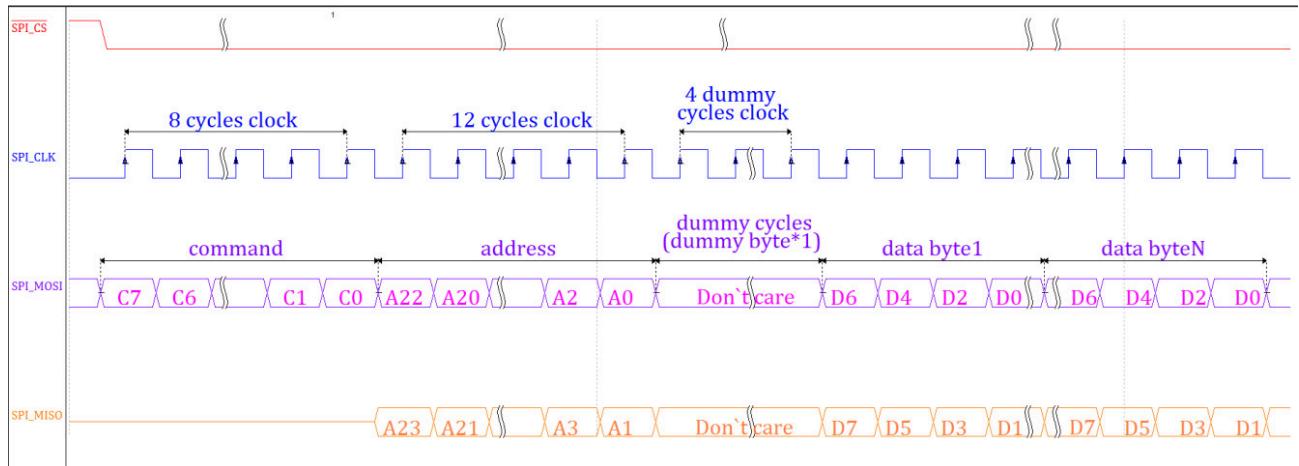


### 8.3.3.7 SPI Dual-Input/Dual-Output and Dual I/O Mode

The dual read mode (SPI x2) is selected when the DRM is set in [SPI\\_BCC\[28\]](#). Using the dual mode allows data to be transferred to or from the device at double the rate of standard single mode, the data can be read at fast speed using two data bits (MOSI and MISO) at a time. The following figure describes the dual-input/dual-output SPI (Figure 8-25) and the dual I/O SPI (Figure 8-26).

**Figure 8-25 SPI Dual-Input/Dual-Output Mode**

In the dual-input/dual-output SPI mode, the command, address, and the dummy bytes output in a unit of a single bit in serial mode through the SPI\_MOSI line, only the data bytes are output (write) and input (read) in a unit of dual bits through the SPI\_MOSI and SPI\_MISO.

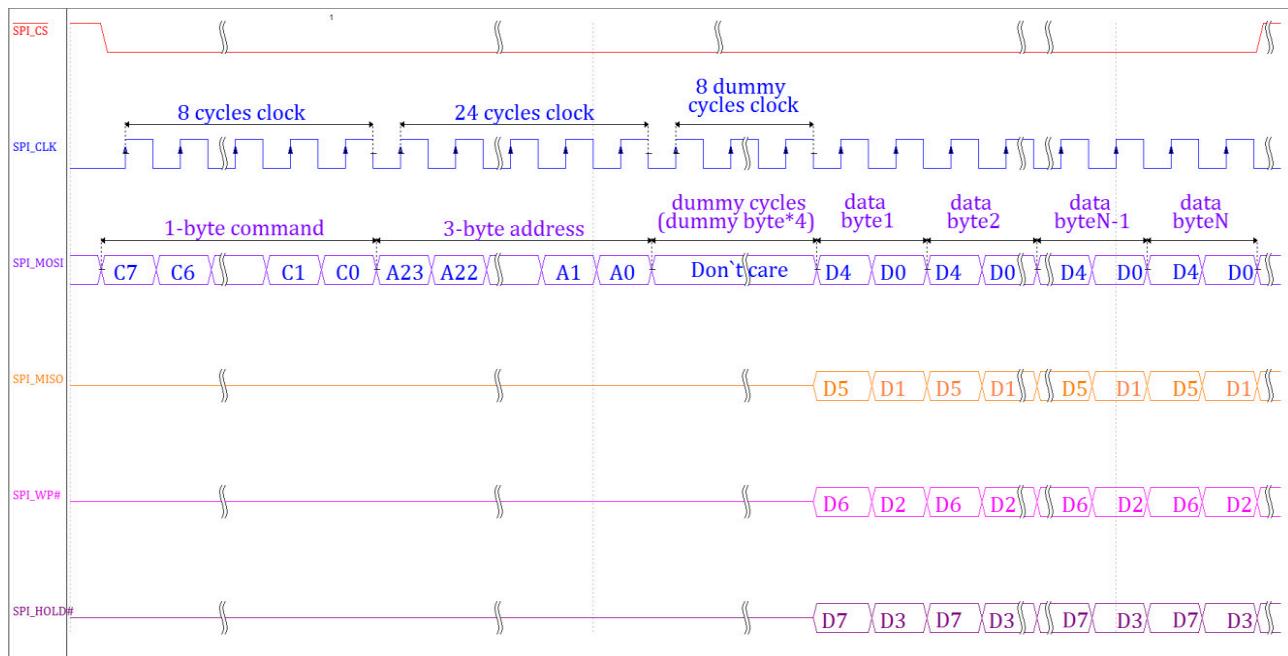
**Figure 8-26 SPI Dual I/O Mode**

In the dual I/O SPI mode, only the command bytes output in a unit of a single bit in serial mode through the SPI\_MOSI line. The address bytes and the dummy bytes output in a unit of dual bits through the SPI\_MOSI and SPI\_MISO. And the data bytes output (write) and input (read) in a unit of dual bits through the SPI\_MOSI and SPI\_MISO.

### 8.3.3.8 SPI Quad-Input/Quad-Output Mode

The quad read mode (SPI x4) is selected when the Quad\_EN is set in [SPI\\_BCC\[29\]](#). Using the quad mode allows data to be transferred to or from the device at 4 times the rate of standard single mode, the data can be read at fast speed using four data bits (MOSI, MISO, IO2 (WP#) and IO3 (HOLD#)) at the same time. The following figure describes the quad-input/quad-output SPI.

**Figure 8-27 SPI Quad-Input/Quad-Output Mode**



In the quad-input/quad-output SPI mode, the command, address, and the dummy bytes output in a unit of a single bit in serial mode through SPI\_MOSI line. Only the data bytes output (write) and input (read) in a unit of quad bits through the SPI\_MOSI, SPI\_MISO, SPI\_WP#, and SPI\_HOLD#.

### 8.3.3.9 Transmission/Reception Bursts in Master Mode

In SPI master mode, the transmission and reception bursts (byte in unit) are configured before the SPI transfers the serial data between the processor and external device. The transmission bursts are written in MWTC (bit[23:0]) of the [SPI Master Transmit Counter Register](#). The transmission bursts in single mode before automatically sending dummy bursts are written in STC (bit[23:0]) of the [SPI Master Transmit Counter Register](#). For dummy data, the SPI controller can automatically send before receiving by writing DBC (bit[27:24]) in the [SPI Master Transmit Counter Register](#). If users do not use the SPI controller to send dummy data automatically, then the dummy bursts are used as the transmission counters to write together in MWTC (bit[23:0]) of the [SPI Master Transmit Counter Register](#). In master mode, the total burst numbers are written in MBC (bit[23:0]) of the [SPI Master Transmit Counter Register](#). When all transmission and reception bursts are transferred, the SPI controller will send a completed interrupt, at the same time, the SPI controller will clear [DBC](#), [MWTC](#), and [MBC](#).

### 8.3.3.10 SPI Sample Mode and Run Clock Configuration

The SPI controller runs at 3 kHz–100 MHz at its interface to external SPI devices. The internal SPI clock should run at the same frequency as the outgoing clock in the master mode. The SPI clock is selected from different clock sources, the SPI must configure different work mode. There are three work modes: normal sample mode, delay half-cycle sample mode, delay one-cycle sample mode. The Delay half-cycle sample mode is the default mode of the SPI controller. When the SPI runs at 40 MHz or below 40 MHz, the SPI can work at normal sample mode or delay half-cycle sample mode. When the SPI runs over 80 MHz, setting the SDC bit in the [SPI Transfer Control Register](#) to '1' makes the internal read sample point with a half-cycle delay of SPI\_CLK, which is used in high speed read operation to reduce the error caused by the time delay of SPI\_CLK between master and slave. Table 8-10 and Table 8-11 show the different configurations of the SPI sample mode.

**Table 8-10 SPI Old Sample Mode and Run Clock**

SPI Sample Mode	SDM(bit13)	SDC(bit11)	Run Clock
normal sample	1	0	<=24 MHz
delay half cycle sample	0	0	<=40 MHz
delay one cycle sample	0	1	>=80 MHz



**CAUTION**

The remaining spectrum is not recommended. Because when the output delay of SPI flash (refer to the datasheet of the manufacturer for the specific delay time) is the same with the half-cycle time of SPI working clock, the variable edge of the output data for the device bumps into the clock sampling edge of the controller, so setting 1 cycle of sampling delay would cause stability problem.

**Table 8-11 SPI New Sample Mode**

SPI Sample Mode	SDM (bit13)	SDC (bit11)	SDC1 (bit15)
normal sample	1	0	0
delay half cycle sample	0	0	0
delay one cycle sample	0	1	0
delay 1.5 cycle sample	1	1	0
delay 2 cycle sample	1	0	1
delay 2.5 cycle sample	0	0	1
delay 3 cycle sample	0	1	1

### 8.3.3.11 SPI Error Conditions

If any error conditions occur, the hardware will set the corresponding status bits in the [SPI Interrupt Status Register](#) and stop the transfer. For the SPI controller, the following error scenarios can happen.

#### 1. TX\_FIFO Underrun

The TX\_FIFO underrun happens when the CPU/DMA reads data from TX FIFO when it is empty. In the case, the SPI controller will end the transaction and flag the error bit along with the TF\_UDF bit in the [SPI Interrupt Status Register](#). The SPI controller will generate an interrupt if interrupts are enabled. The software has to clear the error bit and the TF\_UDF bit. To start a new transaction, the software has to reset the FIFO by writing to the SRST (soft reset) bit in the [SPI Global Control Register](#).

#### 2. TX\_FIFO Overflow

The TX\_FIFO overflow happens when the CPU/DMA writes data into the TX FIFO when it is full. In the case, the SPI controller will end the transaction and flag the error bit along with the TF\_OVF bit in the [SPI Interrupt Status Register](#). The SPI controller will generate an interrupt if interrupts are enabled. The software has to clear the error bit and the TF\_OVF bit. To start a new transaction, the software has to reset the FIFO by writing to the SRST (soft reset) bit in the [SPI Global Control Register](#).

#### 3. RX\_FIFO Underrun

The RX\_FIFO underrun happens when the CPU/DMA reads data from RX FIFO when it is empty. In the case, the SPI controller will end the transaction and flag the error bit along with the RF\_UDF bit in the [SPI Interrupt Status Register](#). The SPI controller will generate an interrupt if interrupts are enabled. The software has to clear the error bit and the RF\_UDF bit. To start a new transaction, the software has to reset the fifo by writing to the SRST (soft reset) bit in the [SPI Global Control Register](#).

#### 4. RX\_FIFO Overflow

The RX\_FIFO overflow happens when the CPU/DMA writes data into the RX FIFO when it is full. In the case, the SPI controller will end the transaction and flag the error bit along with the RF\_OVF bit in the [SPI Interrupt Status Register](#). The SPI controller will generate an interrupt if interrupts are enabled. The software has to clear the error bit and the RF\_OVF bit. To start a new transaction, the software has to reset the FIFO by writing to the SRST (soft reset) bit in the [SPI Global Control Register](#).

### 8.3.4 Programming Guidelines

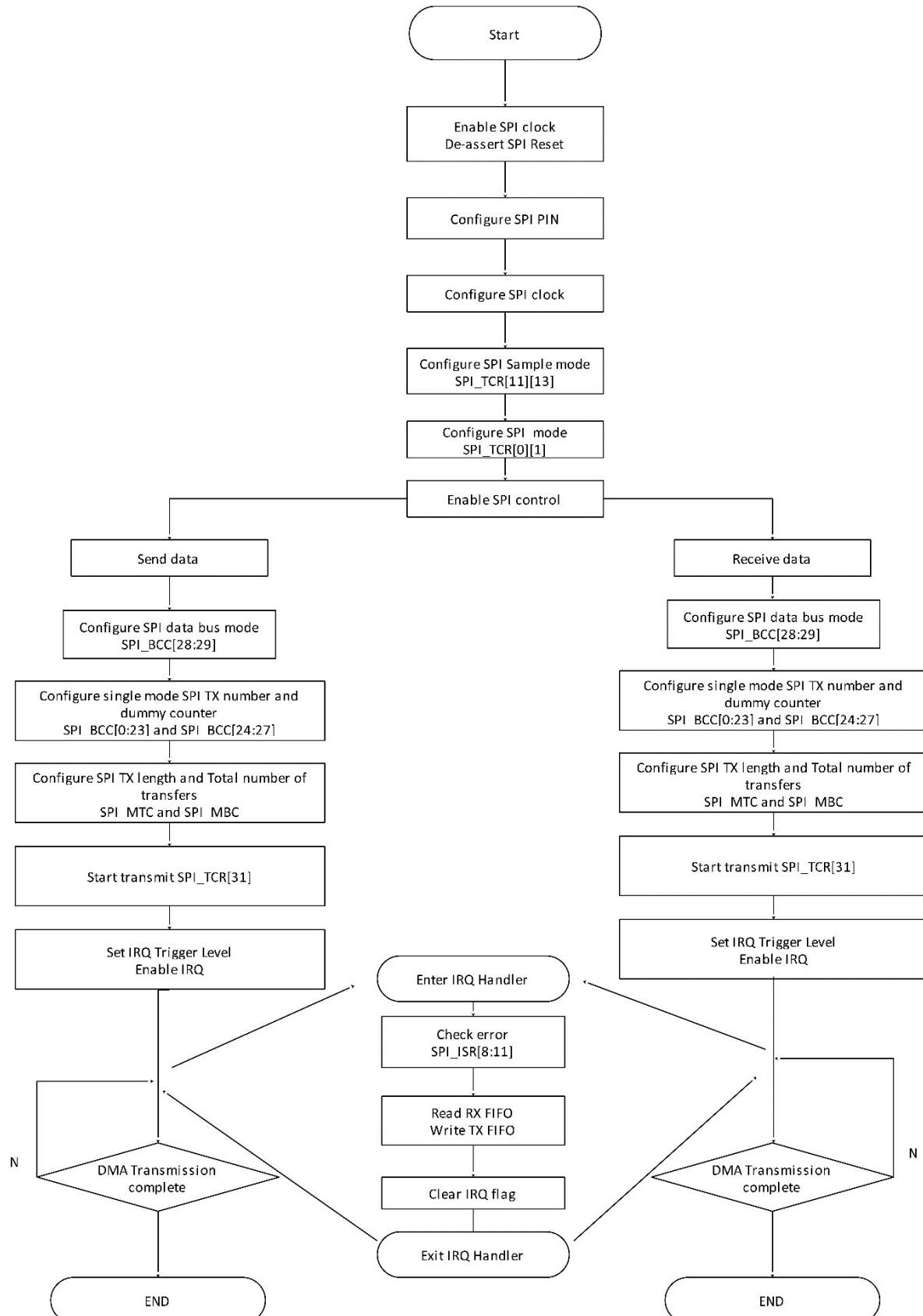
#### 8.3.4.1 Writing/Reading Data Process

The SPI transfers serial data between the processor and the external device. The CPU mode and DMA mode are the two main operational modes for SPI. For each SPI, the data is simultaneously transmitted (shifted out serially) and received (shifted in serially). The SPI has 2 channels, including the TX channel and RX channel. The TX channel has the path from TX FIFO to the external device. The RX channel has the path from the external device to RX FIFO.

**Write Data:** The CPU or DMA must write data on the [SPI\\_TXD](#) register, the data on the register are automatically moved to TX FIFO.

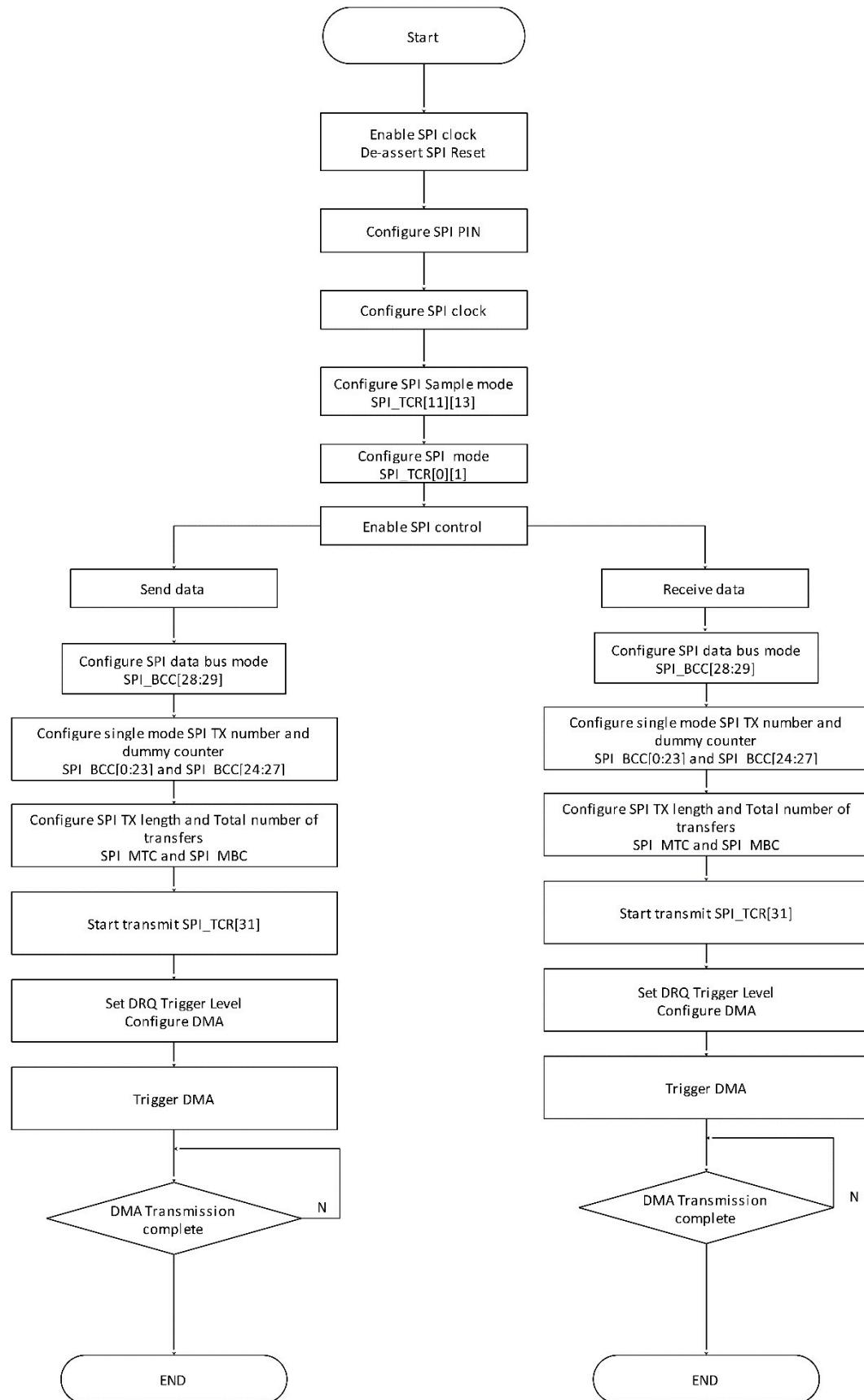
**Read Data:** To read data from RX FIFO, the CPU or DMA must access the [SPI\\_RXD](#) register and the data are automatically sent to the [SPI\\_RXD](#) register.

In CPU or DMA mode, the SPI sends a completed interrupt ([SPI\\_ISR\[TC\]](#)) to the processor after each transmission is complete.

**CPU Mode****Figure 8-28 SPI Write/Read Data in CPU Mode**

## DMA Mode

Figure 8-29 SPI Write/Read Data in DMA Mode



### 8.3.4.2 Calibrate Delay Chain

The SPI has one delay chain which is used to generate delay to make proper timing between the internal SPI clock signal and data signals. Delay chain is made up of 64 delay cells. The delay time of one delay cell can be estimated through delay chain calibration.

The steps to calibrate delay chain are as follows:

- Step 1** Enable SPI. To calibrate the delay chain by the operation registers in SPI, the SPI must be enabled through AHB reset and AHB clock gating control registers.
- Step 2** Configure a proper clock for SPI. The calibration delay chain is based on the clock for SPI from CCU.
- Step 3** Set proper initial delay value. Write 0xA0 to the [SPI Sample Delay Control Register](#) to set initial delay value 0x20 to delay chain. Then write 0x0 to the [SPI Sample Delay Control Register](#) to clear this value.
- Step 4** Write 0x8000 to the [SPI Sample Delay Control Register](#) to start to calibrate the delay chain.
- Step 5** Wait until the flag (bit14 in the [SPI Sample Delay Control Register](#)) of calibration done is set. The number of delay cells is shown at the bit[13:8] of the [SPI Sample Delay Control Register](#). The delay time generated by these delay cells is equal to the cycle of the SPI clock nearly. This value is the result of calibration.
- Step 6** Calculate the delay time of one delay cell according to the cycle of the SPI clock and the result of calibration.

### 8.3.5 Register List

Module Name	Base Address
SPI0	0x04025000

Register Name	Offset	Description
SPI_GCR	0x0004	SPI Global Control Register
SPI_TCR	0x0008	SPI Transfer Control Register
SPI_IER	0x0010	SPI Interrupt Control Register
SPI_ISR	0x0014	SPI Interrupt Status Register
SPI_FCR	0x0018	SPI FIFO Control Register
SPI_FSR	0x001C	SPI FIFO Status Register
SPI_WCR	0x0020	SPI Wait Clock Register
SPI_SAMP_DL	0x0028	SPI Sample Delay Control Register
SPI_MBC	0x0030	SPI Master Burst Counter Register

Register Name	Offset	Description
SPI_MTC	0x0034	SPI Master Transmit Counter Register
SPI_BCC	0x0038	SPI Master Burst Control Register
SPI_BATCR	0x0040	SPI Bit-Aligned Transfer Configure Register
SPI_BA_CCR	0x0044	SPI Bit-Aligned Clock Configuration Register
SPI_TBR	0x0048	SPI TX Bit Register
SPI_RBR	0x004C	SPI RX Bit Register
SPI_NDMA_MODE_CTL	0x0088	SPI Normal DMA Mode Control Register
SPI_TXD	0x0200	SPI TX Data Register
SPI_RXD	0x0300	SPI RX Data Register

### 8.3.6 Register Description

#### 8.3.6.1 0x0004 SPI Global Control Register (Default Value: 0x0000\_0080)

Offset:0x0004			Register Name: SPI_GCR
Bit	Read/Write	Default/Hex	Description
31	R/WAC	0x0	<p>SRST Soft reset Writing ‘1’ to this bit will clear the SPI controller, and auto clear to ‘0’ when reset operation completes. Writing ‘0’ to this bit has no effect.</p>
30:8	/	/	/
7	R/W	0x1	<p>TP_EN Transmit Pause Enable In master mode, it is used to control the transmit state machine to stop smart burst sending when RX FIFO is full. 0: Normal operation, ignore RXFIFO status 1: Stop transmit data when RXFIFO full Cannot be written when XCH=1.</p>
6:3	/	/	/
2	R/W	0x0	<p>MODE_SELEC Sample Timing Mode Select 0: Old mode of Sample Timing 1: New mode of Sample Timing Cannot be written when XCH=1.</p>

Offset:0x0004			Register Name: SPI_GCR
Bit	Read/Write	Default/Hex	Description
1	R/W	0x0	<p>MODE SPI Function Mode Select 0: Slave mode 1: Master mode Cannot be written when XCH=1.</p>
0	R/W	0x0	<p>EN SPI Module Enable Control 0: Disable 1: Enable After transforming from bit_mode to byte_mode, it must enable the SPI module again.</p>

### 8.3.6.2 0x0008 SPI Transfer Control Register (Default Value: 0x0000\_0087)

Offset: 0x0008			Register Name: SPI_TCR
Bit	Read/Write	Default/Hex	Description
31	R/WAC	0x0	<p>XCH Exchange Burst In master mode, it is used to start SPI burst. 0: Idle 1: Initiates exchange. Writing “1” to this bit will start the SPI burst, and will auto-clear after finishing the bursts transfer specified by SPI_MBC. Writing “1” to SRST (SPI_GCR[31]) will also clear this bit. Writing ‘0’ to this bit has no effect. Cannot be written when XCH=1.</p>
30:16	/	/	/
15	R/W	0x0	<p>SDC1 Master Sample Data Control register1 Set this bit to ‘1’ to make the internal read sample point with a delay of half-cycle of SPI_CLK. It is used in high speed read operation to reduce the error caused by the time delay of SPI_CLK between master and slave. 0: normal operation, do not delay the internal read sample point 1: delay the internal read sample point Cannot be written when XCH=1.</p>

Offset: 0x0008			Register Name: SPI_TCR
Bit	Read/Write	Default/Hex	Description
14	R/W	0x0	<p>SDDM Sending Data Delay Mode 0: Normal sending 1: Delay sending Set the bit to "1" to make the data that should be sent with a delay of half-cycle for SPI_CLK in dual IO mode of SPI mode0. Cannot be written when XCH=1.</p>
13	R/W	0x0	<p>SDM Master Sample Data Mode 0: Delay sample mode 1: Normal sample mode In normal sample mode, the SPI master samples the data at the correct edge for each SPI mode; In delay sample mode, the SPI master samples data at the edge that is half cycle delayed by the correct edge defined in respective SPI mode. Cannot be written when XCH=1.</p>
12	R/W	0x0	<p>FBS First Transmit Bit Select 0: MSB first 1: LSB first Cannot be written when XCH=1.</p>
11	R/W	0x0	<p>SDC Master Sample Data Control Set this bit to '1' to make the internal read sample point with a delay of half-cycle for SPI_CLK. It is used in high speed read operation to reduce the error caused by the time delay of SPI_CLK between master and slave. 0: Normal operation, do not delay the internal read sample point 1: Delay the internal read sample point Cannot be written when XCH=1.</p>
10	R/W	0x0	<p>RPSM Rapids Mode Select Select rapid mode for high speed write. 0: Normal write mode 1: Rapid write mode Cannot be written when XCH=1.</p>

Offset: 0x0008			Register Name: SPI_TCR
Bit	Read/Write	Default/Hex	Description
9	R/W	0x0	<p>DDB Dummy Burst Type 0: The bit value of dummy SPI burst is zero 1: The bit value of dummy SPI burst is one Cannot be written when XCH=1.</p>
8	R/W	0x0	<p>DHB Discard Hash Burst In master mode, it controls whether discarding unused SPI bursts 0: Receiving all SPI bursts in the BC period 1: Discard unused SPI bursts, only fetching the SPI bursts during the dummy burst period. The burst number is specified by TC. Cannot be written when XCH=1.</p>
7	R/W	0x1	<p>SS_LEVEL When control SS signal manually (SS_OWNER (SPI_TCR[6]) == 1), set this bit to '1' or '0' to control the level of SS signal. 0: Set SS to low 1: Set SS to high Cannot be written when XCH=1.</p>
6	R/W	0x0	<p>SS_OWNER SS Output Owner Select Usually, the controller sends the SS signal automatically with data together. When this bit is set to 1, the software must manually write SS_LEVEL (SPI_TCR[7]) to 1 or 0 to control the level of the SS signal. 0: SPI controller 1: Software Cannot be written when XCH=1.</p>
5:4	R/W	0x0	<p>SS_SEL SPI Chip Select Select one of four external SPI Master/Slave Devices 00: SPI_SS0 will be asserted 01: SPI_SS1 will be asserted 10: SPI_SS2 will be asserted 11: SPI_SS3 will be asserted Cannot be written when XCH=1.</p>

Offset: 0x0008			Register Name: SPI_TCR
Bit	Read/Write	Default/Hex	Description
3	R/W	0x0	<p>SSCTL</p> <p>In master mode, this bit selects the output waveform for the SPI_SSx signal. Only valid when SS_OWNER (SPI_TCR[6])= 0.</p> <p>0: SPI_SSx remains asserted between SPI bursts</p> <p>1: Negate SPI_SSx between SPI bursts</p> <p>Cannot be written when XCH=1.</p>
2	R/W	0x1	<p>SPOL</p> <p>SPI Chip Select Signal Polarity Control</p> <p>0: Active high polarity (0 = Idle)</p> <p>1: Active low polarity (1 = Idle)</p> <p>Cannot be written when XCH=1.</p>
1	R/W	0x1	<p>CPOL</p> <p>SPI Clock Polarity Control</p> <p>0: Active high polarity (0 = Idle)</p> <p>1: Active low polarity (1 = Idle)</p> <p>Cannot be written when XCH=1.</p>
0	R/W	0x1	<p>CPHA</p> <p>SPI Clock/Data Phase Control</p> <p>0: Phase 0 (Leading edge for sample data)</p> <p>1: Phase 1 (Leading edge for setup data)</p> <p>Cannot be written when XCH=1.</p>

### 8.3.6.3 0x0010 SPI Interrupt Control Register (Default Value: 0x0000\_0000)

Offset: 0x0010			Register Name: SPI_IER
Bit	Read/Write	Default/Hex	Description
31:14	/	/	/
13	R/W	0x0	<p>SS_INT_EN</p> <p>SSI Interrupt Enable</p> <p>Chip select signal (SSx) from the valid state to the invalid state</p> <p>0: Disable</p> <p>1: Enable</p>

Offset: 0x0010			Register Name: SPI_IER
Bit	Read/Write	Default/Hex	Description
12	R/W	0x0	<p>TC_INT_EN Transfer Completed Interrupt Enable 0: Disable 1: Enable</p>
11	R/W	0x0	<p>TF_UDR_INT_EN TXFIFO Underrun Interrupt Enable 0: Disable 1: Enable</p>
10	R/W	0x0	<p>TF_OVF_INT_EN TX FIFO Overflow Interrupt Enable 0: Disable 1: Enable</p>
9	R/W	0x0	<p>RF_UDR_INT_EN RXFIFO Underrun Interrupt Enable 0: Disable 1: Enable</p>
8	R/W	0x0	<p>RF_OVF_INT_EN RX FIFO Overflow Interrupt Enable 0: Disable 1: Enable</p>
7	/	/	/
6	R/W	0x0	<p>TF_FUL_INT_EN TX FIFO Full Interrupt Enable 0: Disable 1: Enable</p>
5	R/W	0x0	<p>TX_EMP_INT_EN TX FIFO Empty Interrupt Enable 0: Disable 1: Enable</p>
4	R/W	0x0	<p>TX_ERQ_INT_EN TX FIFO Empty Request Interrupt Enable 0: Disable 1: Enable</p>
3	/	/	/

Offset: 0x0010			Register Name: SPI_IER
Bit	Read/Write	Default/Hex	Description
2	R/W	0x0	RF_FUL_INT_EN RX FIFO Full Interrupt Enable 0: Disable 1: Enable
1	R/W	0x0	RX_EMP_INT_EN RX FIFO Empty Interrupt Enable 0: Disable 1: Enable
0	R/W	0x0	RF_RDY_INT_EN RX FIFO Ready Request Interrupt Enable 0: Disable 1: Enable

#### 8.3.6.4 0x0014 SPI Interrupt Status Register (Default Value: 0x0000\_0032)

Offset: 0x0014			Register Name: SPI_ISR
Bit	Read/Write	Default/Hex	Description
31:14	/	/	/
13	R/W1C	0x0	SSI SS Invalid Interrupt When SSI is 1, it indicates that SPI_SS has changed from valid state to invalid state. Writing 1 to this bit clears it.
12	R/W1C	0x0	TC Transfer Completed In master mode, it indicates that all bursts specified by SPI_MBC have been exchanged. In other conditions, when setting, this bit indicates that all the data in TXFIFO has been loaded in the Shift register, and the Shift register has shifted out all the bits. Writing 1 to this bit clears it. 0: Busy 1: Transfer completed

Offset: 0x0014			Register Name: SPI_ISR
Bit	Read/Write	Default/Hex	Description
11	R/W1C	0x0	<p>TF_UDF TXFIFO Underrun This bit is set when the TXFIFO is underrun. Writing 1 to this bit clears it.</p> <p>0: TXFIFO is not underrun 1: TXFIFO is underrun</p>
10	R/W1C	0x0	<p>TF_OVF TXFIFO Overflow This bit is set when the TXFIFO is overflowed. Writing 1 to this bit clears it.</p> <p>0: TXFIFO is not overflowed 1: TXFIFO is overflowed</p>
9	R/W1C	0x0	<p>RX_UDF RXFIFO Underrun When set, this bit indicates that RXFIFO is underrun. Writing 1 to this bit clears it.</p> <p>0: RXFIFO is not underrun 1: RXFIFO is underrun</p>
8	R/W1C	0x0	<p>RX_OVF RXFIFO Overflow When set, this bit indicates that RXFIFO has overflowed. Writing 1 to this bit clears it.</p> <p>0: RXFIFO is not overflowed 1: RXFIFO is overflowed</p>
7	/	/	/
6	R/W1C	0x0	<p>TX_FULL TXFIFO Full This bit is set when the TXFIFO is full. Writing 1 to this bit clears it.</p> <p>0: TXFIFO is not Full 1: TXFIFO is Full</p>
5	R/W1C	0x1	<p>TX_EMP TXFIFO Empty This bit is set when the TXFIFO is empty. Writing 1 to this bit clears it.</p> <p>0: TXFIFO contains one or more words 1: TXFIFO is empty</p>

Offset: 0x0014			Register Name: SPI_ISR
Bit	Read/Write	Default/Hex	Description
4	R/W1C	0x1	<p>TX_READY TXFIFO Ready 0: TX_WL &gt; TX_TRIG_LEVEL 1: TX_WL &lt;= TX_TRIG_LEVEL</p> <p>This bit will be immediately set to 1 if TX_WL &lt;= TX_TRIG_LEVEL. Writing "1" to this bit clears it. The TX_WL is the water level of TXFIFO.</p>
3	/	/	/
2	R/W1C	0x0	<p>RX_FULL RXFIFO Full</p> <p>This bit is set when the RXFIFO is full. Writing 1 to this bit clears it.</p> <p>0: Not Full 1: Full</p>
1	R/W1C	0x1	<p>RX_EMP RXFIFO Empty</p> <p>This bit is set when the RXFIFO is empty. Writing 1 to this bit clears it.</p> <p>0: Not empty 1: empty</p>
0	R/W1C	0x0	<p>RX_RDY RXFIFO Ready</p> <p>0: RX_WL &lt; RX_TRIG_LEVEL 1: RX_WL &gt;= RX_TRIG_LEVEL</p> <p>This bit will be immediately set to 1 if RX_WL &gt;= RX_TRIG_LEVEL. Writing "1" to this bit clears it. The RX_WL is the water level of RXFIFO.</p>

### 8.3.6.5 0x0018 SPI FIFO Control Register (Default Value: 0x0040\_0001)

Offset: 0x0018			Register Name: SPI_FCR
Bit	Read/Write	Default/Hex	Description
31	R/WAC	0x0	<p>TX_FIFO_RST TX FIFO Reset</p> <p>Writing '1' to this bit will reset the control portion of the TX FIFO and auto clear to '0' when completing reset operation, writing to '0' has no effect.</p>

Offset: 0x0018			Register Name: SPI_FCR
Bit	Read/Write	Default/Hex	Description
30	R/W	0x0	<p>TF_TEST_ENB TX Test Mode Enable 0: Disable 1: Enable</p> <p>In normal mode, the TXFIFO can only be read by the SPI controller, writing '1' to this bit will switch the read and write function of the TXFIFO to AHB bus. This bit is used to test the TXFIFO, do not set in normal operation, and do not set RF_TEST and TF_TEST at the same time.</p>
29:25	/	/	/
24	R/W	0x0	<p>TF_DRQ_EN TXFIFO DMA Request Enable 0: Disable 1: Enable</p>
23:16	R/W	0x40	<p>TX_TRIG_LEVEL TXFIFO Empty Request Trigger Level</p>
15	R/WAC	0x0	<p>RF_RST RXFIFO Reset</p> <p>Writing '1' to this bit will reset the control portion of the receiver FIFO, and auto clear to '0' when completing reset operation, writing '0' to this bit has no effect.</p>
14	R/W	0x0	<p>RF_TEST RX Test Mode Enable 0: Disable 1: Enable</p> <p>In normal mode, the RXFIFO can only be written by the SPI controller, writing '1' to this bit will switch the read and write function of RXFIFO to AHB bus. This bit is used to test the RXFIFO, do not set in normal operation, and do not set RF_TEST and TF_TEST at the same time.</p>
13:9	/	/	/
8	R/W	0x0	<p>RF_DRQ_EN RX FIFO DMA Request Enable 0: Disable 1: Enable</p>
7:0	R/W	0x1	<p>RX_TRIG_LEVEL RX FIFO Ready Request Trigger Level</p>

### 8.3.6.6 0x001C SPI FIFO Status Register (Default Value: 0x0000\_0000)

Offset: 0x001C			Register Name: SPI_FSR
Bit	Read/Write	Default/Hex	Description
31	R	0x0	TB_WR TX FIFO Write Buffer Write Enable
30:28	R	0x0	TB_CNT TX FIFO Write Buffer Counter These bits indicate the number of words in TX FIFO Write Buffer
27:24	/	/	/
23:16	R	0x0	TF_CNT TX FIFO Counter These bits indicate the number of words in TX FIFO 0: 0 byte in TX FIFO 1: 1 byte in TX FIFO ... 64: 64 bytes in TX FIFO other: Reserved
15	R	0x0	RB_WR RX FIFO Read Buffer Write Enable
14:12	R	0x0	RB_CNT RX FIFO Read Buffer Counter These bits indicate the number of words in RX FIFO Read Buffer
11:8	/	/	/
7:0	R	0x0	RF_CNT RX FIFO Counter These bits indicate the number of words in RX FIFO 0: 0 byte in RX FIFO 1: 1 byte in RX FIFO ... 64: 64 bytes in RX FIFO other: Reserved

### 8.3.6.7 0x0020 SPI Wait Clock Register (Default Value: 0x0000\_0000)

Offset: 0x0020			Register Name: SPI_WCR
Bit	Read/Write	Default/Hex	Description
31:20	/	/	/
19:16	R/W	0x0	<p>SWC Dual mode direction switch wait clock counter (for master mode only).</p> <p>These bits control the number of wait states to be inserted before starting dual data transfer in dual SPI mode. The SPI module counts SPI_SCLK by SWC for delaying the next word data transfer.</p> <p>0: No wait states inserted n: n SPI_SCLK wait states inserted Cannot be written when XCH=1.</p>
15:0	R/W	0x0	<p>WCC Wait Clock Counter (In master mode)</p> <p>These bits control the number of wait states to be inserted in data transfers. The SPI module counts SPI_SCLK by WCC for delaying the next word data transfer.</p> <p>0: No wait states inserted n: n SPI_SCLK wait states inserted Cannot be written when XCH=1.</p>

### 8.3.6.8 0x0028 SPI Sample Delay Control Register (Default Value: 0x0000\_2000)

Offset: 0x0028			Register Name: SPI_SAMP_DL
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15	R/W	0x0	<p>SAMP_DL_CAL_START Sample Delay Calibration Start</p> <p>When set, the sample delay chain calibration is started.</p> <p>Cannot be written when XCH=1.</p>
14	R	0x0	<p>SAMP_DL_CAL_DONE Sample Delay Calibration Done</p> <p>When set, it means that sample delay chain calibration is done and the result of calibration is shown in SAMP_DL.</p> <p>Cannot be written when XCH=1.</p>

Offset: 0x0028			Register Name: SPI_SAMP_DL
Bit	Read/Write	Default/Hex	Description
13:8	R	0x20	<p>SAMP_DL Sample Delay</p> <p>It indicates the number of delay cells corresponding to the current card clock. The delay time generated by these delay cells is equal to the cycle of the card clock nearly.</p> <p>Generally, it is necessary to do drive delay calibration when the card clock is changed.</p> <p>This bit is valid only when SAMP_DL_CAL_DONE is set.</p> <p>Cannot be written when XCH=1.</p>
7	R/W	0x0	<p>SAMP_DL_SW_EN Sample Delay Software Enable</p> <p>When set, it indicates that enable sample delay specified at SAMP_DL_SW.</p> <p>Cannot be written when XCH=1.</p>
6	/	/	/
5:0	R/W	0x0	<p>SAMP_DL_SW Sample Delay Software</p> <p>The relative delay between the clock line and command line, data lines.</p> <p>It can be determined according to the value of SAMP_DL, the cycle of the card clock, and the input timing requirement of the device.</p> <p>Cannot be written when XCH=1.</p>

### 8.3.6.9 0x0030 SPI Master Burst Counter Register (Default Value: 0x0000\_0000)

Offset: 0x0030			Register Name: SPI_MBC
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/

Offset: 0x0030			Register Name: SPI_MBC
Bit	Read/Write	Default/Hex	Description
23:0	R/W	0x0	<p>MBC Master Burst Counter In master mode, this field specifies the total burst number. The total transfer data include the TXD, RXD, and dummy burst.</p> <p>0: 0 burst 1: 1 burst ... N: N bursts</p> <p>Cannot be written when XCH=1.</p>

### 8.3.6.10 0x0034 SPI Master Transmit Counter Register (Default Value: 0x0000\_0000)

Offset: 0x0034			Register Name: SPI_MTC
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23:0	R/W	0x0	<p>MWTC Master Write Transmit Counter In master mode, this field specifies the burst number that should be sent to TXFIFO before automatically sending dummy bursts. For saving bus bandwidth, the dummy bursts (all zero bits or all one bits) are sent by SPI Controller automatically.</p> <p>0: 0 burst 1: 1 burst ... N: N bursts</p> <p>Cannot be written when XCH=1.</p>

### 8.3.6.11 0x0038 SPI Master Burst Control Counter Register (Default Value: 0x0000\_0000)

Offset: 0x0038			Register Name: SPI_BCC
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/

Offset: 0x0038			Register Name: SPI_BCC
Bit	Read/Write	Default/Hex	Description
29	R/W	0x0	<p>Quad_EN Quad Mode Enable The quad mode includes Quad-Input and Quad-Output. 0: Quad mode disable 1: Quad mode enable Cannot be written when XCH=1.</p>
28	R/W	0x0	<p>DRM Master Dual Mode RX Enable It is only valid when Quad_Mode_EN=0. 0: RX uses the single-bit mode 1: RX uses the dual-bit mode Cannot be written when XCH=1.</p>
27:24	R/W	0x0	<p>DBC Master Dummy Burst Counter In master mode, this field specifies the burst number that should be sent before receiving in dual SPI mode. The data does not care by the device. 0: 0 burst 1: 1 burst ... N: N bursts Cannot be written when XCH=1.</p>
23:0	R/W	0x0	<p>STC Master Single Mode Transmit Counter In master mode, this field specifies the burst number that should be sent in the single mode before automatically sending dummy bursts. This is the first transmit counter in all bursts. 0: 0 burst 1: 1 burst ... N: N bursts Cannot be written when XCH=1.</p>

## 8.3.6.12 0x0040 SPI Bit-Aligned Transfer Configure Register (Default Value: 0x0000\_00A0)

Offset: 0x0040			Register Name: SPI_BATC
Bit	Read/Write	Default/Hex	Description
31	R/WAC	0x0	<p>TCE Transfer Control Enable In master mode, it is used to start to transfer the serial bit frame, it is only valid when Work Mode Select==0x10/0x11. 0: Idle 1: Initiates transfer Writing “1” to this bit will start to transfer serial bit frame (the value comes from the SPI TX Bit Register or SPI RX Bit Register), and will auto-clear after the bursts transfer completely. Writing ‘0’ to this bit has no effect.</p>
30	R/W	0x0	<p>MSMS Master Sample Standard 0: Delay Sample Mode 1: Standard Sample Mode In Standard Sample Mode, the SPI master samples the data at the standard rising edge of SCLK for each SPI mode; In Delay Sample Mode, the SPI master samples data at the edge that is half cycle delayed by the standard rising edge of SCLK defined in respective SPI mode.</p>
29:26	/	/	/
25	R/W1C	0x0	<p>TBC Transfer Bits Completed When set, this bit indicates that the last bit of the serial data frame in SPI TX Bit Register (or SPI RX Bit Register) has been transferred completely. Writing 1 to this bit clears it. 0: Busy 1: Transfer Completed It is only valid when Work Mode Select==0x10/0x11.</p>
24	R/W	0x0	<p>TBC_INT_EN Transfer Bits Completed Interrupt Enable 0: Disable 1: Enable It is only valid when Work Mode Select==0x10/0x11.</p>
23:22	/	/	/

Offset: 0x0040			Register Name: SPI_BATC
Bit	Read/Write	Default/Hex	Description
21:16	R/W	0x00	<p>Configure the length of serial data frame (burst) of RX</p> <p>000000: 0 bit 000001: 1 bit ... 100000: 32 bits Other values: reserved</p> <p>It is only valid when Work Mode Select==0x10/0x11, and cannot be written when TCE (bit31) is 1.</p>
15:14	/	/	/
13:8	R/W	0x00	<p>TX_FRM_LEN</p> <p>Configure the length of serial data frame (burst) of TX</p> <p>000000: 0 bit 000001: 1 bit ... 100000: 32 bits Other values: reserved</p> <p>It is only valid when Work Mode Select==0x10/0x11, and cannot be written when TCE (bit31) is 1.</p>
7	R/W	0x1	<p>SS_LEVEL</p> <p>When control the SS signal manually, set this bit to '1' or '0' to control the level of SS signal.</p> <p>0: Set SS to low 1: Set SS to high</p> <p>It is only valid when Work Mode Select==0x10/0x11, and only work in Mode0, cannot be written when TCE (bit31) is 1.</p>
6	R/W	0x0	<p>SS_OWNER</p> <p>SS Output Owner Select</p> <p>Usually, the controller sends the SS signal automatically with data together. When this bit is set to 1, the software must manually write SS_LEVEL to 1 or 0 to control the level of the SS signal.</p> <p>0: SPI controller 1: Software</p> <p>It is only valid when Work Mode Select==0x10/0x11, and only work in Mode0, cannot be written when TCE (bit31) is 1.</p>

Offset: 0x0040			Register Name: SPI_BATC
Bit	Read/Write	Default/Hex	Description
5	R/W	0x1	<p>SPOL SPI Chip Select Signal Polarity Control 0: Active high polarity (0 = Idle) 1: Active low polarity (1 = Idle)</p> <p>It is only valid when Work Mode Select==0x10/0x11, and only work in Mode0, cannot be written when TCE (bit31) is 1.</p>
4	/	/	/
3:2	R/W	0x0	<p>SS_SEL SPI Chip Select Select one of four external SPI Master/Slave Devices 00: SPI_SS0 will be asserted 01: SPI_SS1 will be asserted 10: SPI_SS2 will be asserted 11: SPI_SS3 will be asserted</p> <p>It is only valid when Work Mode Select= =0x10/0x11, and only work in Mode0, cannot be written when TCE (bit31) is 1.</p>
1:0	R/W	0x0	<p>Work Mode Select 00: Data frame is byte aligned in standard SPI, dual-output/dual input SPI, dual IO SPI, and quad-output/quad-input SPI 01: Reserved 10: Data frame is bit aligned in 3-wire SPI 11: Data frame is bit aligned in standard SPI</p>

### 8.3.6.13 0x0044 SPI Bit-Aligned CLOCK Configuration Register (Default Value: 0x0000\_0000)

Offset: 0x0044			Register Name: SPI_BA_CCR
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R/W	0x0	<p>CDR_N Clock Divide Rate (Master Mode Only)</p> <p>The SPI_SCLK is determined according to the following equation: <math>SPI\_CLK = Source\_CLK / (2 * (CDR_N + 1))</math>.</p> <p>This register is only valid when Work Mode Select==0x10/0x11.</p>

**8.3.6.14 0x0048 SPI TX Bit Register (Default Value: 0x0000\_0000)**

Offset: 0x0048			Register Name: SPI_TBR
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<p>VTB The Value of the Transmit Bits This register is used to store the value of the transmitted serial data frame. In the process of transmission, the LSB is transmitted first. This register is only valid when Work Mode Select==0x10/0x11.</p>

**8.3.6.15 0x004C SPI RX Bit Register (Default Value: 0x0000\_0000)**

Offset: 0x004C			Register Name: SPI_RBR
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<p>VRB The Value of the Receive Bits This register is used to store the value of the received serial data frame. In the process of transmission, the LSB is transmitted first. This register is only valid when Work Mode Select==0x10/0x11.</p>

**8.3.6.16 0x0088 SPI Normal DMA Mode Control Register (Default Value: 0x0000\_00E5)**

Offset: 0x0088			Register Name: SPI_NDMA_MODE_CTL
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:6	R/W	0x11	<p>SPI_ACT_M SPI NDMA Active Mode 00: dma_active is low 01: dma_active is high 10: dma_active is controlled by dma_request (DRQ) 11: dma_active is controlled by controller</p>
5	R/W	0x1	<p>SPI_ACK_M SPI NDMA Acknowledge Mode 0: active fall do not care ack 1: active fall must after detect ack is high</p>

Offset: 0x0088			Register Name: SPI_NDMA_MODE_CTL
Bit	Read/Write	Default/Hex	Description
4:0	R/W	0x05	SPI_DMA_WAIT The counts of hold cycles from DMA last signal high to dma_active high

### 8.3.6.17 0x0200 SPI TX Data Register (Default Value: 0x0000\_0000)

Offset: 0x0200			Register Name: SPI_TXD
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	TDATA Transmit Data This register can be accessed in the byte, half-word, or word unit by AHB. In the byte accessing method, if there are rooms in TXFIFO, one burst data is written to TXFIFO and the depth is increased by 1. In the half-word accessing method, two SPI burst data are written and the TXFIFO depth is increased by 2. In the word accessing method, four SPI burst data are written and the TXFIFO depth is increased by 4. <b>Note:</b> This address is writable-only if TF_TEST is '0', and if TF_TEST is set to '1', this address is readable and writable to test the TXFIFO through the AHB bus.

**8.3.6.18 0x0300 SPI RX Data Register (Default Value: 0x0000\_0000)**

Offset: 0x0300			Register Name: SPI_RXD
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	<p>RDATA Receive Data</p> <p>This register can be accessed in the byte, half-word, or word unit by AHB. In the byte accessing method, if there are data in RXFIFO, the top word is returned and the RXFIFO depth is decreased by 1. In the half-word accessing method, two SPI bursts are returned and the RXFIFO depth is decreased by 2. In the word accessing method, the four SPI bursts are returned and the RXFIFO depth is decreased by 4.</p> <p><b>Note:</b> This address is readable-only if RF_TEST is '0', and if RF_TEST is set to '1', this address is readable and writable to test the RXFIFO through the AHB bus.</p>

## 8.4 SPI\_DBI

### 8.4.1 Overview

The H133 provides a 3/4 line SPI display bus interface (SPI\_DBI) for video data transmission. It supports DBI mode or SPI mode. The DBI mode is compatible with multiple video data formats at the same time. The SPI mode is used for low-cost display schemes.

The SPI mode has the following features:

- Full-duplex synchronous serial interface
- Master/slave configurable
- 8-bit wide by 64-entry FIFO for both transmit and receive data
- Polarity and phase of the Chip Select (SPI\_SS) and SPI Clock (SPI\_SCLK) are configurable
- Supports interrupts and DMA
- Supports mode0, mode1, mode2, and mode3
- Supports 3-wire/4-wire SPI
- Supports programmable serial data frame length: 1 bit to 32 bits
- Supports standard SPI, dual-output/dual-input SPI, dual I/O SPI, quad-output/quad-input SPI
- Supports maximum IO rate of the mass production: 100 MHz

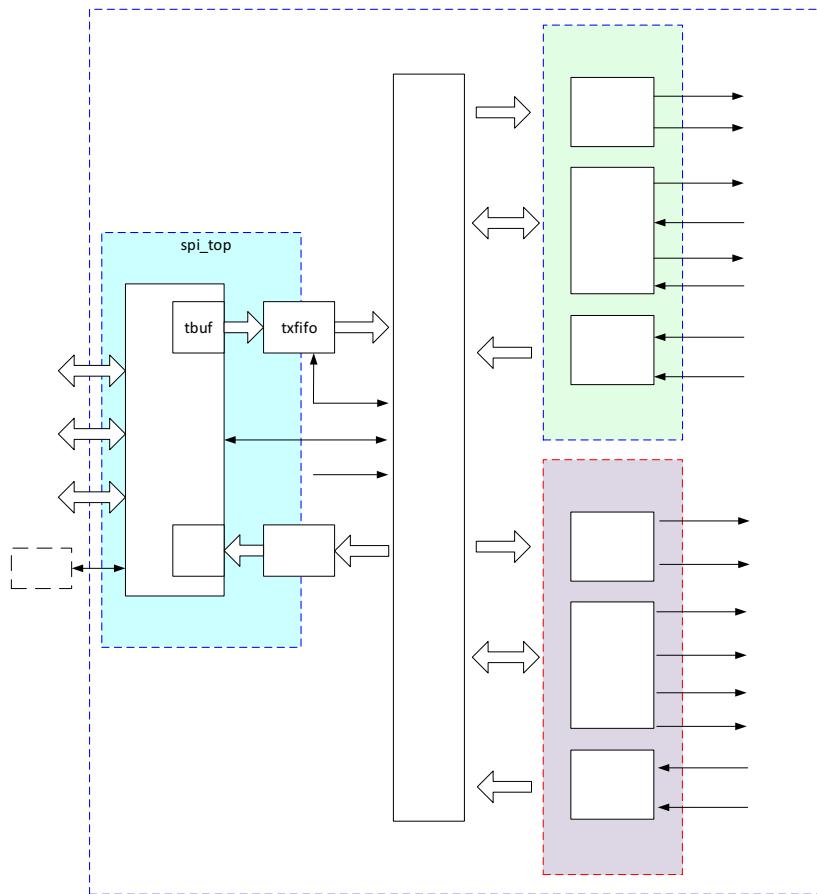
The DBI mode has the following features:

- Supports DBI Type C 3 Line/4 Line Interface Mode
- Supports 2 Data Lane Interface Mode
- Supports data source from CPU or DMA
- Supports RGB111/444/565/666/888 video format
- Maximum resolution of RGB666 240 x 320@30Hz with single data lane
- Maximum resolution of RGB888 240 x 320@60Hz or 320 x 480@30Hz with dual data lane
- Supports Tearing effect
- Supports software flexible control video frame rate

### 8.4.2 Block Diagram

Figure 8-30 shows a block diagram of the SPI\_DBI.

**Figure 8-30 SPI\_DBI Block Diagram**



SPI\_DBI contains the following sub-blocks:

**Table 8-12 SPI\_DBI Sub-blocks**

Sub-block	Description
spi_rf	Responsible for implementing the internal register, interrupt, and DMA Request.
spi_tbuf	The data length transmitted from AHB to TXFIFO is converted into 8 bits, then the data is written into the RXFIFO.
spi_rbuf	The block is used to convert the RXFIFO data into the reading data length of AHB.
txfifo, rxfifo	The data transmitted from the SPI to the external serial device is written into the TXFIFO; the data received from the external serial device into SPI is pushed into the RXFIFO.
spi_cmu	Responsible for implementing SPI bus clock, chip select, internal sample, and the generation of transfer clock.

Sub-block	Description
spi_tx	Responsible for implementing SPI data transfer, the interface of the internal TXFIFO, and status register.
spi_rx	Responsible for implementing SPI data receive, the interface of the internal RXFIFO, and status register.
dbi_ctrl	Responsible for implementing DBI bus clock, chip select, data command select, RGB format reshape.
dbi_tx	Responsible for implementing DBI data transfer, the interface of the internal TXFIFO, and status register.
dbi_rx	Responsible for implementing DBI data receive, the interface of the internal RXFIFO, and status register.

### 8.4.3 Functional Description

#### 8.4.3.1 External Signals

The following table describes the external signals of SPI\_DBI. When using SPI\_DBI, the corresponding PADs are selected as SPI\_DBI function via section 8.7 “[GPIO](#)”.

**Table 8-13 SPI\_DBI External Signals**

External Signal	Description	Type
DBI Mode	DBI-CSX	Chip select signal, low active
	DBI-SCLK	Serial clock signal
	DBI-SDO	Data output signal
	DBI-SDI	Data input signal, the data is sampled on the rising edge and the falling edge
	DBI-TE	Tearing effect input, it is used to capture the external TE signal edge.
	DBI-DCX	DCX pin is the select output signal of data and command. DCX = 0: register command; DCX = 1: data or parameter.
	DBI-WRX	When DBI operates in dual data lane format, the RGB666 format 2 can use WRX to transfer data
SPI Mode	SPI1-CS	SPI1 chip select signal, low active When the device is not selected, data will not be accepted via the SI pin, and the SO pin will stop transmission.

External Signal	Description	Type
SPI1-CLK	SPI1 clock signal This pin is used to provide a clock to the device and is used to control the flow of data to and from the device.	I/O
SPI1-MOSI	SPI1 master data out, slave data in	I/O
SPI1-MISO	SPI1 master data in, slave data out	I/O
SPI1-WP	Write protection and active low It also can be used for serial data input and output for SPI Quad Input or Quad Output mode.	I/O
SPI1-HOLD	When the device is selected and a serial sequence is underway, the HOLD pin is can be used to temporarily pause the serial communication with the master device without deselecting or resetting the device serial sequence. While the HOLD pin is asserted, the SO pin is at high impedance, and all transitions on the SCK pin and data on the SI pin are ignored.  It also can be used for serial data input and output for SPI Quad Input or Quad Output mode.	I/O

#### 8.4.3.2 Clock Sources

The SPI\_DBI controller gets 5 different clock sources, users can select one of them to make SPI\_DBI clock source. The following table describes the clock sources for SPI\_DBI. For more details on the clock setting, configuration, and gating information, see section 3.3 “[CCU](#)”.

**Table 8-14 SPI\_DBI Clock Sources**

Clock Sources	Description
HOSC	24 MHz Crystal
PLL_PERI(1X)	Peripheral Clock, the default value is 600 MHz
PLL_PERI(2X)	Peripheral Clock, the default value is 1200 MHz
PLL_AUDIO0(DIV2)	Audio Clock, the default value is 1536 MHz
PLL_AUDIO0(DIV5)	Audio Clock, the default value is 614.4 MHz

#### 8.4.3.3 Typical Application

Figure 8-31 shows the application block diagram when the SPI master device is connected to a slave device.

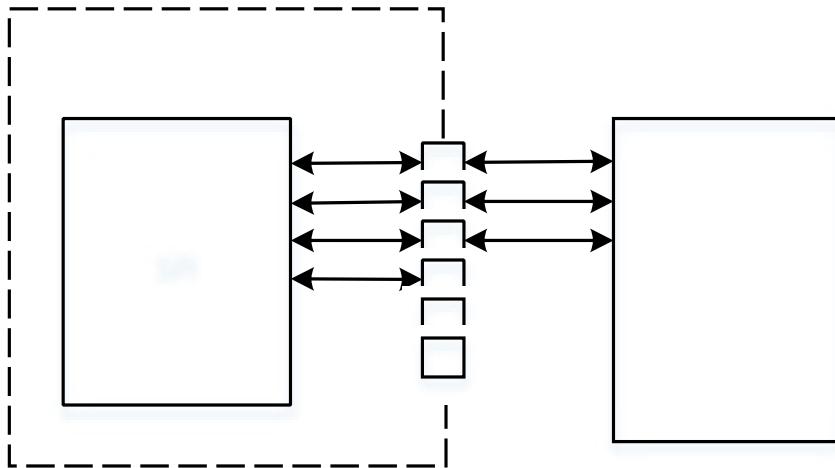
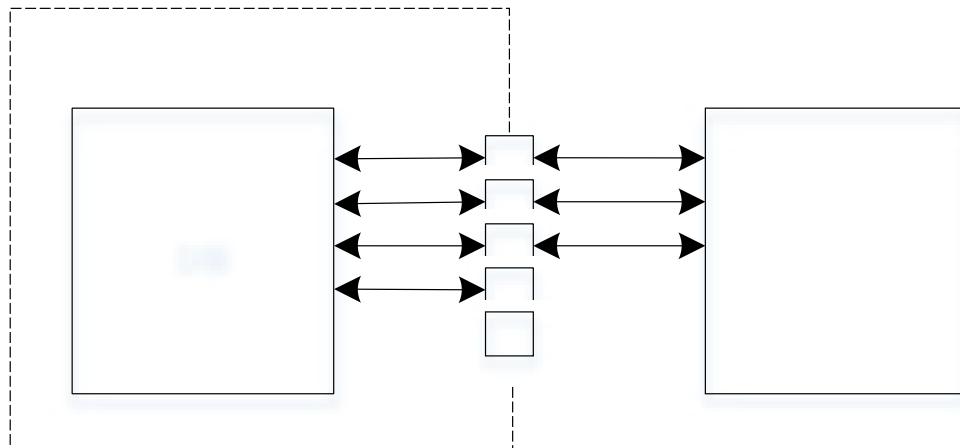
**Figure 8-31 SPI Application Block Diagram**

Figure 8-32 shows the application block diagram when the DBI master device is connected to a display bus interface device.

**Figure 8-32 DBI Application Block Diagram**

#### 8.4.3.4 SPI Transmission Format

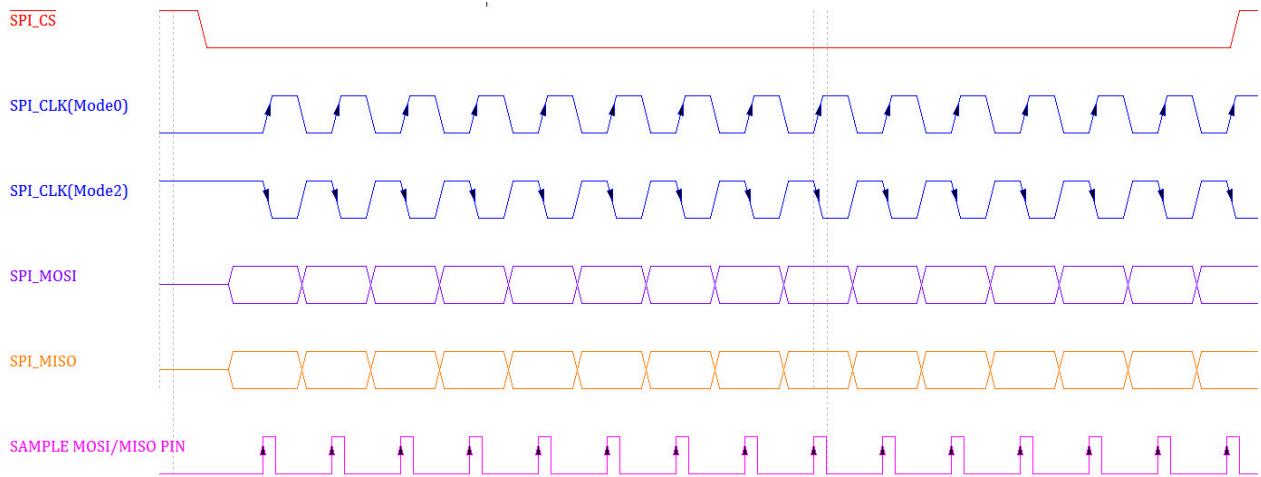
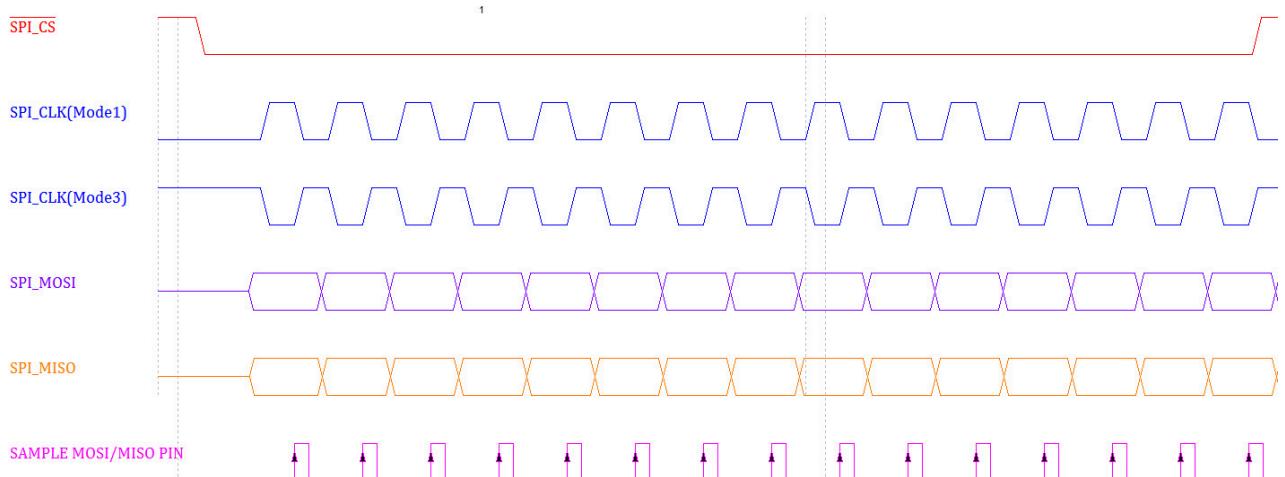
The SPI supports 4 different formats for data transmission. The software can select one of the four modes in which the SPI works by setting the bit1 (Polarity) and bit0 (Phase) of [SPI\\_TCR](#). The SPI controller master uses the SPI\_SCLK signal to transfer data in and out of the shift register. Data is clocked using any one of four programmable clock phase and polarity combinations.

The CPOL ([SPI\\_TCR\[1\]](#)) defines the polarity of the clock signal (SPI\_SCLK). The SPI\_SCLK is a high level when CPOL is '1' and it is a low level when CPOL is '0'. The CPHA ([SPI\\_TCR\[0\]](#)) decides whether the leading edge of SPI\_SCLK is used to setup or sample data. The leading edge is used to setup data when CPHA is '1', and sample data when CPHA is '0'. The following table lists the four modes.

**Table 8-15 SPI Transmit Format**

SPI Mode	Polarity (CPOL)	Phase (CPHA)	Leading Edge	Trailing Edge
mode0	0	0	Sample on the rising edge	Setup on the falling edge
mode1	0	1	Setup on the rising edge	Sample on the falling edge
mode2	1	0	Sample on the falling edge	Setup on the rising edge
mode3	1	1	Setup on the falling edge	Sample on the rising edge

Figure 8-33 and Figure 8-34 describe four waveforms for SPI\_SCLK.

**Figure 8-33 SPI Phase 0 Timing Diagram****Figure 8-34 SPI Phase 1 Timing Diagram**

#### 8.4.3.5 SPI Master and Slave Mode

The SPI controller can be configured to a master or slave device. The master mode is selected by setting the MODE bit ([SPI\\_GCR\[1\]](#)); the slave mode is selected by clearing the MODE bit.

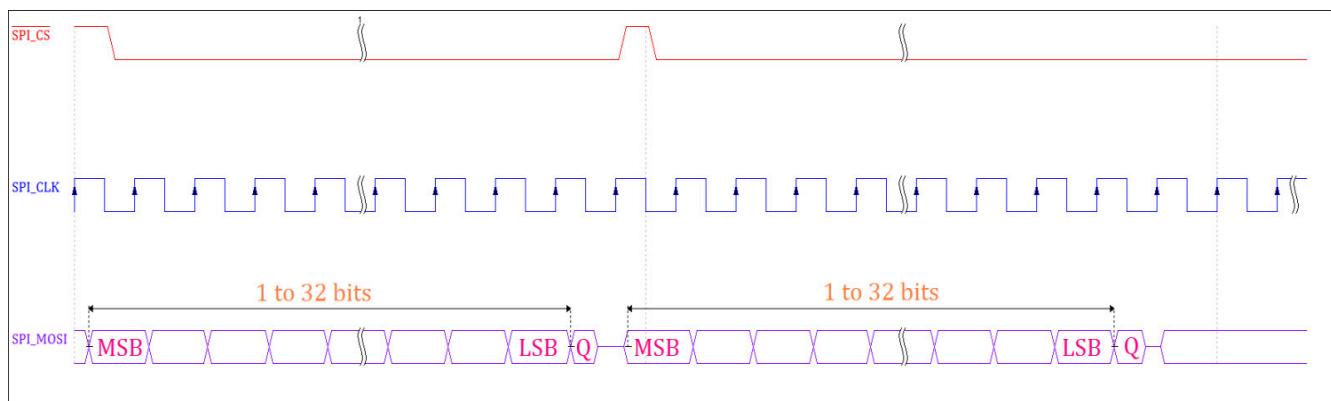
In master mode, the SPI\_CLK is generated and transmitted to the external device, and the data from the TX FIFO is transmitted on the MOSI pin, the data from the slave is received on the MISO pin and sent to RX FIFO. The Chip Select (SPI\_SS) is an active low signal, and it must be set low before the data are transmitted or received. The SPI\_SS can be selected the auto control mode or the software manual control mode. When using auto control, the SS\_OWNER ([SPI\\_TCR\[6\]](#)) must be cleared (default value is 0); when using manual control, the SS\_OWNER must be set. And the level of SPI\_SS is controlled by SS\_LEVEL ([SPI\\_TCR\[7\]](#)).

In slave mode, after the software selects the MODE bit ([SPI\\_GCR\[1\]](#)) to '0', it waits for master initiate a transaction. When the master asserts SPI\_SS, and SPI\_CLK is transmitted to the slave, the slave data is transmitted from TX FIFO on the MISO pin, and the data from the MOSI pin is received in RX FIFO.

#### 8.4.3.6 SPI 3-Wire Mode

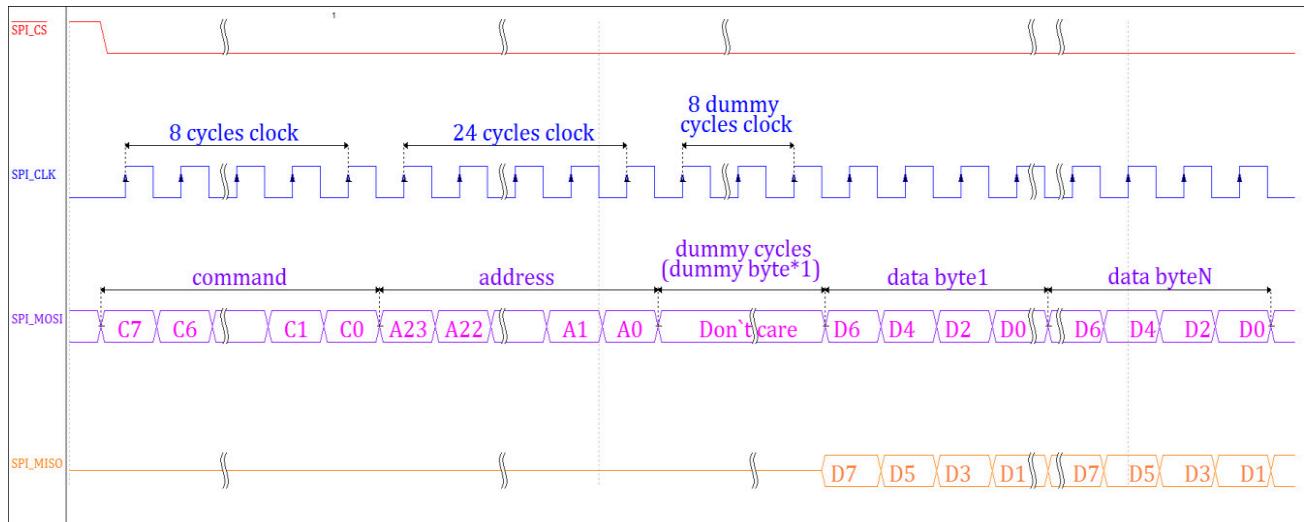
The SPI 3-wire mode is only valid when the SPI controller work in master mode, and is selected when the Work Mode Select bit ([SPI\\_BATC\[1:0\]](#)) is equal to 0x2. And in the 3-wire mode, the input data and the output data use the same single data line. The following figure describes the 3-wire mode.

**Figure 8-35 SPI 3-Wire Mode**

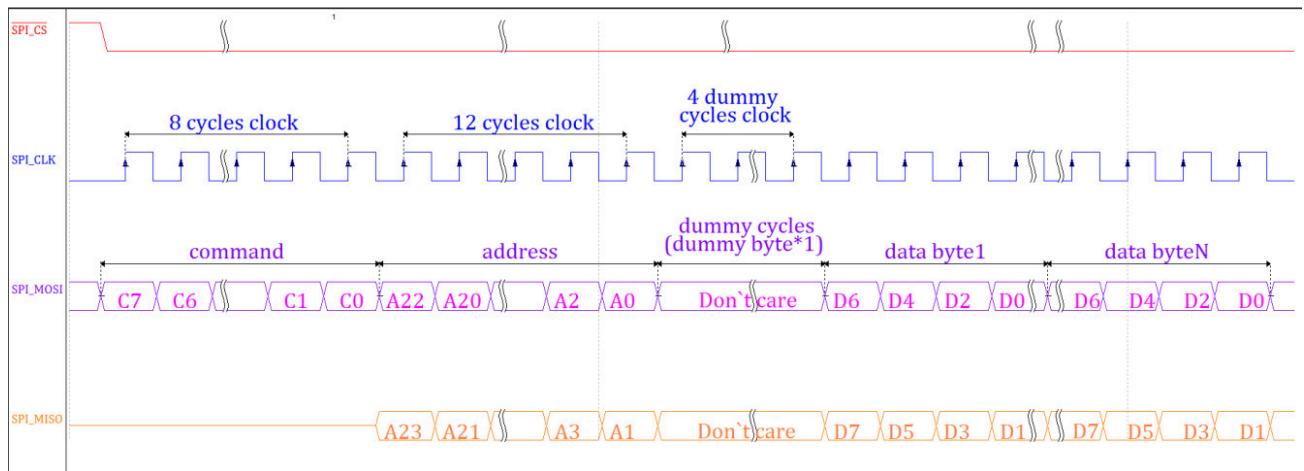


#### 8.4.3.7 SPI Dual-Input/Dual-Output and Dual I/O Mode

The dual read mode (SPI x2) is selected when the DRM is set in [SPI\\_BCC\[28\]](#). Using the dual mode allows data to be transferred to or from the device at double the rate of standard single mode SPI devices, the data can be read at fast speed using two data bits (MOSI and MISO) at a time. The following figure describes the dual-input/dual-output SPI (Figure 8-36) and the dual I/O SPI (Figure 8-37).

**Figure 8-36 SPI Dual-Input/Dual-Output Mode**

In the dual-input/dual-output SPI mode, the command, address, and the dummy bytes output in a unit of a single bit in serial mode through the SPI\_MOSI line, only the data bytes are output (write) and input (read) in a unit of dual bits through the SPI\_MOSI and SPI\_MISO.

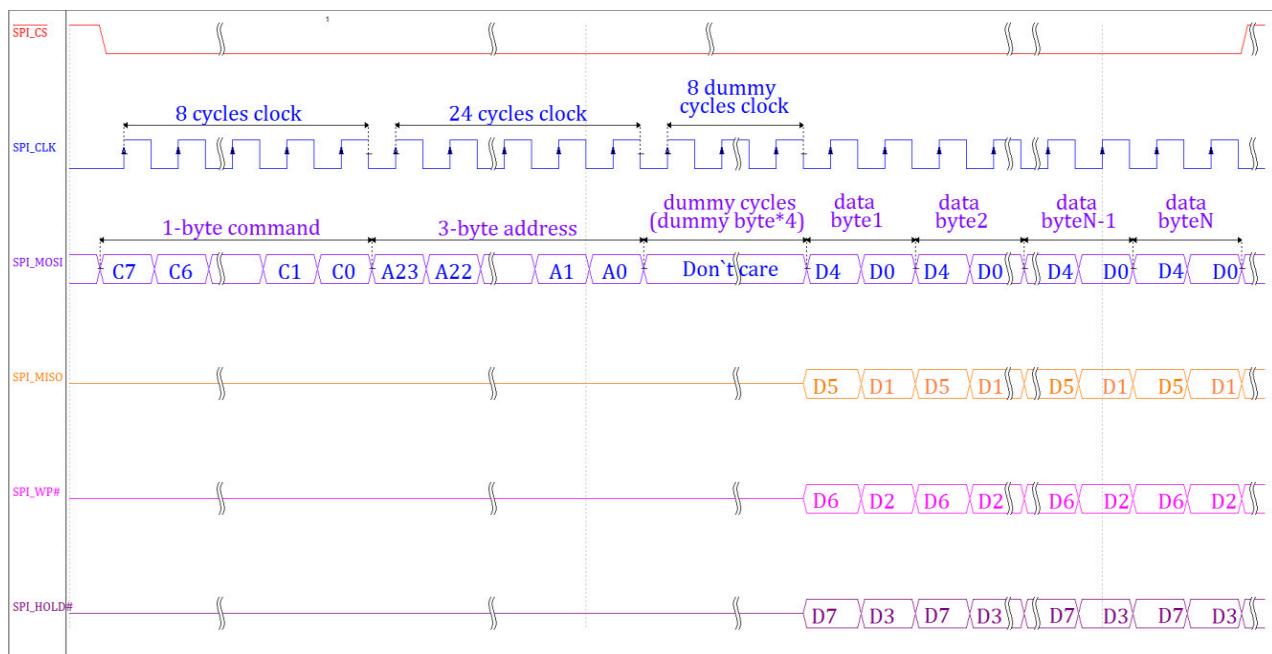
**Figure 8-37 SPI Dual I/O Mode**

In the dual I/O SPI mode, only the command bytes output in a unit of a single bit in serial mode through the SPI\_MOSI line. The address bytes and the dummy bytes output in a unit of dual bits through the SPI\_MOSI and SPI\_MISO. And the data bytes output (write) and input (read) in a unit of dual bits through the SPI\_MOSI and SPI\_MISO.

#### 8.4.3.8 SPI Quad-Input/Quad-Output Mode

The quad read mode (SPI x4) is selected when the Quad\_EN is set in [SPI\\_BCC\[29\]](#). Using the quad mode allows data to be transferred to or from the device at 4 times the rate of standard single mode SPI devices, the data can be read at fast speed using four data bits (MOSI, MISO, IO2 (WP#) and IO3 (HOLD#)) at the same time. The following figure describes the quad-input/quad-output SPI.

**Figure 8-38 SPI Quad-Input/Quad-Output Mode**



In the quad-input/quad-output SPI mode, the command, address, and the dummy bytes output in a unit of a single bit in serial mode through the SPI\_MOSI line. Only the data bytes output (write) and input (read) in a unit of quad bits through the SPI\_MOSI, SPI\_MISO, SPI\_WP#, and SPI\_HOLD#.

#### 8.4.3.9 Transmission/Reception Bursts in Master Mode

In SPI master mode, the transmission and reception bursts (byte in unit) are configured before the SPI transfers serial data between the processor and external device. The transmission bursts are written in MWTC (bit[23:0]) of the [SPI Master Transmit Counter Register](#). The transmission bursts in single mode before automatically sending dummy bursts are written in STC (bit[23:0]) of the [SPI Master Transmit Counter Register](#). For dummy data, the SPI controller can automatically send before receiving by writing DBC (bit[27:24]) in the [SPI Master Transmit Counter Register](#). If users do not use the SPI controller to send dummy data automatically, then the dummy bursts are used as the transmission counters to write together in MWTC (bit[23:0]) of the [SPI Master Transmit Counter Register](#). In master mode, the total burst numbers are written in MBC (bit[23:0]) of the [SPI Master Transmit Counter Register](#). When all transmission and reception bursts are transferred, the SPI controller will send a completed interrupt, at the same time, the SPI controller will clear DBC, MWTC, and MBC.

#### 8.4.3.10 SPI Sample Mode and Run Clock Configuration

The SPI controller runs at 3 kHz–100 MHz at its interface to external SPI devices. The internal SPI clock should run at the same frequency as the outgoing clock in the master mode. The SPI clock is selected from different clock sources, the SPI must configure different work mode. There are three work modes: normal sample mode, delay half-cycle sample mode, delay one-cycle sample mode. Delay half-cycle sample mode is the default mode of the SPI controller. When the SPI runs at 40 MHz or below 40 MHz, the SPI can work at normal sample mode or delay half-cycle sample mode. When the SPI runs over 80 MHz, setting the SDC bit in the [SPI Transfer Control Register](#) to '1' makes the internal read sample point with a half-cycle delay of SPI\_CLK, which is used in high speed read operation to reduce the error caused by the time delay of SPI\_CLK between master and slave. Table 8-16 and Table 8-17 show the different configurations of the SPI sample mode.

**Table 8-16 SPI Old Sample Mode and Run Clock**

SPI Sample Mode	SDM(bit13)	SDC(bit11)	Run Clock
normal sample	1	0	<=24 MHz
delay half cycle sample	0	0	<=40 MHz
delay one cycle sample	0	1	>=80 MHz



**CAUTION**

The remaining spectrum is not recommended. Because when the output delay of SPI flash (refer to the datasheet of the manufacturer for the specific delay time) is the same with the half-cycle time of SPI working clock, the variable edge of the output data for the device bumps into the clock sampling edge of the controller, so setting 1 cycle of sampling delay would cause stability problem.

**Table 8-17 SPI New Sample Mode**

SPI Sample Mode	SDM (bit13)	SDC (bit11)	SDC1 (bit15)
normal sample	1	0	0
delay half cycle sample	0	0	0
delay one cycle sample	0	1	0
delay 1.5 cycle sample	1	1	0
delay 2 cycle sample	1	0	1
delay 2.5 cycle sample	0	0	1
delay 3 cycle sample	0	1	1

#### 8.4.3.11 DBI 3-Line Interface Writing and Reading Timing

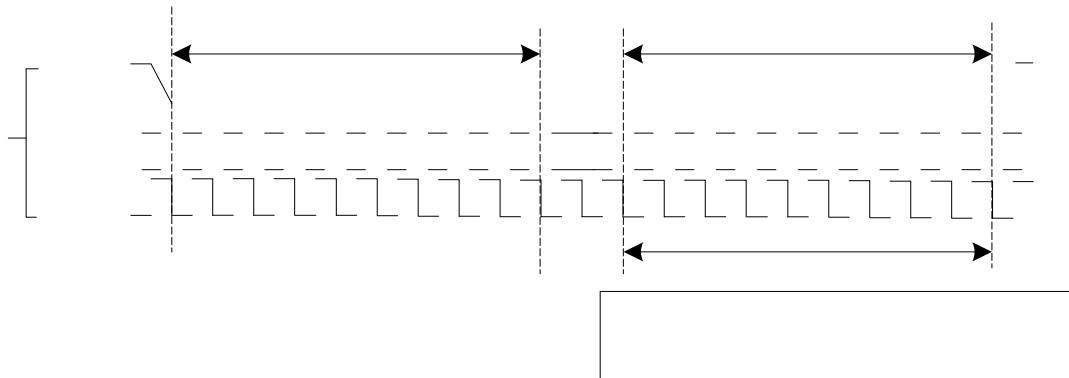
The 3-line DBI Interface I contains CSX, SDA, and SCL, where SDA shares this port for bidirectional port data input and output.

The 3-line DBI Interface II contains CSX, SDA, SCL, and SDI; Data input and output ports are independent of each other.

Since the 3-line display bus mode has no Data/Command data line indicating whether Data or Command is currently being transmitted, an extra bit is added to the data-stream before MSB to indicate whether Data or Command is currently being transmitted. (0: Command, 1: Data)

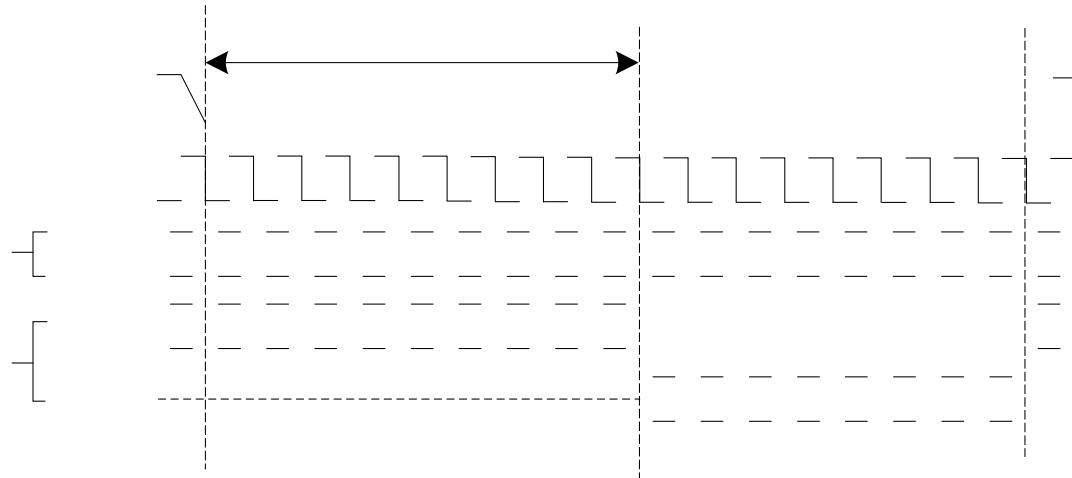
The following figure shows the writing operation format of 3-line DBI Interface I and Interface II.

**Figure 8-39 DBI 3-Line Display Bus Serial Interface Writing Operation Format**

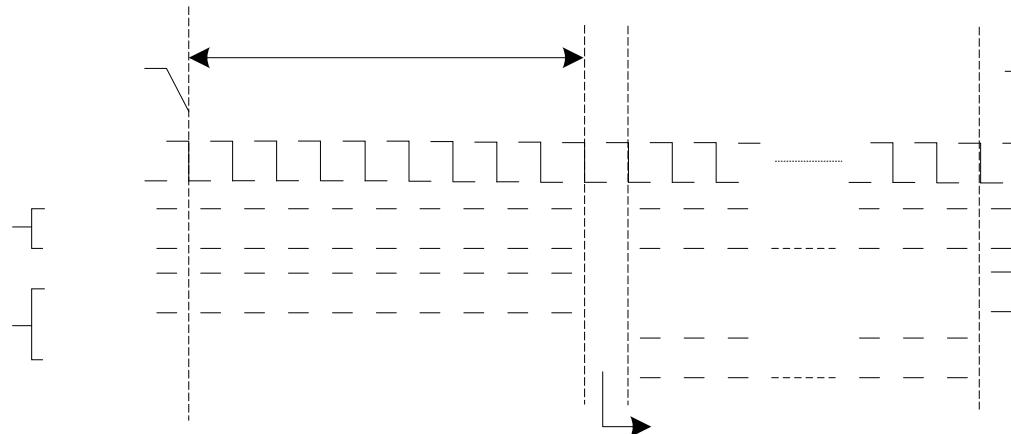


The 3-line DBI Interface I uses the SDA port as bidirectional data input and output port. There are only three cases of data reading volume, 8bits/24bits/32bits, and the first data sampled is high.

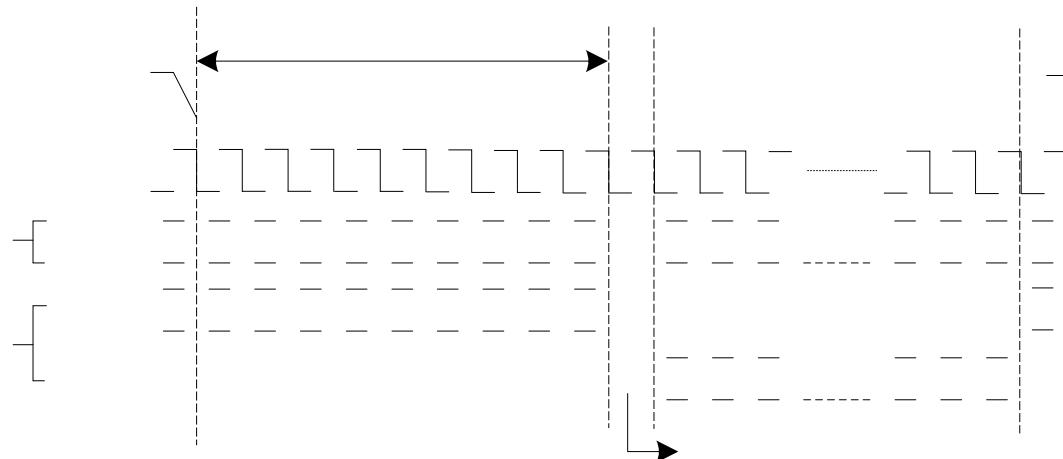
The following figure shows the 8 bits reading operation format of 3-line DBI Interface I and Interface II. After the read command is transmitted, the data is read immediately with on dummy period.

**Figure 8-40 DBI 3-Line Display Bus Serial Interface 8-bit Reading Operation Format**

The following figure shows the 24 bits reading operation format of 3-line DBI Interface I and Interface II. After the read command is transmitted, the data is read after waiting for the dummy clock cycle.

**Figure 8-41 DBI 3-Line Display Bus Serial Interface 24-bit Reading Operation Format**

The following figure shows the 32 bits reading operation format of 3-line DBI Interface I and Interface II. After the read command is transmitted, the data is read after waiting for the dummy clock cycle.

**Figure 8-42 DBI 3-Line Display Bus Serial Interface 32-bit Reading Operation Format**

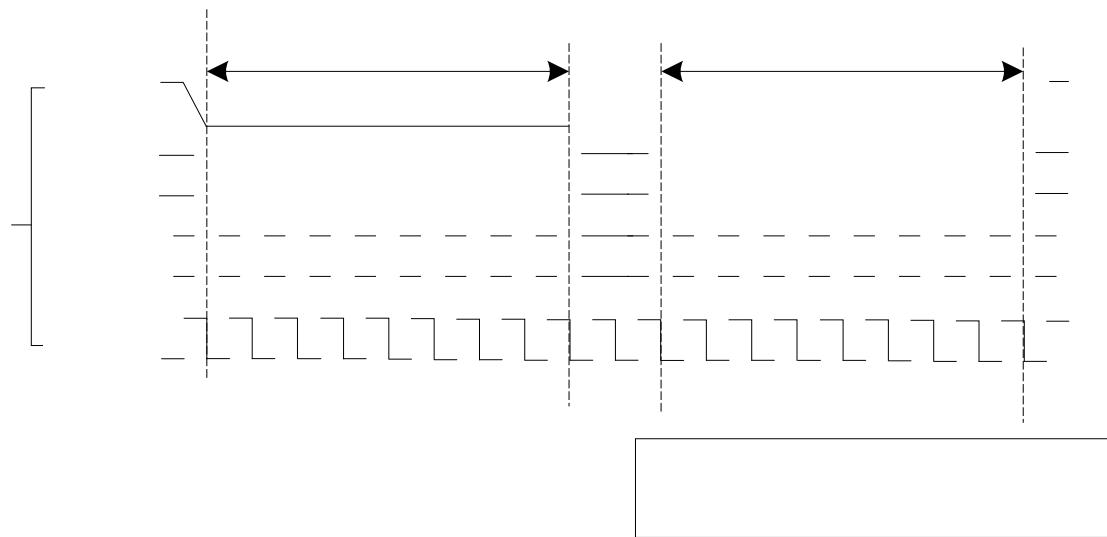
#### 8.4.3.12 DBI 4-Line Interface Writing and Reading Timing

The 4-line DBI Interface I contains CSX, D/CX, SDA, and SCL, where SDA shares this port for bidirectional port data input and output.

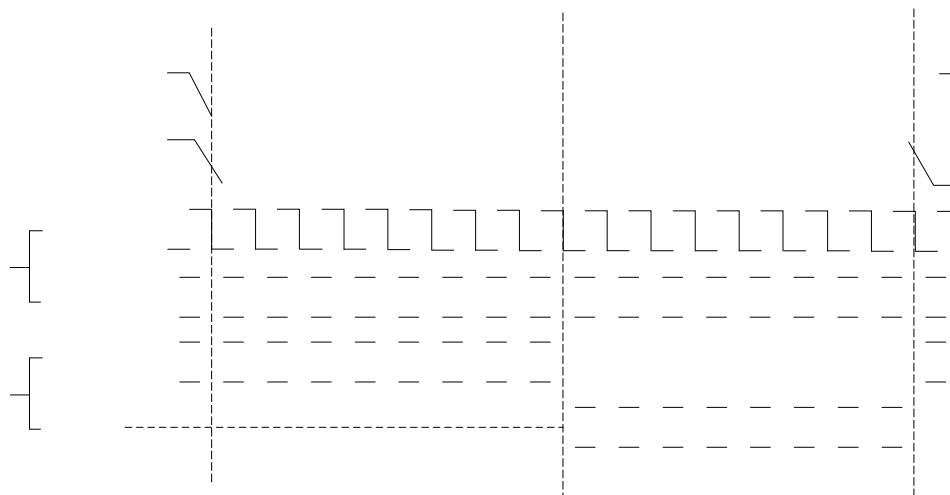
The 4-line DBI Interface II contains CSX, D/CX, SDA, SCL, and SDI; Data input and output ports are independent of each other.

Since the 4-line display bus mode has a Data/Command data line indicating whether Data or Command is currently being transmitted (0: Command, 1: Data). So there is no need to add an extra bit to data-stream before MSB like the 3-line DBI.

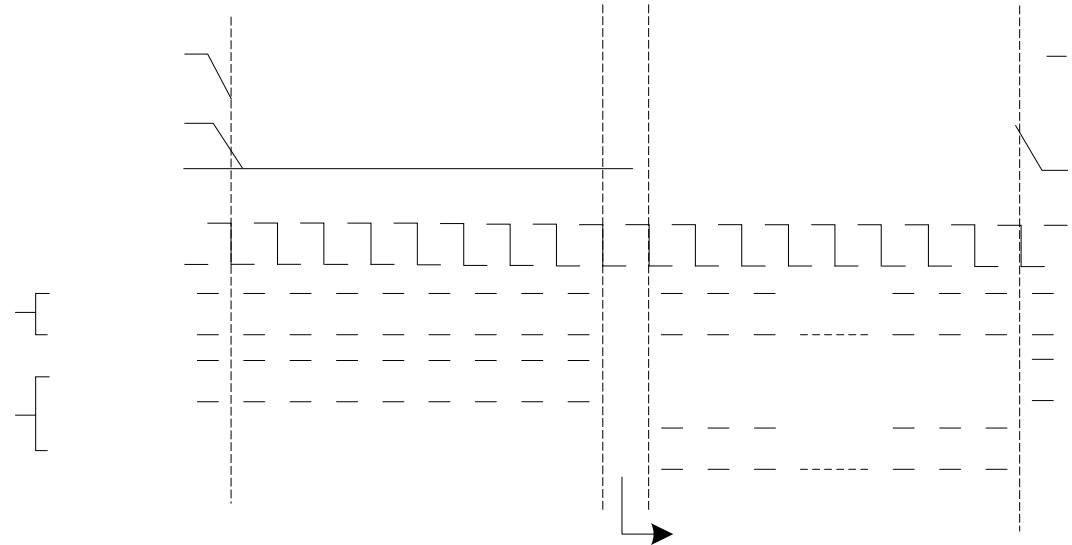
The following figure shows the writing operation format of 4-line DBI Interface I and Interface II.

**Figure 8-43 DBI 4-Line Display Bus Serial Interface Writing Operation Format**

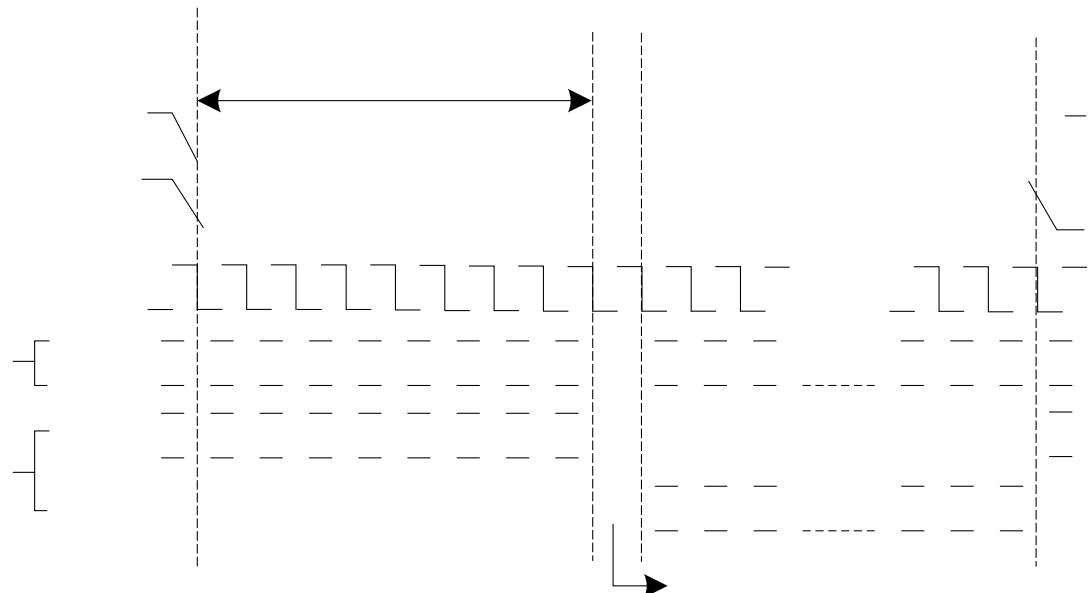
The following figure shows the 8 bits reading operation format of 4-line DBI Interface I and Interface II.

**Figure 8-44 DBI 4-Line Display Bus Serial Interface 8-bit Reading Operation Format**

The following figure shows the 24 bits reading operation format of 4-line DBI Interface I and Interface II.

**Figure 8-45 DBI 4-Line Display Bus Serial Interface 24-bit Reading Operation Format**

The following figure shows the 32 bits reading operation format of 4-line DBI Interface I and Interface II.

**Figure 8-46 DBI 4-Line Display Bus Serial Interface 32-bit Reading Operation Format**

#### 8.4.3.13 DBI 3-Line Interface Transmit Video Format

Figure 8-47 RGB111 3-Line Interface Transmit Video Format

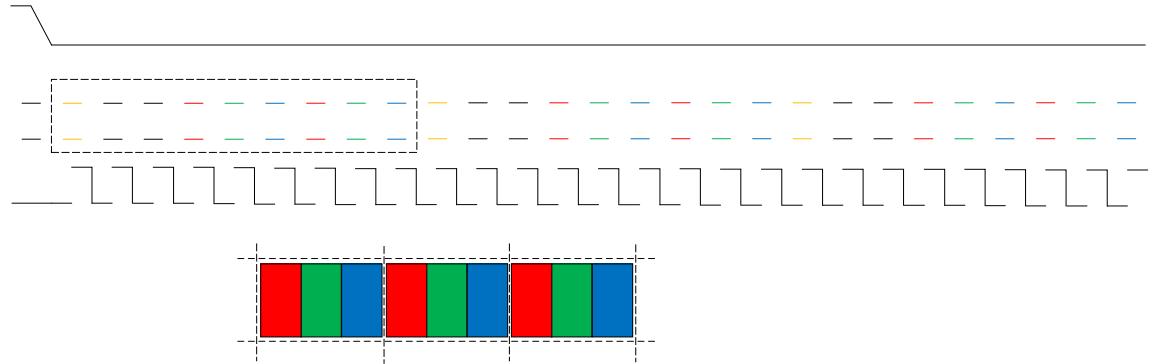


Figure 8-48 RGB444 3-Line Interface Transmit Video Format

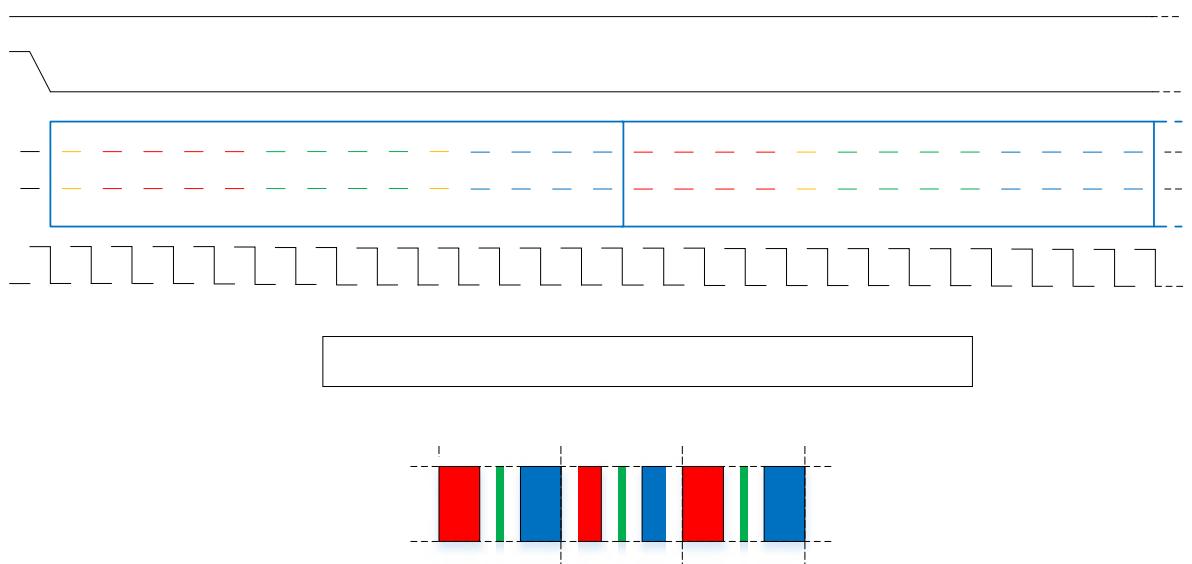


Figure 8-49 RGB565 3-Line Interface Transmit Video Format

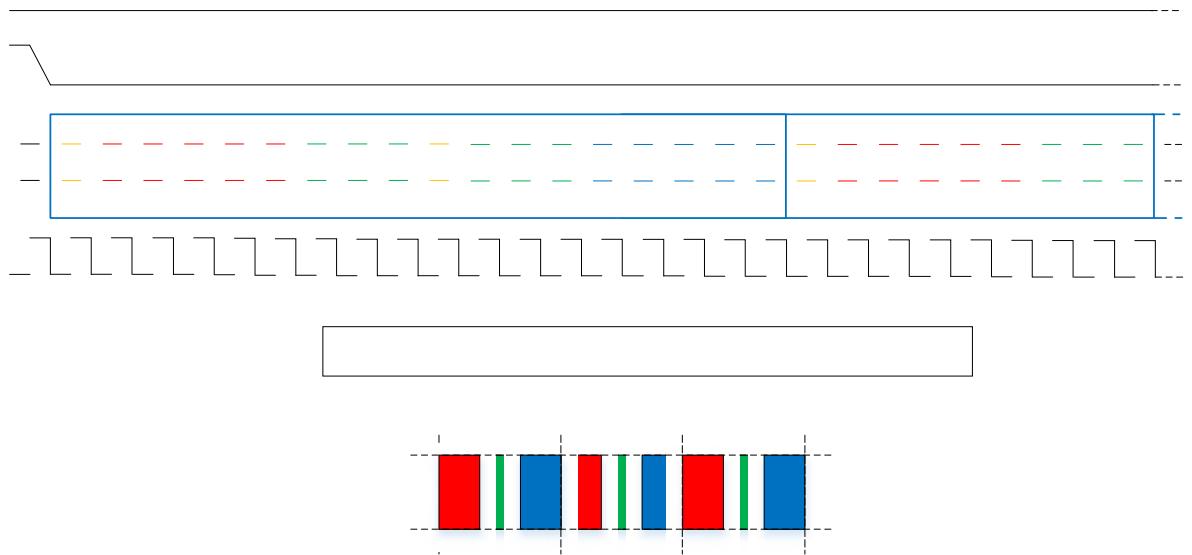
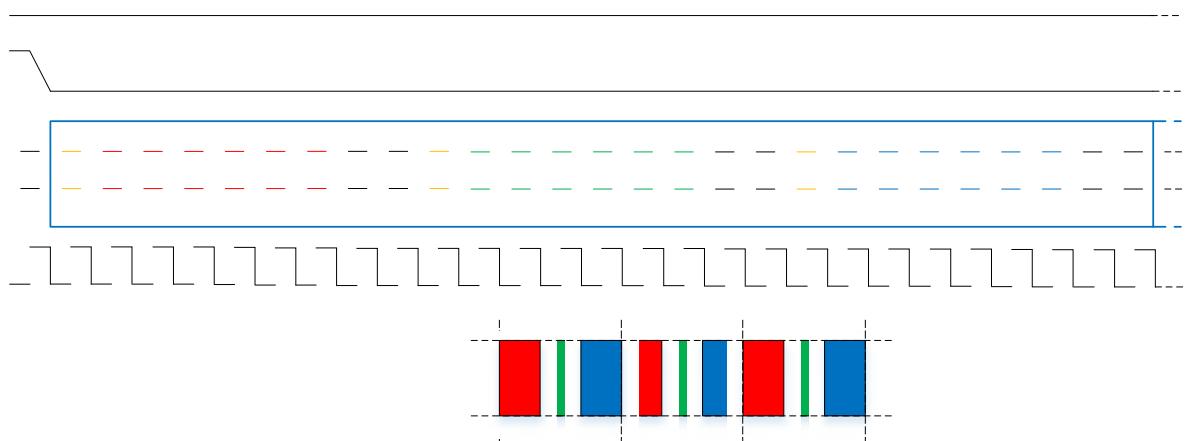


Figure 8-50 RGB666 3-Line Interface Transmit Video Format



#### 8.4.3.14 DBI 4-Line Interface Transmit Video Format

Figure 8-51 RGB111 4-Line Interface Transmit Video Format

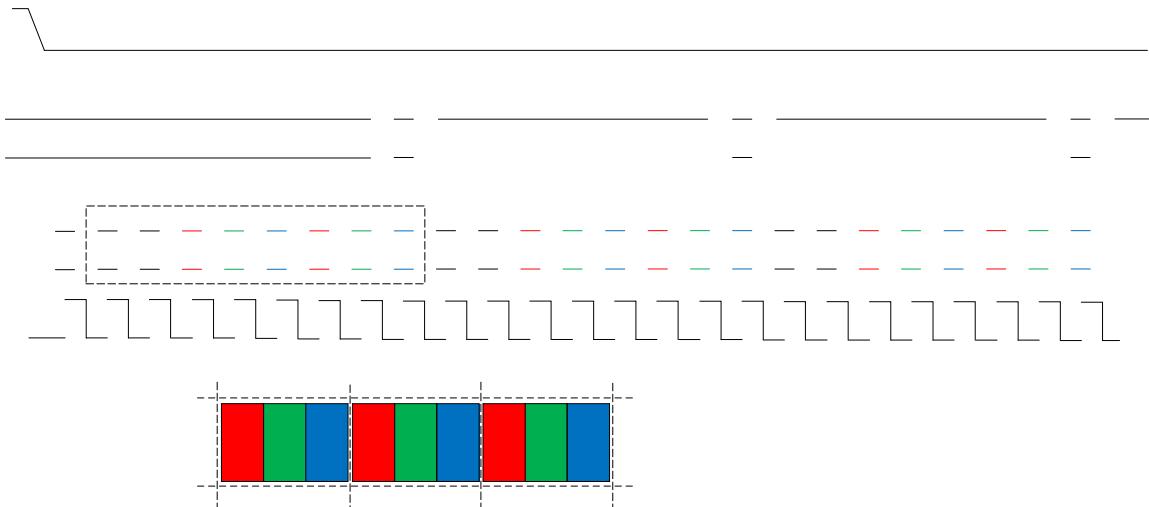


Figure 8-52 RGB444 4-Line Interface Transmit Video Format

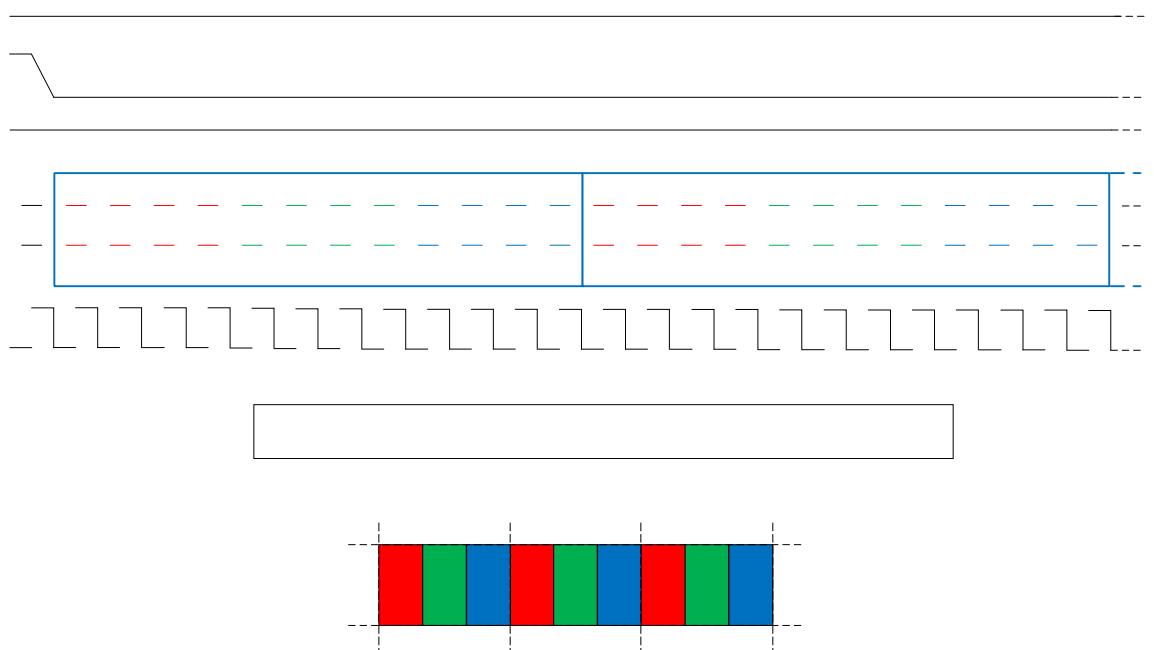


Figure 8-53 RGB565 4-Line Interface Transmit Video Format

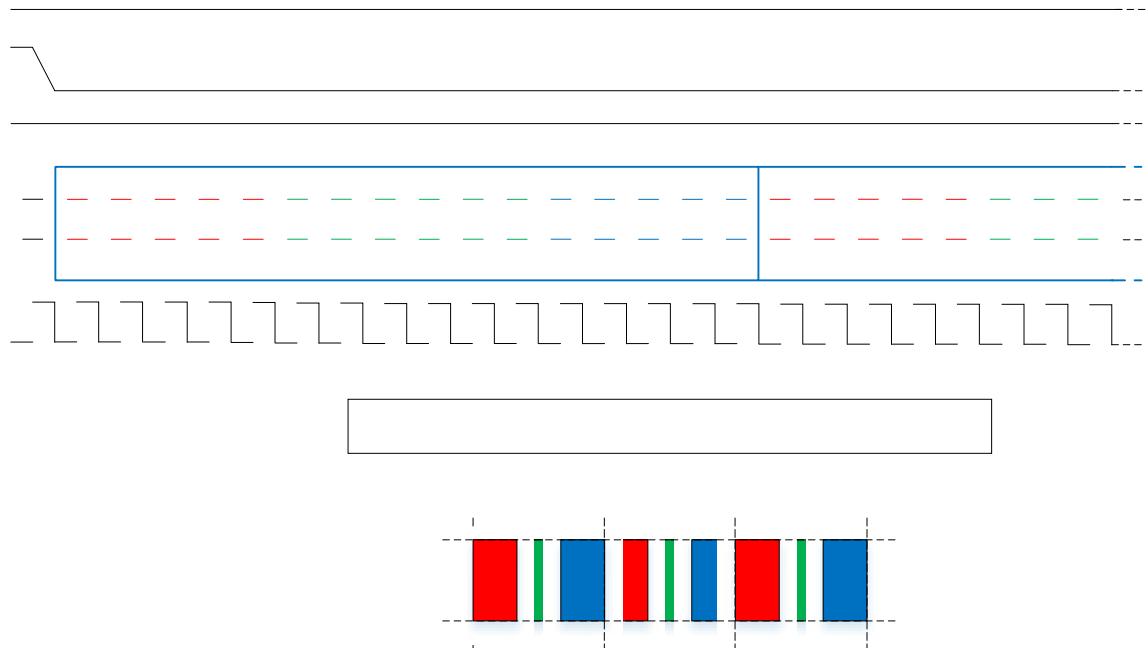
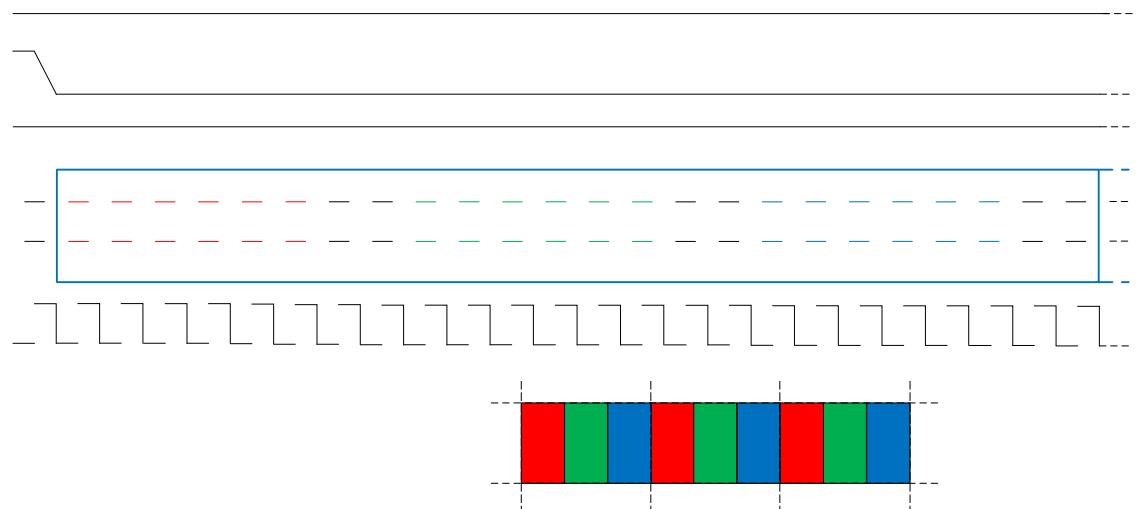


Figure 8-54 RGB666 4-Line Interface Transmit Video Format



#### 8.4.3.15 DBI 2 Data Lane Interface Transmit Video Format

For RGB444:

Figure 8-55 RGB444 2 Data Lane Interface Transmit Video Format

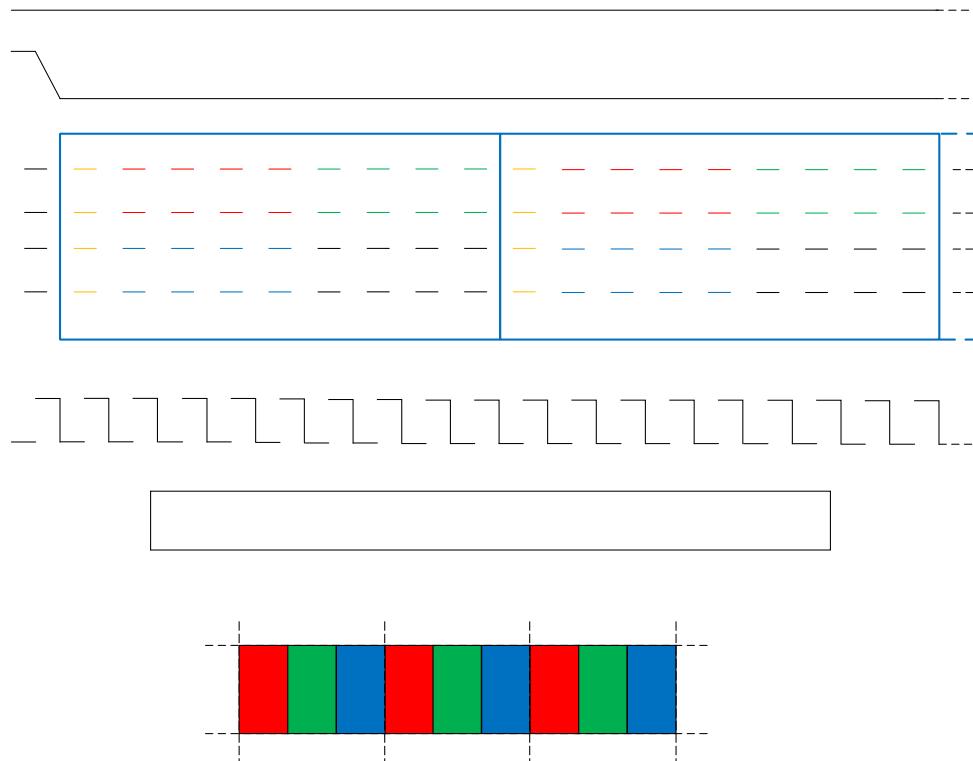
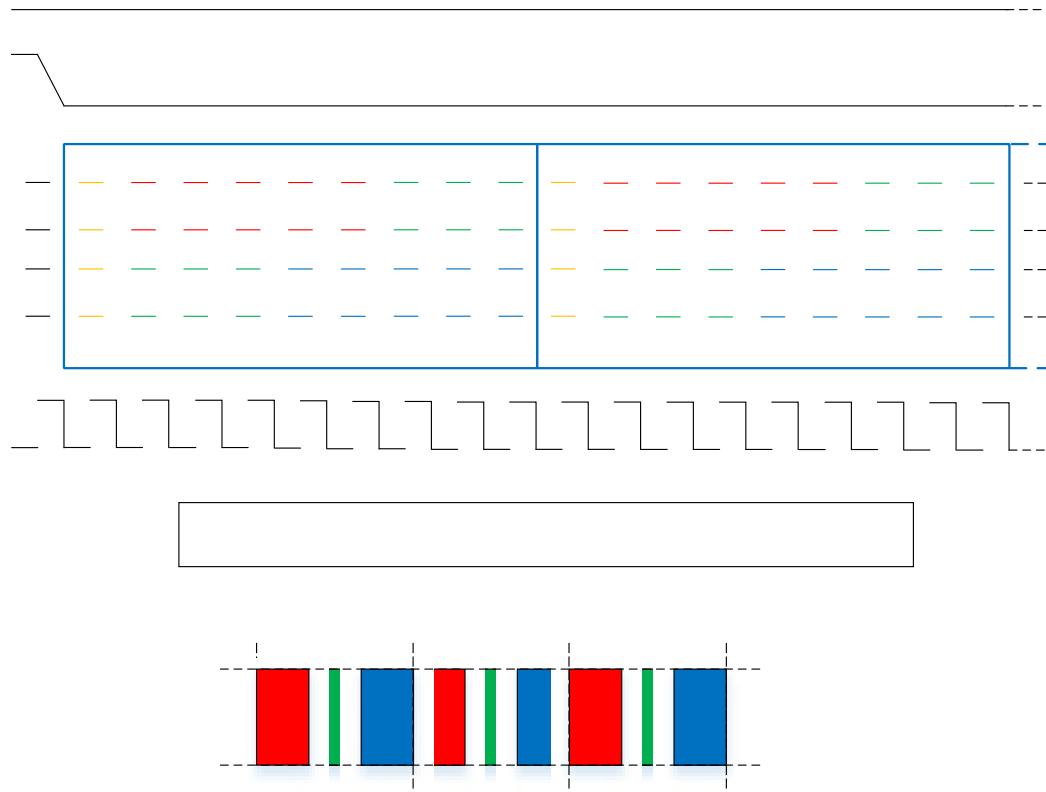


Figure 8-56 RGB565 2 Data Lane Interface Transmit Video Format



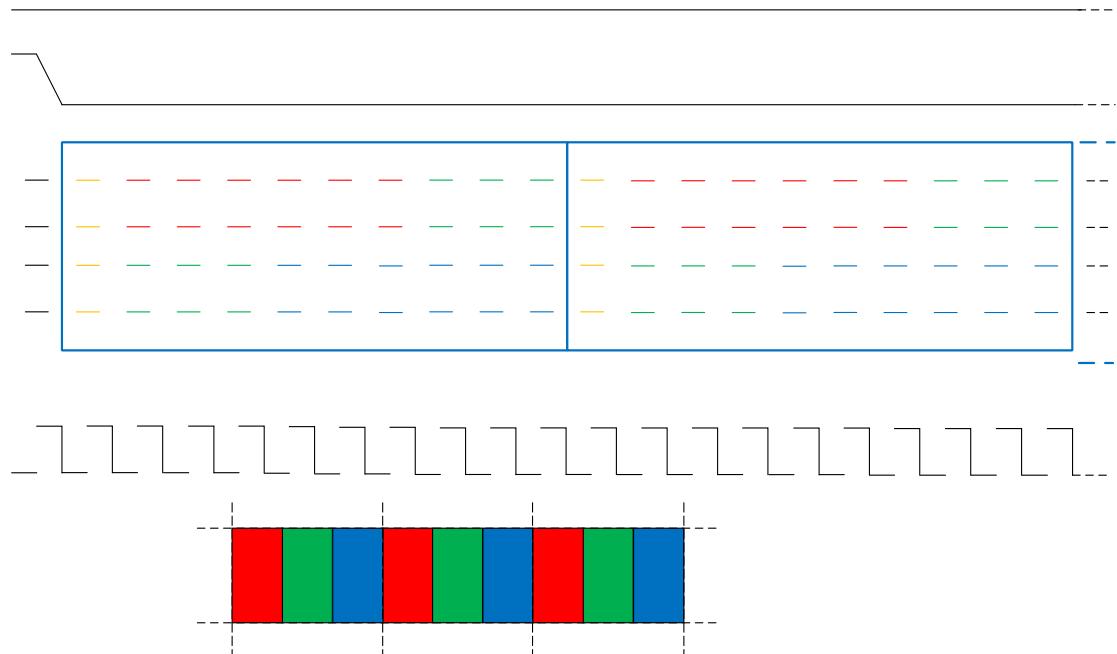
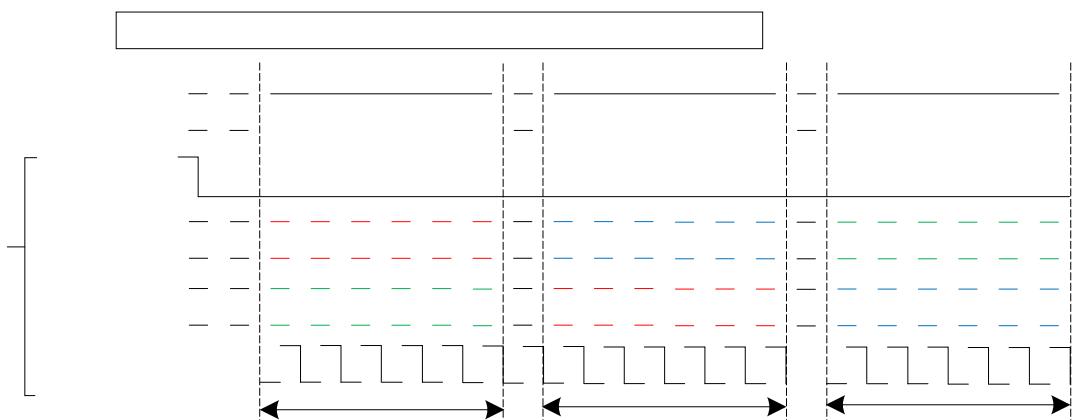
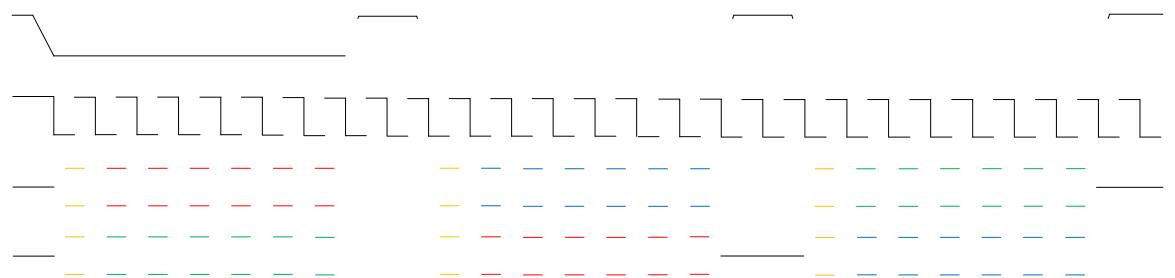
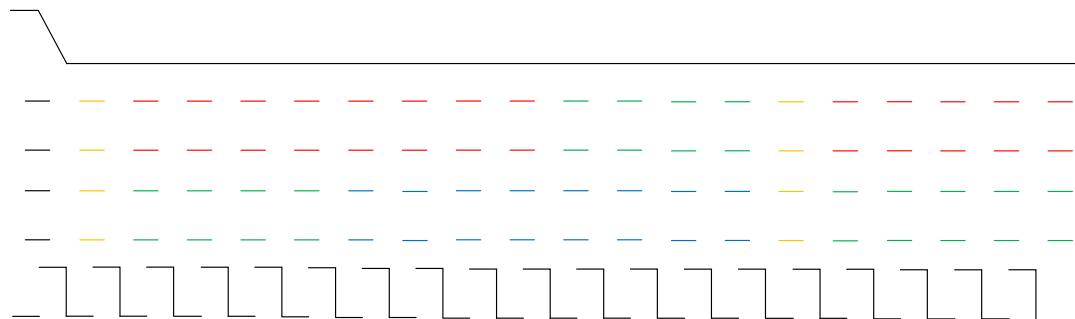
**Figure 8-57 RGB666 2 Data Lane Interface Transmit Video Format 0****Figure 8-58 RGB666 2 Data Lane Interface Transmit Video Format 1 (ilitek)****Figure 8-59 RGB666 2 Data Lane Interface Transmit Video Format 2 (New vision)**

Figure 8-60 RGB888 2 Data Lane Interface Transmit Video Format



#### 8.4.4 Programming Guidelines

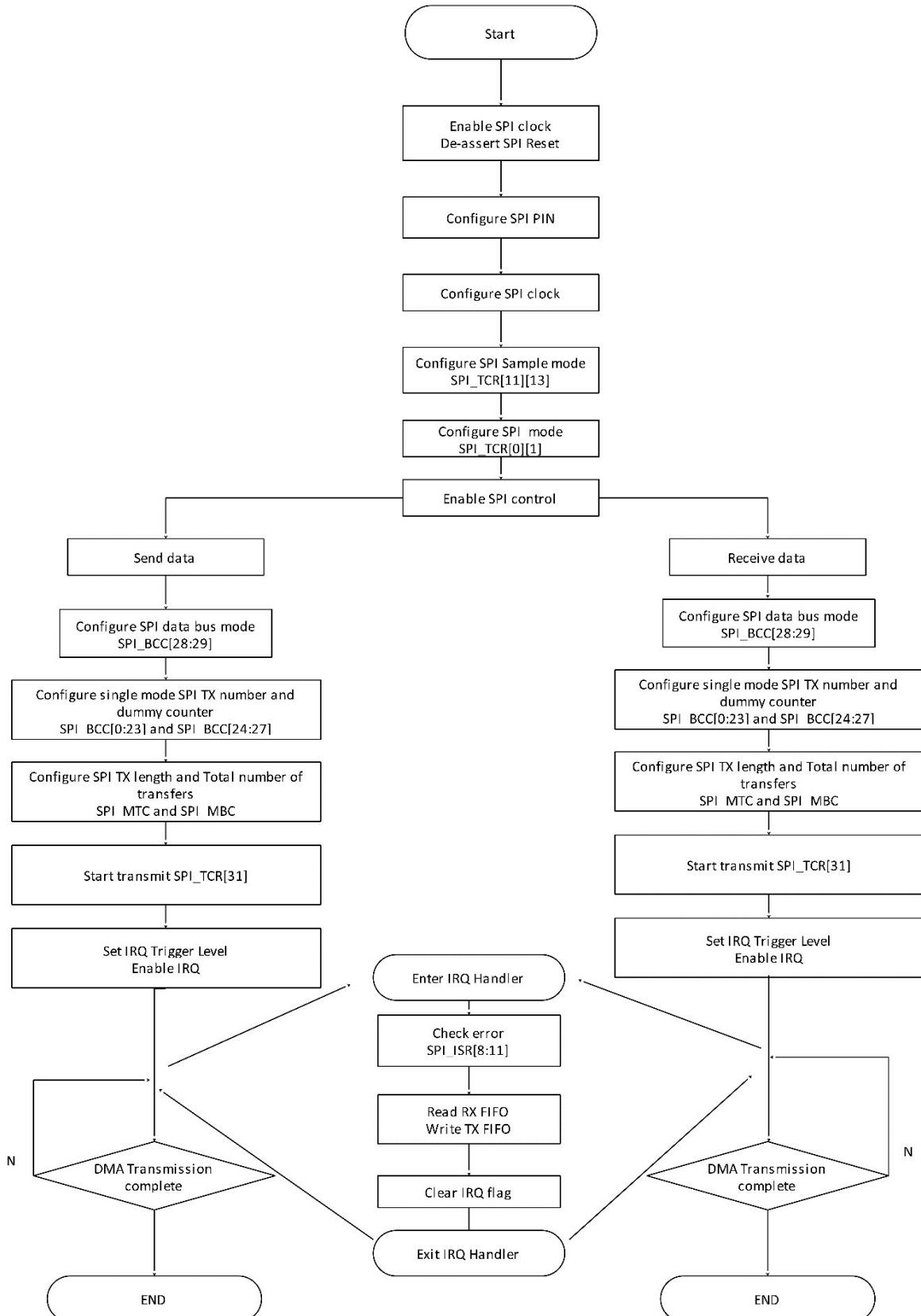
##### 8.4.4.1 Writing/Reading Data Process Using SPI Mode

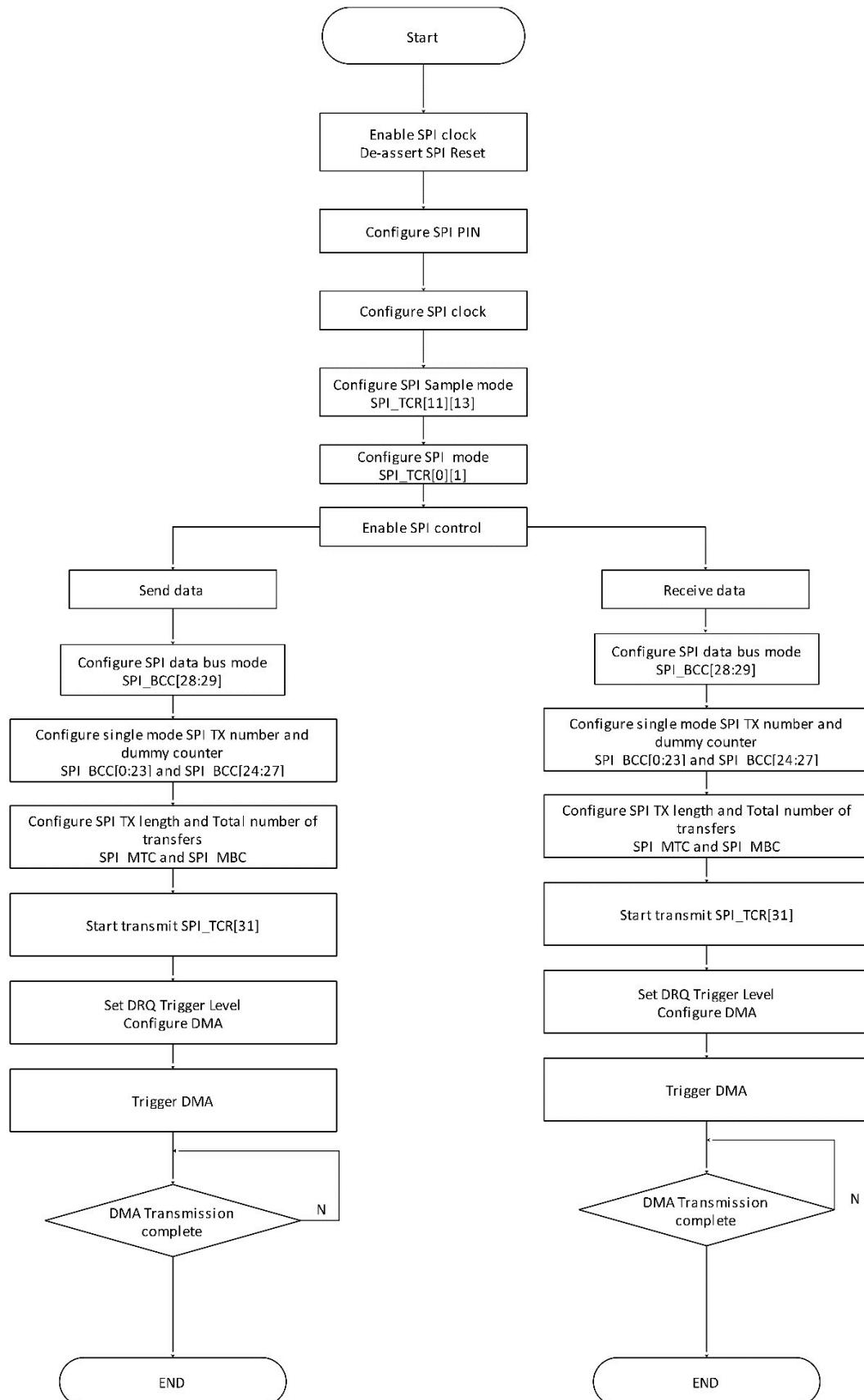
The SPI transfers serial data between the processor and the external device. CPU and DMA are the two main operational modes for SPI. For each SPI, the data is simultaneously transmitted (shifted out serially) and received (shifted in serially). The SPI has 2 channels, including the TX channel and RX channel. The TX channel has the path from TX FIFO to the external device. The RX channel has the path from the external device to RX FIFO.

**Write Data:** CPU or DMA must write data on the [SPI\\_TXD](#) register, the data on the register are automatically moved to TX FIFO.

**Read Data:** To read data from RX FIFO, CPU or DMA must access the register [SPI\\_RXD](#) and data are automatically sent to the register [SPI\\_RXD](#).

In CPU or DMA mode, the SPI sends a completed interrupt ([SPI\\_ISR\[TC\]](#)) to the processor at the end of each transfer.

**CPU Mode****Figure 8-61 SPI Write/Read Data in CPU Mode**

**DMA Mode****Figure 8-62 SPI Write/Read Data in DMA Mode**

#### 8.4.4.2 Calibrate Delay Chain Using SPI Mode

The SPI has one delay chain which is used to generate delay to make proper timing between the internal SPI clock signal and data signals. Delay chain is made up of 64 delay cells. The delay time of one delay cell can be estimated through delay chain calibration.

The steps to calibrate delay chain are as follows:

- Step 1** Enable SPI. To calibrate the delay chain by operation registers in SPI, the SPI must be enabled through AHB reset and AHB clock gating control registers.
- Step 2** Configure a proper clock for SPI. The calibration delay chain is based on the clock for SPI from CCU.
- Step 3** Set proper initial delay value. Write 0xA0 to the [SPI Sample Delay Control Register](#) to set initial delay value 0x20 to delay chain. Then write 0x0 to the [SPI Sample Delay Control Register](#) to clear this value.
- Step 4** Write 0x8000 to the [SPI Sample Delay Control Register](#) to start to calibrate the delay chain.
- Step 5** Wait until the flag (Bit14 in the [SPI Sample Delay Control Register](#)) of calibration done is set. The number of delay cells is shown at Bit[13:8] in [SPI Sample Delay Control Register](#). The delay time generated by these delay cells is equal to the cycle of SPI's clock nearly. This value is the result of calibration.
- Step 6** Calculate the delay time of one delay cell according to the cycle of the SPI clock and the result of calibration.

#### 8.4.4.3 Transmitting Write Command Using DBI Mode

- Step 1** Set the [SPI DBI MODE SEL](#) (bit3) of [SPI GCR](#) (0x0004) to 1 to select DBI mode.
- Step 2** Set the [DBI EN MODE SEL](#) (bit[30:29]) of [DBI CTL 1](#) (0x0104) to 0 to select the trigger mode of DBI.
- Step 3** Configure the [DBI CTL 0](#) (0x0100).
  - Set [DBI CTL 0](#)[Command Type] (bit31) to 0 to configure the writing command.
  - Set [DBI CTL 0](#)[Write Command Dummy Cycles] (bit[30:20]) to configure the number of dummy cycles between commands.
  - Set [DBI CTL 0](#)[Output Data Sequence] (bit19) to select the MSB or LSB.
  - Set [DBI CTL 0](#)[Transmit Mode] (bit15) to 0 to select the command path.
  - Set [DBI CTL 0](#)[Output Data Format] (bit[14:12]) to 0 to transmit the command.
  - Set [DBI CTL 0](#)[DBI interface Select] (bit[10:8]) to select the DBI interface type.
  - The remaining values of the [DBI CTL 0](#) register remain the default value.

**Step 4** Set [DBI\\_CTL\\_1](#)[DCX\_DATA] (bit22) to 0 to send the command.

**Step 5** DMA Path: Configure the [SPI\\_FCR](#) register (0x0018).

- Set [SPI\\_FCR](#)[TF\_DRQ\_EN] (bit24) to 1 to enable TXFIFO DMA.
- Set [SPI\\_FCR](#)[TX\_TRIG\_LEVEL] (bit[23:16]) to 255. It indicates the controller requests data from DMA if the remaining space of TX FIFO is greater than 255.

CPU Path: Write the command to be sent to the 0x200 address.

**Step 6** Set [SPI\\_GCR](#)[DBI\_EN] (bit4) to 1 to start transmitting the command.

**Step 7** Wait until the TX FIFO underrun interrupt ([SPI\\_ISR](#)[TF\_UDF]) is 1. It indicates that the command written to the TX FIFO is transmitted completely.

#### 8.4.4.4 Transmitting Parameter Using DBI Mode

**Step 1** Set the [SPI\\_DBI\\_MODE\\_SEL](#) (bit3) of [SPI\\_GCR](#) (0x0004) to 1 to select DBI mode.

**Step 2** Set the [DBI\\_EN\\_MODE\\_SEL](#) (bit[30:29]) of [DBI\\_CTL\\_1](#) (0x0104) to 0 to select the trigger mode of DBI.

**Step 3** Configure the [DBI\\_CTL\\_0](#) register (0x0100).

- Set [DBI\\_CTL\\_0](#)[Command Type] (bit31) to 0 to configure the writing command.
- Set [DBI\\_CTL\\_0](#)[Write Command Dummy Cycles] (bit[30:20]) to configure the number of dummy cycles between commands.
- Set [DBI\\_CTL\\_0](#)[Output Data Sequence] (bit19) to select the MSB or LSB.
- Set [DBI\\_CTL\\_0](#)[Transmit Mode] (bit15) to 0 to select the command path.
- Set [DBI\\_CTL\\_0](#)[Output Data Format] (bit[14:12]) to 0 to transmit the command.
- Set [DBI\\_CTL\\_0](#)[DBI interface Select] (bit[10:8]) to select the DBI interface type.
- The remaining values of the [DBI\\_CTL\\_0](#) register remain the default value.

**Step 4** Set [DBI\\_CTL\\_1](#)[DCX\_DATA] (bit22) to 1 to send the parameter.

**Step 5** DMA Path: Configure the [C:\Users\chenxiaofang\AppData\Roaming\Microsoft\Word\Hlk49435465-Hlk49435792](#) register (0x0018).

- Set [SPI\\_FCR](#)[TF\_DRQ\_EN] (bit24) to 1 to enable TXFIFO DMA.
- Set [SPI\\_FCR](#)[TX\_TRIG\_LEVEL] (bit[23:16]) to 255. It indicates the controller requests data from DMA if the remaining space of TX FIFO is greater than 255.

CPU Path: Write the command to be sent to the 0x0200 address.

**Step 6** Set [SPI\\_GCR](#)[DBI\_EN] (bit4) to 1 to start transmitting the command.

**Step 7** Wait until the TX FIFO underrun interrupt ([SPI\\_ISR](#)[TF\_UDF]) is 1. It indicates that the command written to the TX FIFO is transmitted completely.

#### 8.4.4.5 Transmitting Video Using DBI Mode

Set the [SPI\\_DBI\\_MODE\\_SEL](#) (bit3) of [SPI\\_GCR](#) (0x0004) to 1 to select DBI mode.

If the data is from the CPU path, the controller writes the command to be sent to the 0x0200 address by the AHB bus.

If the data is from the DMA path, configure [DBI\\_CTL\\_1](#)[DBI\_FIFO\_DRQ\_EN] (bit15) to 1 and [DBI\\_CTL\\_1](#)[TX\_TRIG\_LEVEL] (bit[14:8]) to 64, which indicates the controller requests data from DMA if the remaining space of TX FIFO is greater than 64.

#### Software Trigger Mode

The software enables DBI\_en\_trigger when the edge interrupt of TE is detected.

After transmitting each frame image, the controller clears automatically the line\_cnt, pixel\_cnt and stops transmitting data.

Wait for the edge interrupt of TE, the software needs to enable DBI\_en\_trigger, in circulation.

The operation process is as follows.

**Step 1** Set the [SPI\\_DBI\\_MODE\\_SEL](#) (bit3) of [SPI\\_GCR](#) (0x0004) to 1 to select DBI mode.

**Step 2** Set the [DBI\\_EN\\_MODE\\_SEL](#) (bit[30:29]) of [DBI\\_CTL\\_1](#) (0x0104) to 1 to select the software trigger mode.

**Step 3** Configure the [DBI\\_CTL\\_0](#) register (0x0100).

- Set [DBI\\_CTL\\_0](#)[Command Type] (bit31) to 0 to set the writing command.
- Set [DBI\\_CTL\\_0](#)[Write Command Dummy Cycles] (bit[30:20]) to configure the number of dummy cycles between commands.
- Set [DBI\\_CTL\\_0](#)[Output Data Sequence] (bit19) to select the MSB or LSB.
- Set [DBI\\_CTL\\_0](#)[Transmit Mode] (bit15) to 1 to select the image path.
- Set [DBI\\_CTL\\_0](#)[Output Data Format] (bit[14:12]) to select RGB111//444/565/666/888.
- Set [DBI\\_CTL\\_0](#)[DBI interface Select] (bit[10:8]) to select the DBI interface type.
- The remaining values of the [DBI\\_CTL\\_0](#) register remain the default value.

**Step 4** Set [DBI\\_CTL\\_1](#)[DCX\_DATA] (bit22) to 0 to send the image data.

**Step 5** Configure [DBI\\_Video\\_Size](#) (0x110) according to the sent image size.

**Step 6** Configure [DBI\\_CTL\\_2](#) (0x0108) to set the TE-related parameter.

**Step 7** Detect the TE interrupt of the [DBI\\_INT](#) (0x0120) register.

**Step 8** Configure [DBI\\_CTL\\_1](#)[DBI\_soft\_trigger] to 1.

## Timer Trigger Mode

The software configures timer\_en to enable timer counting, and when the counter reaches the specified value, the DBI\_EN automatically can be enabled to start transmitting data.

After transmitting each frame image, the controller clears automatically the line\_cnt, pixel\_cnt, and stops transmitting data.

The timer starts counting again. When the counter reaches the specified value, the controller automatically enables DBI\_EN, and in circulation until the software turns off the timer\_en.

The operation process is as follows.

**Step 1** Set the [SPI\\_DBI\\_MODE\\_SEL](#) (bit3) of [SPI\\_GCR](#) (0x0004) to 1 to select DBI mode.

**Step 2** Set the [DBI\\_EN\\_MODE\\_SEL](#) (bit30:29) of [DBI\\_CTL\\_1](#) (0x0104) to 2 to select the timer trigger mode.

**Step 3** Configure the [DBI\\_CTL\\_0](#) register (0x0100).

- Set [DBI\\_CTL\\_0](#)[Command Type] (bit31) to 0 to set the writing command.
- Set [DBI\\_CTL\\_0](#)[Write Command Dummy Cycles] (bit[30:20]) to configure the number of dummy cycles between commands.
- Set [DBI\\_CTL\\_0](#)[Output Data Sequence] (bit19) to select the MSB or LSB.
- Set [DBI\\_CTL\\_0](#)[Transmit Mode] (bit15) to 1 to select the image path.
- Set [DBI\\_CTL\\_0](#)[Output Data Format] (bit[14:12]) to select RGB111/444/565/666/888.
- Set [DBI\\_CTL\\_0](#)[DBI interface Select] (bit[10:8]) to select the DBI interface type.
- The remaining values of the [DBI\\_CTL\\_0](#) register remain the default value.

**Step 4** Set [DBI\\_CTL\\_1](#)[DCX\_DATA] (bit22) to 0 to send the image data.

**Step 5** Configure [DBI\\_Video\\_Size](#) (0x110) to transmit the image size.

**Step 6** Configure the related parameter of [DBI\\_Timer](#) (0x10C).

## TE Trigger Mode

When the edge changes of the TE are detected (The rising and falling edges are optional), the DBI\_EN automatically can be enabled to start transmitting data.

After transmitting each frame image, the controller clears automatically the line\_cnt, pixel\_cnt, and stops transmitting data.

When the edge changes of the TE are detected (The rising and falling edges are optional), the DBI\_EN automatically can be enabled to start transmitting data until the software shuts down TE\_EN or the screen no longer sends TE signals.

The operation process is as follows.

**Step 1** Set the [SPI DBI MODE SEL](#) (bit3) of [SPI GCR](#) (0x0004) to 1 to select DBI mode.

**Step 2** Set the [DBI EN MODE SEL](#) (bit30:29) of [DBI CTL 1](#) (0x0104) to 3 to select the TE trigger mode.

**Step 3** Configure the [DBI CTL 0](#) register (0x0100).

- Set [DBI CTL 0\[Command Type\]](#) (bit31) to 0 to set the writing command.
- Set [DBI CTL 0\[Write Command Dummy Cycles\]](#) (bit[30:20]) to configure the number of dummy cycles between commands.
- Set [DBI CTL 0\[Output Data Sequence\]](#) (bit19) to select the MSB or LSB.
- Set [DBI CTL 0\[Transmit Mode\]](#) (bit15) to 1 to select the image path.
- Set [DBI CTL 0\[Output Data Format\]](#) (bit[14:12]) to select RGB111/444/565/666/888.
- Set [DBI CTL 0\[DBI interface Select\]](#) (bit[10:8]) to select the DBI interface type.
- The remaining values of the [DBI CTL 0](#) register remain the default value.

**Step 4** Configure [DBI CTL 1\[DCX\\_DATA\]](#) (bit22) to 0 to send the image data.

**Step 5** Configure [DBI Video Size](#) (0x0110) to transmit the image size.

**Step 6** Configure [DBI CTL 2](#) (0x0108) to set the TE-related parameter.

#### 8.4.4.6 Transmitting Read Command and Read Data Using DBI Mode

**Step 1** Set the [SPI DBI MODE SEL](#) (bit3) of [SPI GCR](#) (0x0004) to 1 to select DBI mode.

**Step 2** Set the [DBI EN MODE SEL](#) (bit[30:29]) of [DBI CTL 1](#) (0x0104) to 0.

**Step 3** Configure the [DBI CTL 0](#) register (0x0100).

- Set [DBI CTL 0\[Command Type\]](#) (bit31) to 0 to set the reading command.
- Set [DBI CTL 0\[Output Data Sequence\]](#) (bit19) to select the MSB or LSB.
- Set [DBI CTL 0\[Transmit Mode\]](#) (bit15) to 0 to select the command path.
- Set [DBI CTL 0\[Output Data Format\]](#) (bit[14:12]) to 0.
- Set [DBI CTL 0\[DBI interface Select\]](#) (bit[10:8]) to select the DBI interface type.

- The remaining values of the [DBI\\_CTL\\_0](#) register remain the default value.

**Step 4** Configure the [DBI\\_CTL\\_1](#) register (0x0104).

- Configure [DBI\\_CTL\\_1](#)[DCX\_DATA] (bit22) to 0 to send the command.
- Configure [DBI\\_CTL\\_1](#)[Read\_MSB\_First] (bit20) to select whether the first bit of the read data is the highest or lowest bit of data.
- Configure [DBI\\_CTL\\_1](#)[Read Data Number of Bytes] to set the byte number to be read.
- Configure [DBI\\_CTL\\_1](#)[Read Command Dummy Cycles] to set the dummy cycle between the read command and the read data, when the dummy cycle is complete, the data starts to be sampled.

**Step 5** DMA Path: Configure the [SPI\\_FCR](#) register (0x0018).

- Set [SPI\\_FCR](#)[RF\_DRQ\_EN] (bit8) to 1 to enable RXFIFO DMA.
- Set [SPI\\_FCR](#)[RX\_TRIG\_LEVEL] (bit[7:0]) to 32, which indicates the controller requests receiving data from DMA if the data of the RX FIFO is greater than 64.

CPU Path: Read data in RX FIFO from the 0x0300 address.

**Step 6** Set [SPI\\_GCR](#)[DBI\_EN] (bit4) to 1 to start transmitting command.

**Step 7** Wait until [DBI\\_INT](#)[RD\_DONE\_INT] is 1. It indicates that the data is read completely.

#### 8.4.5 Register List

Module Name	Base Address
SPI_DBI	0x04026000

Register Name	Offset	Description
SPI_GCR	0x0004	SPI Global Control Register
SPI_TCR	0x0008	SPI Transfer Control Register
SPI_IER	0x0010	SPI Interrupt Control Register
SPI_ISR	0x0014	SPI Interrupt Status Register
SPI_FCR	0x0018	SPI FIFO Control Register
SPI_FSR	0x001C	SPI FIFO Status Register
SPI_WCR	0x0020	SPI Wait Clock Register
SPI_SAMP_DL	0x0028	SPI Sample Delay Control Register
SPI_MBC	0x0030	SPI Master Burst Counter Register
SPI_MTC	0x0034	SPI Master Transmit Counter Register

Register Name	Offset	Description
SPI_BCC	0x0038	SPI Master Burst Control Register
SPI_BATCR	0x0040	SPI Bit-Aligned Transfer Configure Register
SPI_BA_CCR	0x0044	SPI Bit-Aligned Clock Configuration Register
SPI_TBR	0x0048	SPI TX Bit Register
SPI_RBR	0x004C	SPI RX Bit Register
SPI_NDMA_MODE_CTL	0x0088	SPI Normal DMA Mode Control Register
DBI_CTL_0	0x0100	DBI Control Register 0
DBI_CTL_1	0x0104	DBI Control Register 1
DBI_CTL_2	0x0108	DBI Control Register 2
DBI_TIMER	0x010C	DBI Timer Control Register
DBI_VIDEO_SZIE	0x0110	DBI Video Size Configuration Register
DBI_INT	0x0120	DBI Interrupt Register
DBI_DEBUG_0	0x0124	DBI BEBUG 0 Register
DBI_DEBUG_1	0x0128	DBI BEBUG 1 Register
SPI_TXD	0x0200	SPI TX Data register
SPI_RXD	0x0300	SPI RX Data register

#### 8.4.6 Register Description

##### 8.4.6.1 0x0004 SPI Global Control Register (Default Value: 0x0000\_0080)

Offset:0x0004			Register Name: SPI_GCR
Bit	Read/Write	Default/Hex	Description
31	R/WAC	0x0	<p>SRST Soft reset</p> <p>Writing '1' to this bit will clear the SPI controller, and auto clear to '0' when reset operation completes.</p> <p>Writing '0' to this bit has no effect.</p>
30:8	/	/	/

Offset:0x0004			Register Name: SPI_GCR
Bit	Read/Write	Default/Hex	Description
7	R/W	0x1	<p>TP_EN Transmit Pause Enable In master mode, it is used to control transmit state machine to stop smart burst sending when RX FIFO is full. 0: Normal operation, ignore RXFIFO status 1: Stop transmit data when RXFIFO full Cannot be written when XCH=1.</p>
6:5	/	/	/
4	R/W	0x0	<p>DBI EN DBI Module Enable Control 0: Disable 1: Enable</p>
3	R/W	0x0	<p>SPI_DBI_MODE_SEL DBI Working Mode Select 0: SPI MODE 1: DBI MODE</p>
2	R/W	0x0	<p>MODE_SELEC Sample Timing Mode Select 0: Old mode of Sample Timing 1: New mode of Sample Timing Cannot be written when XCH=1.</p>
1	R/W	0x0	<p>MODE SPI Function Mode Select 0: Slave mode 1: Master mode Cannot be written when XCH=1.</p>
0	R/W	0x0	<p>EN SPI Module Enable Control 0: Disable 1: Enable After transforming from bit_mode to byte_mode, it must enable the SPI module again.</p>

#### 8.4.6.2 0x0008 SPI Transfer Control Register (Default Value: 0x0000\_0087)

Offset: 0x0008			Register Name: SPI_TCR
Bit	Read/Write	Default/Hex	Description
31	R/WAC	0x0	<p>XCH Exchange Burst In master mode, it is used to start SPI burst 0: Idle 1: Initiates exchange. Writing “1” to this bit will start the SPI burst, and will auto-clear after finishing the bursts transfer specified by BC. Writing “1” to SRST will also clear this bit. Writing ‘0’ to this bit has no effect. Cannot be written when XCH=1.</p>
30:16	/	/	/
15	R/W	0x0	<p>SDC1 Master Sample Data Control register1 Set this bit to ‘1’ to make the internal read sample point with a delay of half-cycle of SPI_CLK. It is used in high speed read operation to reduce the error caused by the time delay of SPI_CLK between master and slave. 0: Normal operation, do not delay the internal read sample point 1: Delay the internal read sample point Cannot be written when XCH=1.</p>
14	R/W	0x0	<p>SDDM Sending Data Delay Mode 0: Normal sending 1: Delay sending Set the bit to "1" to make the data that should be sent with a delay of half-cycle of SPI_CLK in dual IO mode for SPI mode 0. Cannot be written when XCH=1.</p>
13	R/W	0x0	<p>SDM Master Sample Data Mode 0: Delay sample mode 1: Normal sample mode In normal sample mode, the SPI master samples the data at the correct edge for each SPI mode; In delay sample mode, the SPI master samples data at the edge that is half cycle delayed by the correct edge defined in respective SPI mode. Cannot be written when XCH=1.</p>

Offset: 0x0008			Register Name: SPI_TCR
Bit	Read/Write	Default/Hex	Description
12	R/W	0x0	<p>FBS First Transmit Bit Select 0: MSB first. The upper bits are transmitted first. 1: LSB first. The lower bits are transmitted first. Cannot be written when XCH=1.</p>
11	R/W	0x0	<p>SDC Master Sample Data Control Set this bit to '1' to make the internal read sample point with a delay of half-cycle of SPI_CLK. It is used in high speed read operation to reduce the error caused by the time delay of SPI_CLK between master and slave. 0: Normal operation, do not delay the internal read sample point 1: Delay the internal read sample point Cannot be written when XCH=1.</p>
10	R/W	0x0	<p>RPSM Rapids Mode Select Select rapid mode for high speed write. 0: Normal write mode 1: Rapid write mode Cannot be written when XCH=1.</p>
9	R/W	0x0	<p>DDB Dummy Burst Type 0: The bit value of dummy SPI burst is zero 1: The bit value of dummy SPI burst is one Cannot be written when XCH=1.</p>
8	R/W	0x0	<p>DHB Discard Hash Burst In master mode, it controls whether discarding unused SPI bursts 0: Receiving all SPI bursts in the BC period 1: Discard unused SPI bursts, only fetching the SPI bursts during the dummy burst period. The burst number is specified by TC. Cannot be written when XCH=1.</p>

Offset: 0x0008			Register Name: SPI_TCR
Bit	Read/Write	Default/Hex	Description
7	R/W	0x1	<p>SS_LEVEL SPI Chip Select Level When control SS signal manually (SS_OWNER (SPI_TCR[6])=1), set this bit to '1' or '0' to control the level of SS signal. 0: Set SS to low 1: Set SS to high Cannot be written when XCH=1.</p>
6	R/W	0x0	<p>SS_OWNER SS Output Owner Select Usually, the controller sends the SS signal automatically with data together. When this bit is set to 1, the software must manually write SS_LEVEL (SPI_TCR[7]) to 1 or 0 to control the level of the SS signal. 0: SPI controller 1: Software Cannot be written when XCH=1.</p>
5:4	R/W	0x0	<p>SS_SEL SPI Chip Select Select one of four external SPI Master/Slave Devices 00: SPI_SS0 will be asserted 01: SPI_SS1 will be asserted 10: SPI_SS2 will be asserted 11: SPI_SS3 will be asserted Cannot be written when XCH=1.</p>
3	R/W	0x0	<p>SSCTL In master mode, this bit selects the output waveform for the SPI_SSx signal. Only valid when SS_OWNER (SPI_TCR[6])=0. 0: SPI_SSx remains asserted between SPI bursts 1: Negate SPI_SSx between SPI bursts Cannot be written when XCH=1.</p>
2	R/W	0x1	<p>SPOL SPI Chip Select Signal Polarity Control 0: Active high polarity (0 = Idle) 1: Active low polarity (1 = Idle) Cannot be written when XCH=1.</p>

Offset: 0x0008			Register Name: SPI_TCR
Bit	Read/Write	Default/Hex	Description
1	R/W	0x1	CPOL SPI Clock Polarity Control 0: Active high polarity (0 = Idle) 1: Active low polarity (1 = Idle) Cannot be written when XCH=1.
0	R/W	0x1	CPHA SPI Clock/Data Phase Control 0: Phase 0 (Leading edge for sample data) 1: Phase 1 (Leading edge for setup data) Cannot be written when XCH=1.

#### 8.4.6.3 0x0010 SPI Interrupt Control Register (Default Value: 0x0000\_0000)

Offset: 0x0010			Register Name: SPI_IER
Bit	Read/Write	Default/Hex	Description
31:14	/	/	/
13	R/W	0x0	SS_INT_EN SSI Interrupt Enable Chip select signal (SSx) from the valid state to the invalid state 0: Disable 1: Enable
12	R/W	0x0	TC_INT_EN Transfer Completed Interrupt Enable 0: Disable 1: Enable
11	R/W	0x0	TF_UDR_INT_EN TXFIFO Underrun Interrupt Enable 0: Disable 1: Enable
10	R/W	0x0	TF_OVF_INT_EN TX FIFO Overflow Interrupt Enable 0: Disable 1: Enable

Offset: 0x0010			Register Name: SPI_IER
Bit	Read/Write	Default/Hex	Description
9	R/W	0x0	RF_UDR_INT_EN RX FIFO Underrun Interrupt Enable 0: Disable 1: Enable
8	R/W	0x0	RF_OVF_INT_EN RX FIFO Overflow Interrupt Enable 0: Disable 1: Enable
7	/	/	/
6	R/W	0x0	TF_FUL_INT_EN TX FIFO Full Interrupt Enable 0: Disable 1: Enable
5	R/W	0x0	TX_EMP_INT_EN TX FIFO Empty Interrupt Enable 0: Disable 1: Enable
4	R/W	0x0	TX_ERQ_INT_EN TX FIFO Empty Request Interrupt Enable 0: Disable 1: Enable
3	/	/	/
2	R/W	0x0	RF_FUL_INT_EN RX FIFO Full Interrupt Enable 0: Disable 1: Enable
1	R/W	0x0	RX_EMP_INT_EN RX FIFO Empty Interrupt Enable 0: Disable 1: Enable
0	R/W	0x0	RF_RDY_INT_EN RX FIFO Ready Request Interrupt Enable 0: Disable 1: Enable

#### 8.4.6.4 0x0014 SPI Interrupt Status Register (Default Value: 0x0000\_0032)

Offset: 0x0014			Register Name: SPI_ISR
Bit	Read/Write	Default/Hex	Description
31:14	/	/	/
13	R/W1C	0x0	<p>SSI SS Invalid Interrupt When SSI is 1, it indicates that SS has changed from the valid state to the invalid state. Writing 1 to this bit clears it.</p>
12	R/W1C	0x0	<p>TC Transfer Completed In master mode, it indicates that all bursts specified by BC have been exchanged. In other conditions, when set, this bit indicates that all the data in TXFIFO has been loaded in the Shift register, and the Shift register has shifted out all the bits. Writing 1 to this bit clears it. 0: Busy 1: Transfer completed</p>
11	R/W1C	0x0	<p>TF_UDF TXFIFO Underrun This bit is set when the TXFIFO is underrun. Writing 1 to this bit clears it. 0: TXFIFO is not underrun 1: TXFIFO is underrun</p>
10	R/W1C	0x0	<p>TF_OVF TXFIFO Overflow This bit is set when the TXFIFO is overflowed. Writing 1 to this bit clears it. 0: TXFIFO is not overflowed 1: TXFIFO is overflowed</p>
9	R/W1C	0x0	<p>RX_UDF RXFIFO Underrun When set, this bit indicates that RXFIFO is underrun. Writing 1 to this bit clears it. 0: RXFIFO is not underrun 1: RXFIFO is underrun</p>

Offset: 0x0014			Register Name: SPI_ISR
Bit	Read/Write	Default/Hex	Description
8	R/W1C	0x0	<p>RX_OVF RXFIFO Overflow When set, this bit indicates that RXFIFO is overflowed. Writing 1 to this bit clears it. 0: RXFIFO is not overflowed 1: RXFIFO is overflowed</p>
7	/	/	/
6	R/W1C	0x0	<p>TX_FULL TXFIFO Full This bit is set when the TXFIFO is full. Writing 1 to this bit clears it. 0: TXFIFO is not Full 1: TXFIFO is Full</p>
5	R/W1C	0x1	<p>TX_EMP TXFIFO Empty This bit is set when the TXFIFO is empty. Writing 1 to this bit clears it. 0: TXFIFO contains one or more words. 1: TXFIFO is empty</p>
4	R/W1C	0x1	<p>TX_READY TXFIFO Ready 0: TX_WL &gt; TX_TRIG_LEVEL 1: TX_WL &lt;= TX_TRIG_LEVEL This bit will be immediately set to 1 if TX_WL &lt;= TX_TRIG_LEVEL. Writing "1" to this bit clears it. The TX_WL is the water level of TXFIFO.</p>
3	/	/	/
2	R/W1C	0x0	<p>RX_FULL RXFIFO Full This bit is set when the RXFIFO is full. Writing 1 to this bit clears it. 0: Not Full 1: Full</p>
1	R/W1C	0x1	<p>RX_EMP RXFIFO Empty This bit is set when the RXFIFO is empty. Writing 1 to this bit clears it. 0: Not empty 1: empty</p>

Offset: 0x0014			Register Name: SPI_ISR
Bit	Read/Write	Default/Hex	Description
0	R/W1C	0x0	<p>RX_RDY RXFIFO Ready 0: RX_WL &lt; RX_TRIG_LEVEL 1: RX_WL &gt;= RX_TRIG_LEVEL</p> <p>This bit will be immediately set to 1 if RX_WL &gt;= RX_TRIG_LEVEL. Writing "1" to this bit clears it. The RX_WL is the water level of RXFIFO.</p>

#### 8.4.6.5 0x0018 SPI FIFO Control Register (Default Value: 0x0040\_0001)

Offset: 0x0018			Register Name: SPI_FCR
Bit	Read/Write	Default/Hex	Description
31	R/WAC	0x0	<p>TX_FIFO_RST TX FIFO Reset Writing '1' to this bit will reset the control portion of the TXFIFO and auto clear to '0' when completing the reset operation, writing '0' to this bit has no effect.</p>
30	R/W	0x0	<p>TF_TEST_ENB TX Test Mode Enable 0: Disable 1: Enable In normal mode, the TXFIFO can only be read by the SPI controller, writing '1' to this bit will switch the read and write function of TXFIFO to AHB bus. This bit is used to test the TXFIFO, do not set in normal operation, and do not set RF_TEST and TF_TEST at the same time.</p>
29:25	/	/	/
24	R/W	0x0	<p>TF_DRQ_EN TX FIFO DMA Request Enable 0: Disable 1: Enable</p>
23:16	R/W	0x40	<p>TX_TRIG_LEVEL TX FIFO Empty Request Trigger Level</p>

Offset: 0x0018			Register Name: SPI_FCR
Bit	Read/Write	Default/Hex	Description
15	R/WAC	0x0	RF_RST RXFIFO Reset Writing '1' to this bit will reset the control portion of the receiver FIFO, and auto clear to '0' when completing the reset operation, writing '0' to this bit has no effect.
14	R/W	0x0	RF_TEST RX Test Mode Enable 0: Disable 1: Enable In normal mode, the RXFIFO can only be written by the SPI controller, writing '1' to this bit will switch the read and write function of RXFIFO to AHB bus. This bit is used to test the RXFIFO, do not set in normal operation, and do not set RF_TEST and TF_TEST at the same time.
13:9	/	/	/
8	R/W	0x0	RF_DRQ_EN RXFIFO DMA Request Enable 0: Disable 1: Enable
7:0	R/W	0x1	RX_TRIG_LEVEL RXFIFO Ready Request Trigger Level

#### 8.4.6.6 0x001C SPI FIFO Status Register (Default Value: 0x0000\_0000)

Offset: 0x001C			Register Name: SPI_FSR
Bit	Read/Write	Default/Hex	Description
31	R	0x0	TB_WR TXFIFO Write Buffer Write Enable
30:28	R	0x0	TB_CNT TXFIFO Write Buffer Counter These bits indicate the number of words in TXFIFO Write Buffer
27:24	/	/	/

Offset: 0x001C			Register Name: SPI_FSR
Bit	Read/Write	Default/Hex	Description
23:16	R	0x0	<p>TF_CNT TXFIFO Counter These bits indicate the number of words in TXFIFO 0: 0 byte in TXFIFO 1: 1 byte in TXFIFO ... 64: 64 bytes in TXFIFO other: Reserved</p>
15	R	0x0	<p>RB_WR RXFIFO Read Buffer Write Enable</p>
14:12	R	0x0	<p>RB_CNT RXFIFO Read Buffer Counter These bits indicate the number of words in RXFIFO Read Buffer</p>
11:8	/	/	/
7:0	R	0x0	<p>RF_CNT RXFIFO Counter These bits indicate the number of words in RXFIFO 0: 0 byte in RXFIFO 1: 1 byte in RXFIFO ... 64: 64 bytes in RXFIFO other: Reserved</p>

#### 8.4.6.7 0x0020 SPI Wait Clock Register (Default Value: 0x0000\_0000)

Offset: 0x0020			Register Name: SPI_WCR
Bit	Read/Write	Default/Hex	Description
31:20	/	/	/

Offset: 0x0020			Register Name: SPI_WCR
Bit	Read/Write	Default/Hex	Description
19:16	R/W	0x0	<p>SWC Dual mode direction switch wait clock counter (for master mode only).</p> <p>These bits control the number of wait states to be inserted before starting dual data transfer in dual SPI mode. The SPI module counts SPI_SCLK by SWC for delaying the next word data transfer.</p> <p>0: No wait states inserted n: n SPI_SCLK wait states inserted Cannot be written when XCH=1.</p>
15:0	R/W	0x0	<p>WCC Wait Clock Counter (In master mode)</p> <p>These bits control the number of wait states to be inserted in data transfers. The SPI module counts SPI_SCLK by WCC for delaying the next word data transfer.</p> <p>0: No wait states inserted n: n SPI_SCLK wait states inserted Cannot be written when XCH=1.</p>

#### 8.4.6.8 0x0028 SPI Sample Delay Control Register (Default Value: 0x0000\_2000)

Offset: 0x0028			Register Name: SPI_SAMP_DL
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15	R/W	0x0	<p>SAMP_DL_CAL_START Sample Delay Calibration Start When set, it indicates that start sample delay chain calibration. Cannot be written when XCH=1.</p>
14	R	0x0	<p>SAMP_DL_CAL_DONE Sample Delay Calibration Done When set, it indicates that the sample delay chain calibration is done and the result of calibration is shown in SAMP_DL. Cannot be written when XCH=1.</p>

Offset: 0x0028			Register Name: SPI_SAMP_DL
Bit	Read/Write	Default/Hex	Description
13:8	R	0x20	<p>SAMP_DL Sample Delay</p> <p>It indicates the number of delay cells corresponding to the current card clock. The delay time generated by these delay cells is equal to the cycle of the card clock nearly.</p> <p>Generally, it is necessary to do drive delay calibration when the card clock is changed.</p> <p>This bit is valid only when SAMP_DL_CAL_DONE is set.</p> <p>Cannot be written when XCH=1.</p>
7	R/W	0x0	<p>SAMP_DL_SW_EN Sample Delay Software Enable</p> <p>When set, it indicates that enable sample delay specified at SAMP_DL_SW.</p> <p>Cannot be written when XCH=1.</p>
6	/	/	/
5:0	R/W	0x0	<p>SAMP_DL_SW Sample Delay Software</p> <p>The relative delay between the clock line and command line, data lines.</p> <p>It can be determined according to the value of SAMP_DL, the cycle of the card clock, and the input timing requirement of the device.</p> <p>Cannot be written when XCH=1.</p>

#### 8.4.6.9 0x0030 SPI Master Burst Counter Register (Default Value: 0x0000\_0000)

Offset: 0x0030			Register Name: SPI_MBC
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/

Offset: 0x0030			Register Name: SPI_MBC
Bit	Read/Write	Default/Hex	Description
23:0	R/W	0x0	<p>MBC Master Burst Counter In master mode, this field specifies the total burst number which includes the TXD, RXD, and dummy burst.</p> <p>0: 0 burst 1: 1 burst ... N: N bursts</p> <p>Cannot be written when XCH=1.</p>

#### 8.4.6.10 0x0034 SPI Master Transmit Counter Register (Default Value: 0x0000\_0000)

Offset: 0x0034			Register Name: SPI_MTC
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23:0	R/W	0x0	<p>MWTC Master Write Transmit Counter In master mode, this field specifies the burst number that should be sent to TXFIFO before automatically sending dummy bursts. For saving bus bandwidth, the dummy bursts (all zero bits or all one bits) are sent by SPI Controller automatically.</p> <p>0: 0 burst 1: 1 burst ... N: N bursts</p> <p>Cannot be written when XCH=1.</p>

#### 8.4.6.11 0x0038 SPI Master Burst Control Counter Register (Default Value: 0x0000\_0000)

Offset: 0x0038			Register Name: SPI_BCC
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/

Offset: 0x0038			Register Name: SPI_BCC
Bit	Read/Write	Default/Hex	Description
29	R/W	0x0	<p>Quad_EN Quad_Mode_EN The Quad mode includes Quad-Input and Quad-Output. 0: Quad mode disable 1: Quad mode enable Cannot be written when XCH=1.</p>
28	R/W	0x0	<p>DRM Master Dual Mode RX Enable It is only valid when Quad_Mode_EN=0. 0: RX uses the single-bit mode 1: RX uses the dual mode Cannot be written when XCH=1.</p>
27:24	R/W	0x0	<p>DBC Master Dummy Burst Counter In master mode, this field specifies the burst number that should be sent before receiving in dual SPI mode. The data does not care by the device. 0: 0 burst 1: 1 burst ... N: N bursts Cannot be written when XCH=1</p>
23:0	R/W	0x0	<p>STC Master Single Mode Transmit Counter In master mode, this field specifies the burst number that should be sent in the single mode before automatically sending dummy bursts. This is the first transmit counter in all bursts. 0: 0 burst 1: 1 burst ... N: N bursts Cannot be written when XCH=1</p>

**8.4.6.12 0x0040 SPI Bit-Aligned Transfer Configure Register (Default Value: 0x0000\_00A0)**

Offset: 0x0040			Register Name: SPI_BATC
Bit	Read/Write	Default/Hex	Description
31	R/WAC	0x0	<p>TCE Transfer Control Enable In master mode, it is used to start to transfer the serial bits frame, it is only valid when Work Mode Select==0x10/0x11. 0: Idle 1: Initiates transfer Writing “1” to this bit will start to transfer serial bits frame (the value comes from the SPI TX Bit Register or SPI RX Bit Register), and will auto-clear after the bursts transfer completely. Writing ‘0’ to this bit has no effect.</p>
30	R/W	0x0	<p>MSMS Master Sample Standard 0: Delay Sample Mode 1: Standard Sample Mode In Standard Sample Mode, the SPI master samples the data at the standard rising edge of SCLK for each SPI mode; In Delay Sample Mode, the SPI master samples data at the edge that is half cycle delayed by the standard rising edge of SCLK defined in respective SPI mode.</p>
29:26	/	/	/
25	R/W1C	0x0	<p>TBC Transfer Bits Completed When set, this bit indicates that the last bit of the serial data frame in SPI TX Bit Register (or SPI RX Bit Register) has been transferred completely. Writing 1 to this bit clears it. 0: Busy 1: Transfer Completed It is only valid when Work Mode Select==0x10/0x11.</p>
24	R/W	0x0	<p>TBC_INT_EN Transfer Bits Completed Interrupt Enable 0: Disable 1: Enable It is only valid when Work Mode Select==0x10/0x11.</p>
23:22	/	/	/

Offset: 0x0040			Register Name: SPI_BATC
Bit	Read/Write	Default/Hex	Description
21:16	R/W	0x00	<p>RX_FEM_LEN</p> <p>Configure the length of serial data frame (burst) of RX</p> <p>000000: 0 bit</p> <p>000001: 1 bit</p> <p>...</p> <p>100000: 32 bits</p> <p>Other values: reserved</p> <p>It is only valid when Work Mode Select==0x10/0x11, and cannot be written when TCE (SPI_BATC[31])=1.</p>
15:14	/	/	/
13:8	R/W	0x00	<p>TX_FEM_LEN</p> <p>Configure the length of serial data frame (burst) of TX</p> <p>000000: 0 bit</p> <p>000001: 1 bit</p> <p>...</p> <p>100000: 32 bits</p> <p>Other values: reserved</p> <p>It is only valid when Work Mode Select==0x10/0x11, and cannot be written when TCE=1.</p>
7	R/W	0x1	<p>SS_LEVEL</p> <p>When control SS signal manually, set this bit to '1' or '0' to control the level of SS signal.</p> <p>0: Set SS to low</p> <p>1: Set SS to high</p> <p>It is only valid when Work Mode Select==0x10/0x11, and only work in Mode0, cannot be written when TCE=1.</p>
6	R/W	0x0	<p>SS_OWNER</p> <p>SS Output Owner Select</p> <p>Usually, the controller sends the SS signal automatically with data together. When this bit is set to 1, the software must manually write SS_LEVEL (SPI_BATC[7]) to 1 or 0 to control the level of the SS signal.</p> <p>0: SPI controller</p> <p>1: Software</p> <p>It is only valid when Work Mode Select==0x10/0x11, and only work in Mode0, cannot be written when TCE=1.</p>

Offset: 0x0040			Register Name: SPI_BATC
Bit	Read/Write	Default/Hex	Description
5	R/W	0x1	<p>SPOL SPI Chip Select Signal Polarity Control 0: Active high polarity (0 = Idle) 1: Active low polarity (1 = Idle) It is only valid when Work Mode Select==0x10/0x11, and only work in Mode0, cannot be written when TCE=1.</p>
4	/	/	/
3:2	R/W	0x0	<p>SS_SEL SPI Chip Select Select one of four external SPI Master/Slave Devices 00: SPI_SS0 will be asserted 01: SPI_SS1 will be asserted 10: SPI_SS2 will be asserted 11: SPI_SS3 will be asserted It is only valid when Work Mode Select==0x10/0x11, and only work in Mode0, cannot be written when TCE=1.</p>
1:0	R/W	0x0	<p>WMS Work Mode Select 00: Data frame is byte aligned in standard SPI, dual-output/dual input SPI, dual IO SPI, and quad-output/quad-input SPI 01: Reserved 10: Data frame is bit aligned in 3-wire SPI 11: Data frame is bit aligned in standard SPI</p>

#### 8.4.6.13 0x0044 SPI Bit-Aligned Clock Configuration Register (Default Value: 0x0000\_0000)

Offset: 0x0044			Register Name: SPI_BA_CCR
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R/W	0x0	<p>CDR_N Clock Divide Rate (Master Mode Only) The SPI_SCLK is determined according to the following equation: <math>SPI\_CLK = Source\_CLK / (2 * (CDR\_N + 1))</math>. This register is only valid when Work Mode Select==0x10/0x11.</p>

**8.4.6.14 0x0048 SPI TX Bit Register (Default Value: 0x0000\_0000)**

Offset: 0x0048			Register Name: SPI_TBR
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<p>VTB The Value of the Transmit Bits This register is used to store the value of the transmitted serial data frame. In the process of transmission, the LSB is transmitted first. This register is only valid when Work Mode Select==0x10/0x11.</p>

**8.4.6.15 0x004C SPI RX Bit Register (Default Value: 0x0000\_0000)**

Offset: 0x004C			Register Name: SPI_RBR
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<p>VRB The Value of the Receive Bits This register is used to store the value of the received serial data frame. In the process of transmission, the LSB is transmitted first. This register is only valid when Work Mode Select==0x10/0x11.</p>

**8.4.6.16 0x0088 SPI Normal DMA Mode Control Register (Default Value: 0x0000\_00E5)**

Offset: 0x0088			Register Name: SPI_NDMA_MODE_CTL
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:6	R/W	0x11	<p>SPI_ACT_M SPI NDMA Active Mode 00: dma_active is low 01: dma_active is high 10: dma_active is controlled by dma_request (DRQ) 11: dma_active is controlled by controller</p>
5	R/W	0x1	<p>SPI_ACK_M SPI NDMA Acknowledge Mode 0: active fall do not care ack 1: active fall must after detect ack is high</p>

Offset: 0x0088			Register Name: SPI_NDMA_MODE_CTL
Bit	Read/Write	Default/Hex	Description
4:0	R/W	0x05	SPI_DMA_WAIT Delay Cycles The counts of hold cycles from DMA last signal high to dma_active high

#### 8.4.6.17 0x0100 DBI Control Register 0 (Default Value: 0x0010\_0000)

Offset: 0x0100			Register Name: DBI_CTL_0
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	CMDT Command Type 0: Write Command 1: Read Command
30:20	R/W	0x1	WCDC Write Command Dummy Cycles Controls dummy cycles between two write commands Range 1~255 Default Condition: there is a dbi_clk cycle between each command or parameter.
19	R/W	0x0	DAT_SEQ Output Data Sequence 0: MSB First 1: LSB First
18:16	R/W	0x0	RGB_SEQ Output RGB Sequence 000: RGB 001: RBG 010: GRB 011: GBR 100: BRG 101: BGR 110, 111: Reserved

Offset: 0x0100			Register Name: DBI_CTL_0
Bit	Read/Write	Default/Hex	Description
15	R/W	0x0	TRAN_MOD Transmit Mode 0: Command/Parameter 1: Video
14:12	R/W	0x0	DAT_FMT Output Data Format 000: RGB111 001: RGB444 010: RGB565 011: RGB666 100: RGB888 (only for 2 Data Lane Interface) 101~111: Reserved
11	/	/	/
10:8	R/W	0x0	DBI Interface 000: 3 Line Interface I 001: 3 Line Interface II 010: 4 Line Interface I 011: 4 Line Interface II 100: 2 Data Lane Interface

Offset: 0x0100			Register Name: DBI_CTL_0
Bit	Read/Write	Default/Hex	Description
7:4	R/W	0x0	<p>RGB_Source_Format                      When video_source_type is RGB32 (DBI_CTL_0[bit0] = 0)                      0000: RGB                      0001: RBG                      0010: GRB                      0011: GBR                      0100: BRG                      0101: BGR                      Others: Reserved</p> <p>When video_source_type is RGB16 (DBI_CTL_0[bit0] = 1)                      0000: RGB                      0001~0100: Reserved                      0101: BGR                      0110: GRBG_0 {G[5:3]R[4:0]B[4:0]G[2:0]}                      0111: GBRG_0 {G[5:3]B[4:0]R[4:0]G[2:0]}                      1000: GRBG_1 {G[2:0]R[4:0]B[4:0]G[5:3]}                      1001: GBRG_1 {G[2:0]B[4:0]R[4:0]G[5:3]}                      Others: Reserved</p>
3	R/W	0x0	<p>DUM_VAL                      Dummy Cycle Value                      Output Value During Dummy Cycle</p>
2	R/W	0x0	<p>RGB_BO                      RGB Bit Order                      0: Remain the sequence of RGB data                      1: Swap the higher bit and the lower bit for each component of DRAM RGB</p>
1	R/W	0x0	<p>ELEMENT_A_POS                      Element A Position                      Only for RGB32 Data Format                      0: A component is in the bit[31:24] of data source                      1: A component is in the bit[7:0] of data source</p>
0	R/W	0x0	<p>VI_SRC_TYPE                      Video Source Type                      0: RGB32                      1: RGB16</p>

**8.4.6.18 0x0104 DBI Control Register 1 (Default Value: 0x0000\_0001)**

Offset: 0x0104			Register Name: DBI_CTL_1
Bit	Read/Write	Default/Hex	Description
31	R/WAC	0x0	<p>DBI_SOFT_TRG DBI soft trigger</p> <p>It is only available for software trigger mode. Writing '1' to this bit will start DBI TX module and auto clear to '0' when completing start operation, writing '0' to this bit has no effect.</p>
30:29	R/W	0x0	<p>DBI EN MODE SEL DBI Enable Mode Select</p> <p>00: Always on DBI mode 01: Software trigger mode 10: Timer trigger mode 11: TE trigger mode</p>
28	/	/	/
27:26	R/W	0x0	<p>RGB666_FMT 2 Data Lane RGB666 Format</p> <p>00: Normal Format 01: Special Format for ILITEK 10: Special Format for New Vision</p>
25	R/W	0x0	<p>DBI_RXCLK_INV DBI rx clock inverse</p> <p>0: Sample data by using the positive edge of the output clock 1: Sample data by using the negative edge of the output clock</p>
24	R/W	0x0	<p>DBI_CLKO_MOD DBI output clock mode</p> <p>0: DBI clock always on (DCX Setup/hold equals one clock cycle) 1: DBI clock auto gating (DCX Setup/hold equals to a half clock cycle)</p>

Offset: 0x0104			Register Name: DBI_CTL_1
Bit	Read/Write	Default/Hex	Description
23	R/W	0x0	<p>DBI_CLKO_INV DBI clock output inverse</p> <p><b>When the bit24 (DBI output clock mode) is 0.</b></p> <p>0: The falling edge releases the CSX signal, and the falling edge releases data</p> <p>1: The rising edge releases the CSX signal, and the rising edge releases data</p> <p><b>When the bit24 (DBI output clock mode) is 1.</b></p> <p>0: The rising edge releases the CSX signal, and the falling edge releases data</p> <p>1: The falling edge releases the CSX signal, and the rising edge releases data</p>
22	R/W	0x0	<p>DCX_DATA DCX Data Value</p> <p>0: DCX Value equal to 0</p> <p>1: DCX Value equal to 1</p>
21	R/W	0x0	<p>RGB 16 Data Source Select RGB 16 Data Source Select</p> <p>0: Pixel1 is stored in the higher bit of address, and Pixel0 is stored in the lower bit of address</p> <p>1: Pixel0 is stored in the higher bit of address, and Pixel1 is stored in the lower bit of address</p>
20	R/W	0x0	<p>RDAT_LSB Bit Order of Read Data</p> <p>0: A reading data is the higher bit</p> <p>1: A reading data is the lower bit</p>
19:16	/	/	/
15:8	R/W	0x0	<p>RCDC Read Command Dummy Cycles</p> <p>The dummy cycle between the read command and read data</p> <p>Reading 1-byte (8 bits) data has not dummy cycle.</p>
7:0	R/W	0x1	<p>RDBN Read Data Number of Bytes</p> <p>Sample Bytes data based on configuration.</p>

## 8.4.6.19 0x0108 DBI Control Register 2 (Default Value: 0x0000\_4000)

Offset: 0x0108			Register Name: DBI_CTL_2
Bit	Read/Write	Default/Hex	Description
31:26	/	/	/
15	R/W	0x0	DBI_FIFO_DRQ_EN DBI FIFO DMA Request Enable 0: Disable 1: Enable
14:8	R/W	0x40	DBI_TRIG_LEVEL DBI FIFO Empty Request Trigger Level
7	/	/	/
6	R/W	0x0	DBI_SDI_OUT_SEL DBI SDI PIN Output Select The signal is used with the DBI SDI PIN Function Sel bit. 0: Output WRX (When DBI DCX PIN Function Sel = 0, the SDI pin outputs data) 1: Output DCX
5	R/W	0x0	DBI_DCX_SEL DBI DCX PIN Function Select 0: DBI DCX Function 1: WRX (2 Data Lane Interface)
4:3	R/W	0x0	DBI_SDI_SEL DBI SDI PIN Function Select 00: DBI_SDI (Interface II) 01: DBI_TE 10: DBI_DCX 11: Reserved
2	R/W	0x0	TE_DBC_SEL TE debounce function select 0: debounce 1: no-debounce
1	R/W	0x0	TE_TRIG_SEL TE edge trigger select 0: TE rising edge 1: TE falling edge
0	R/W	0x0	TE_EN TE enable 0: TE Disable 1: TE Enable

#### 8.4.6.20 0x010C DBI Timer Control Register (Default Value: 0x0000\_0000)

Offset: 0x010C			Register Name: DBI_Timer
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<p>DBI_TM_EN DBI Timer Enable 0: Enable 1: Disable</p>
30:0	R/W	0x0	<p>DBI Timer Value It sets the time interval between sending data twice, which is frame blanking. It is used to set the time at which the interrupt of the DBI Timer is triggered. When the Timer_EN is 1, the timer starts to count (the clock of the counting is SCLK), and the counter reaches the target value to trigger the Timer_INT of DBI, the data will start to send in series.</p> <p><b>Note: Do not count when sending the series data.</b></p>

#### 8.4.6.21 0x0110 DBI Video Size Register (Default Value: 0x01E0\_0140)

Offset: 0x0110			Register Name: DBI_Video_Size
Bit	Read/Write	Default/Hex	Description
31:27	/	/	/
26:16	R/W	0x1E0	V_SIZE It is used to generate the Frame int.
15:11	/	/	/
10:0	R/W	0x140	H_SIZE It is used to generate the Line int.

#### 8.4.6.22 0x0120 DBI Interrupt Register (Default Value: 0x0000\_4000)

Offset: 0x0120			Register Name: DBI_INT
Bit	Read/Write	Default/Hex	Description
31:15	/	/	/

Offset: 0x0120			Register Name: DBI_INT
Bit	Read/Write	Default/Hex	Description
14	R/W1C	0x1	<p>DBI_FIFO_EMPTY_INT DBI FIFO Empty Interrupt Status 0: DBI_FIFO is not empty 1: DBI_FIFO is empty</p>
13	R/W1C	0x0	<p>DBI_FIFO_FULL_INT DBI FIFO Full Interrupt Status 0: DBI_FIFO is not full 1: DBI_FIFO is full</p>
12	R/W1C	0x0	<p>TIMER_INT It indicates that the timer has been count sclk cycles to the value of DBI_Timer Register[30:0]. Writing 1 to this bit clears it. 0: Timer has not been achieved the objective 1: Timer has been achieved the objective</p>
11	R/W1C	0x0	<p>RD_DONE_INT It indicates that the number of byte setting in DBI_Control Register 1[19:8] has been read. Writing 1 to this bit clears it. 0: All data has not been read 1: All data has been read</p>
10	R/W1C	0x0	<p>TE_INT It indicates that the TE signal has been changed. Writing 1 to this bit clears it. 0: TE signal has not been changed 1: TE signal has been changed</p>
9	R/W1C	0x0	<p>FRAM_DONE_INT It indicates that a frame video data has been sent. Writing 1 to this bit clears it. 0: A frame video has not been sent 1: A frame video has been sent</p>
8	R/W1C	0x0	<p>LINE_DONE_INT It indicates that a line of video data has been sent. Writing 1 to this bit clears it. 0: A line of video data has not been sent 1: A line of video data has been sent</p>
7	/	/	/

Offset: 0x0120			Register Name: DBI_INT
Bit	Read/Write	Default/Hex	Description
6	R/W	0x0	DBI_FIFO_EMPTY_INT_EN DBI FIFO Empty Interrupt Enable 0: Disable 1: Enable
5	R/W	0x0	DBI_FIFO_FULL_INT_EN DBI FIFO Full Interrupt Enable 0: Disable 1: Enable
4	R/W	0x0	TIMER_INT_EN Timer Interrupt Enable 0: Disable 1: Enable
3	R/W	0x0	RD_DONE_INT_EN Read Done Interrupt Enable 0: Disable 1: Enable
2	R/W	0x0	TE_INT_EN TE Interrupt Enable 0: Disable 1: Enable
1	R/W	0x0	FRAM_DONE_INT_EN Frame Done Interrupt Enable 0: Disable 1: Enable
0	R/W	0x0	LINE_DONE_INT_EN Line Done Interrupt Enable 0: Disable 1: Enable

#### 8.4.6.23 0x0124 DBI Debug Register 0 (Default Value: 0x007F\_0000)

Offset: 0x0124			Register Name: DBI_Debug_0
Bit	Read/Write	Default/Hex	Description
31:23	/	/	/

Offset: 0x0124			Register Name: DBI_Debug_0
Bit	Read/Write	Default/Hex	Description
22:16	R	0x7F	DBI_FIFO_AVAIL DBI_FIFO ROOM VALID 0~127 Words
15:13	/	/	/
12	R	0x0	TE_VAL TE input value 0: TE not Trigger 1: TE Trigger
11:8	R	0x0	DBI_RXCS FSM for DBI Receive RX_BS0 ~ RX_BS6 , Gray - Code
7:4	R	0x0	SH_CS FSM for shifter 0~11 : SH0~SH11
3:2	/	/	DBI_TXCS FSM for DBI Transmit 00: IDLE 01: SHIF 10: DUMY 11: READ
1:0	R	0x0	MEM_CS FSM for DBI Memory 00: IDLE_FRM 01: FRM_POS 10: FRM_RDY

#### 8.4.6.24 0x0128 DBI Debug Register 1 (Default Value: 0x0000\_0000)

Offset: 0x0128			Register Name: DBI_Debug_1
Bit	Read/Write	Default/Hex	Description
31:26	/	/	/
25:16	R	0x0	LCNT Line counter The number of pixel lines that are currently sent
15:12	/	/	/

Offset: 0x0128			Register Name: DBI_Debug_1
Bit	Read/Write	Default/Hex	Description
11:0	R	0x0	<p>CCNT</p> <p>Component counter</p> <p>The number of RGB components that are currently sent</p> <p>The field is equal to pixel_cnt *3.</p>

#### 8.4.6.25 0x0200 SPI TX Data Register (Default Value: 0x0000\_0000)

Offset: 0x0200			Register Name: SPI_TXD
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<p>TDATA</p> <p>Transmit Data</p> <p>This register can be accessed in the byte, half-word, or word unit by AHB. In the byte accessing method, if there are rooms in TXFIFO, one burst data is written to TXFIFO and the depth is increased by 1. In the half-word accessing method, two SPI burst data are written and the TXFIFO depth is increased by 2. In the word accessing method, four SPI burst data are written and the TXFIFO depth is increased by 4.</p> <p><b>Note: This address is writable-only if TF_TEST is '0', and if TF_TEST is set to '1', this address is readable and writable to test the TX FIFO through the AHB bus.</b></p>

**8.4.6.26 0x0300 SPI RX Data Register (Default Value: 0x0000\_0000)**

Offset: 0x0300			Register Name: SPI_RXD
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	<p>RDATA Receive Data</p> <p>This register can be accessed in the byte, half-word, or word unit by AHB. In the byte accessing method, if there are data in RXFIFO, the top word is returned and the RXFIFO depth is decreased by 1. In the half-word accessing method, two SPI bursts are returned and the RXFIFO depth is decreased by 2. In the word accessing method, the four SPI bursts are returned and the RXFIFO depth is decreased by 4.</p> <p><b>Note:</b> This address is readable-only if RF_TEST is '0', and if RF_TEST is set to '1', this address is readable and writable to test the RX FIFO through the AHB bus.</p>

## 8.5 USB2.0 DRD

### 8.5.1 Overview

The USB2.0 dual-role device (USB2.0 DRD) supports both device and host functions which can also be configured as a Host-only or Device-only controller. It complies with the USB2.0 Specification.

For saving CPU bandwidth, the DMA interface of the DRD module can also support the external DMA controller to do the data transfer between the memory and the DRD FIFO. The DRD core also supports USB power saving functions.

The USB2.0 DRD has the following features:

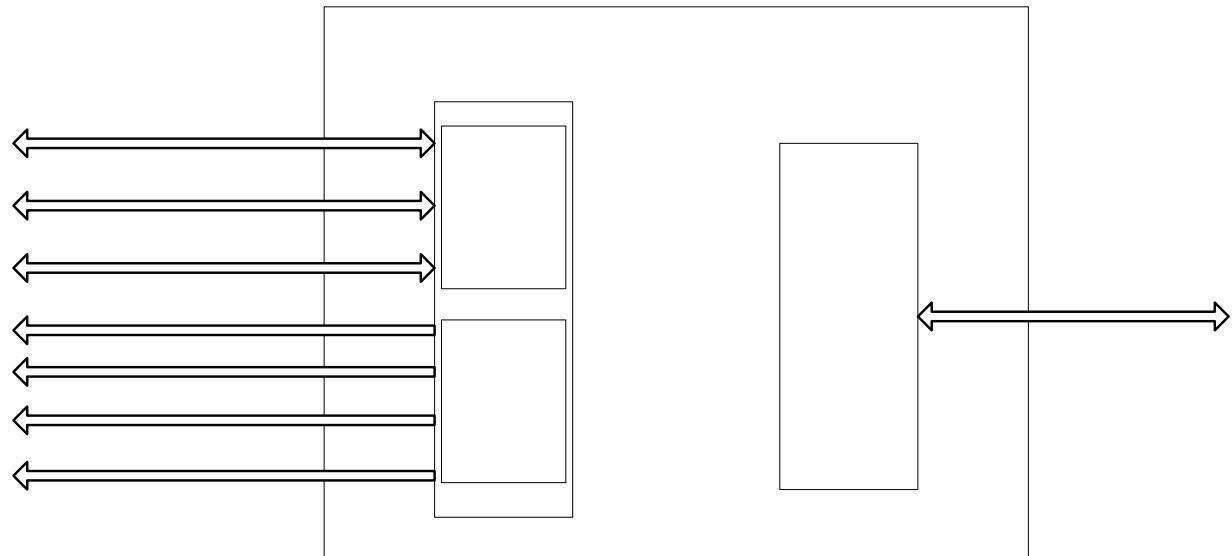
- Complies with USB2.0 Specification
  - Compatible with Enhanced Host Controller Interface (EHCI) Specification, Version 1.0
  - Compatible with Open Host Controller Interface (OHCI) Specification, Version 1.0a
  - Supports High-Speed (HS, 480 Mbit/s), Full-Speed (FS, 12 Mbit/s), and Low-Speed (LS, 1.5 Mbit/s)
  - Supports only 1 USB Root port shared between EHCI and OHCI
- Supports USB Device function
  - Supports High-Speed (HS, 480 Mbit/s), Full-Speed (FS, 12 Mbit/s)
  - Supports bi-directional endpoint0 (EPO) for Control transfer
  - Up to 10 user-configurable endpoints (EP1+, EP1-, EP2+, EP2-, EP3+, EP3-, EP4+, EP4-, EP5+, EP5-) for Bulk transfer, Isochronous transfer and Interrupt transfer
  - Up to (8 KB + 64 Bytes) FIFO for all EPs (including EPO)
  - Supports interface to an external Normal DMA controller for every EP
- Supports an internal DMA controller for data transfer with memory
- Supports High-Bandwidth Isochronous & Interrupt transfers
- Automated splitting/combining of packets for Bulk transfers
- Supports point-to-point and point-to-multipoint transfer in both Host and Peripheral modes
- Includes automatic ping capabilities
- Soft connect/disconnect function
- Performs all transaction scheduling in hardware

- Power optimization and power management capabilities
- Device and host controller share a 8K SRAM and a physical PHY

### 8.5.2 Block Diagram

The following figure shows the block diagram of USB2.0 DRD Controller.

**Figure 8-63 USB2.0 DRD Controller Block Diagram**



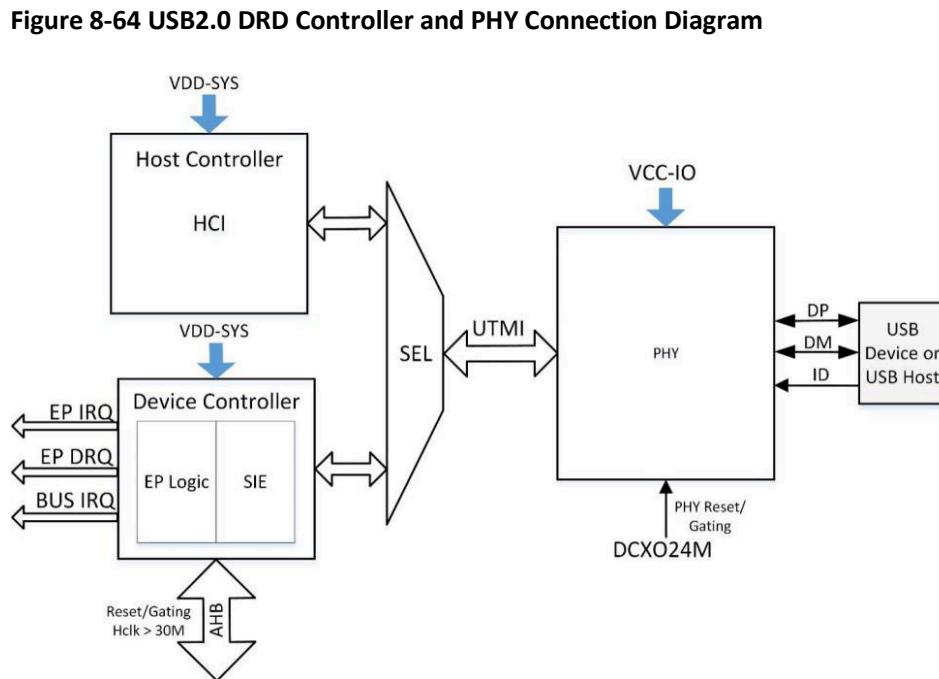
### 8.5.3 Functional Description

#### 8.5.3.1 External Signals

**Table 8-18 USB2.0 DRD External Signals**

Signal	Description	Type
USBO-DP	USB2.0 DRD differential signal positive	AI/O
USBO-DM	USB2.0 DRD differential signal negative	AI/O

### 8.5.3.2 Controller and PHY Connection Diagram



## 8.6 USB2.0 HOST

### 8.6.1 Overview

The USB Host Controller is fully compliant with USB 2.0 Specification, Enhanced Host Controller Interface (EHCI) Specification Revision 1.0 and Open Host Controller Interface (OHCI) Specification Release 1.0a.

The USB2.0 host controller includes the following features:

- Complies with USB2.0 Specification
- Supports USB2.0 Host function
  - Compatible with Enhanced Host Controller Interface (EHCI) Specification, Version 1.0
  - Compatible with Open Host Controller Interface (OHCI) Specification, Version 1.0a
  - Supports High-Speed (HS, 480 Mbit/s), Full-Speed (FS, 12 Mbit/s) and Low-Speed (LS, 1.5 Mbit/s) Device
  - Supports only 1 USB Root port shared between EHCI and OHCI
- An internal DMA Controller for data transfer with memory

### 8.6.2 Block Diagram

The following figure shows the block diagram of USB2.0 Host Controller.

**Figure 8-65 USB2.0 Host Controller Block Diagram**

### 8.6.3 Functional Description

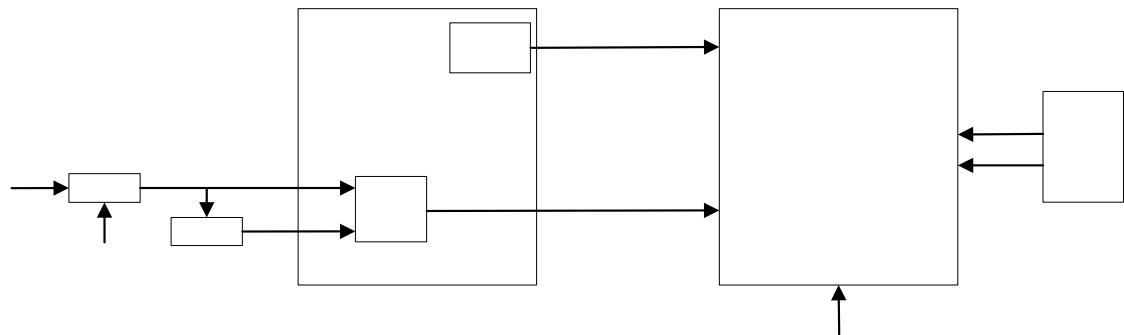
#### 8.6.3.1 External Signals

**Table 8-19 USB2.0 Host External Signals**

Signal	Description	Type
USB1-DP	USB2.0 Host differential signal positive	AI/O
USB1-DM	USB2.0 Host differential signal negative	AI/O

#### 8.6.3.2 Controller and PHY Connection Diagram

**Figure 8-66 USB2.0 Host Controller and PHY Connection Diagram**



### 8.6.4 Register List

Module Name	Base Address
USB1	0x04200000

Register Name	Offset	Description
<b>EHCI Capability Register</b>		
E_CAPLENGTH	0x0000	EHCI Capability Register Length Register
E_HCIVERSION	0x0002	EHCI Host Interface Version Number Register
E_HCSPARAMS	0x0004	EHCI Host Control Structural Parameter Register
E_HCCPARAMS	0x0008	EHCI Host Control Capability Parameter Register

<b>Register Name</b>	<b>Offset</b>	<b>Description</b>
E_HCSPPORTROUTE	0x000C	EHCI Companion Port Route Description
<b>EHCI Operational Register</b>		
E_USBCMD	0x0010	EHCI USB Command Register
E_USBSTS	0x0014	EHCI USB Status Register
E_USBINTR	0x0018	EHCI USB Interrupt Enable Register
E_FRINDEX	0x001C	EHCI USB Frame Index Register
E_CTRLDSSEGMENT	0x0020	EHCI 4G Segment Selector Register
E_PERIODICLISTBASE	0x0024	EHCI Frame List Base Address Register
E_ASYNCNLISTADDR	0x0028	EHCI Next Asynchronous List Address Register
E_CONFIGFLAG	0x0050	EHCI Configured Flag Register
E_PORTSC	0x0054	EHCI Port Status/Control Register
<b>OHCI Control and Status Partition Register</b>		
O_HcControl	0x0404	OHCI Control Register
O_HcCommandStatus	0x0408	OHCI Command Status Register
O_HcInterruptStatus	0x040C	OHCI Interrupt Status Register
O_HcInterruptEnable	0x0410	OHCI Interrupt Enable Register
O_HcInterruptDisable	0x0414	OHCI Interrupt Disable Register
<b>OHCI Memory Pointer Partition Register</b>		
O_HcHCCA	0x0418	OHCI HCCA Base
O_HcPeriodCurrentED	0x041C	OHCI Period Current ED Base
O_HcControlHeadED	0x0420	OHCI Control Head ED Base
O_HcControlCurrentED	0x0424	OHCI Control Current ED Base
O_HcBulkHeadED	0x0428	OHCI Bulk Head ED Base
O_HcBulkCurrentED	0x042C	OHCI Bulk Current ED Base
O_HcDoneHead	0x0430	OHCI Done Head Base
<b>OHCI Frame Counter Partition Register</b>		
O_HcFmInterval	0x0434	OHCI Frame Interval Register
O_HcFmRemaining	0x0438	OHCI Frame Remaining Register
O_HcFmNumber	0x043C	OHCI Frame Number Register
O_HcPeriodicStart	0x0440	OHCI Periodic Start Register
O_HcLSThreshold	0x0444	OHCI LS Threshold Register
<b>OHCI Root Hub Partition Register</b>		
O_HcRhDescriptorA	0x0448	OHCI Root Hub Descriptor Register A
O_HcRhDescriptorB	0x044C	OHCI Root Hub Descriptor Register B

Register Name	Offset	Description
O_HcRhStatus	0x0450	OHCI Root Hub Status Register
O_HcRhPortStatus	0x0454	OHCI Root Hub Port Status Register
<b>HCI Controller and PHY Interface Register</b>		
HCI_Interface	0x0800	HCI Interface Register
HCI_CTRL3	0x0808	HCI Control Register
PHY_Control	0x0810	PHY Control Register
PHY_STATUS	0x0824	PHY Status Register
HCI SIE Port Disable Control	0x0828	HCI SIE Port Disable Control Register

### 8.6.5 EHCI Register Description

#### 8.6.5.1 0x0000 EHCI Identification Register (Default Value:0x10)

Offset:0x0000			Register Name: CAPLENGTH
Bit	Read/Write	Default/Hex	Description
7:0	R	0x10	CAPLENGTH The value in these bits indicates an offset to add to register base to find the beginning of the Operational Register Space.

#### 8.6.5.2 0x0002 EHCI Host Interface Version Number Register (Default Value:0x0100)

Offset: 0x0002			Register Name: HCIVERSION
Bit	Read/Write	Default/Hex	Description
15:0	R	0x0100	HCIVERSION This is a 16-bit register containing a BCD encoding of the EHCI revision number supported by this host controller. The most significant byte of this register represents a major revision and the least significant byte is the minor revision.

#### 8.6.5.3 0x0004 EHCI Host Control Structural Parameter Register (Default Value:0x0000\_1101)

Offset: 0x0004			Register Name: HCSPARAMS
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23:20	R	0x0	Debug Port Number

Offset: 0x0004			Register Name: HCSPARAMS						
Bit	Read/Write	Default/Hex	Description						
			<p>This register identifies which of the host controller ports is the debug port. The value is the port number (one based) of the debug port.</p> <p>This field will always be '0'.</p>						
19:16	/	/	/						
15:12	R	0x1	<p>Number of Companion Controller (N_CC)</p> <p>This field indicates the number of companion controllers associated with this USB2.0 host controller. A zero in this field indicates there are no companion host controllers. And a value larger than zero in this field indicates there are companion USB1.1 host controller(s).</p> <p>This field will always be '0'.</p>						
11:8	R	0x1	<p>Number of Port per Companion Controller (N_PCC)</p> <p>This field indicates the number of ports supported per companion host controller host controller. It is used to indicate the port routing configuration to system software.</p> <p>This field will always fix with '0'.</p>						
7	R	0x0	<p>Port Routing Rules</p> <p>This field indicates the method used by this implementation for how all ports are mapped to companion controllers. The value of this field has the following interpretation:</p> <table border="1"> <thead> <tr> <th>Value</th><th>Meaning</th></tr> </thead> <tbody> <tr> <td>0</td><td>The first N_PCC ports are routed to the lowest numbered function companion host controller, the next N_PCC port are routed to the next lowest function companion controller, and so on.</td></tr> <tr> <td>1</td><td>The port routing is explicitly enumerated by the first N_PORTS elements of the HCSP-PORTROUTE array.</td></tr> </tbody> </table> <p>This field will always be '0'.</p>	Value	Meaning	0	The first N_PCC ports are routed to the lowest numbered function companion host controller, the next N_PCC port are routed to the next lowest function companion controller, and so on.	1	The port routing is explicitly enumerated by the first N_PORTS elements of the HCSP-PORTROUTE array.
Value	Meaning								
0	The first N_PCC ports are routed to the lowest numbered function companion host controller, the next N_PCC port are routed to the next lowest function companion controller, and so on.								
1	The port routing is explicitly enumerated by the first N_PORTS elements of the HCSP-PORTROUTE array.								
6:4	/	/	/						
3:0	R	0x1	<p>N_PORTS</p> <p>This field specifies the number of physical downstream ports implemented on this host controller. The value of this field determines how many port registers are addressable in the Operational Register Space. Valid values are in the range of 0x1 to 0x0f.</p> <p>This field is always 1.</p>						

## 8.6.5.4 0x0008 EHCI Host Control Capability Parameter Register (Default Value:0x0000\_A026)

Offset: 0x0008			Register Name: HCCPARAMS
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:8	R	0xA0	<p>EHCI Extended Capabilities Pointer (EECP)</p> <p>This optional field indicates the existence of a capabilities list. A value of 00b indicates no extended capabilities are implemented. A non-zero value in this register indicates the offset in PCI configuration space of the first EHCI extended capability. The pointer value must be 40h or greater if implemented to maintain consistency of the PCI header defined for this class of device.</p> <p>The value of this field is always '00b'.</p>
7:4	R	0x2	<p>Isochronous Scheduling Threshold</p> <p>This field indicates, relative to the current position of the executing host controller, where software can reliably update the isochronous schedule.</p> <p>When bit[7] is zero, the value of the least significant 3 bits indicates the number of micro-frames a host controller can hold a set of isochronous data structures(one or more) before flushing the state. When bit[7] is a one, then host software assumes the host controller may cache an isochronous data structure for an entire frame.</p>
3	/	/	/
2	R	0x1	<p>Asynchronous Schedule Park Capability</p> <p>If this bit is set to a one, then the host controller supports the park feature for high-speed queue heads in the Asynchronous Schedule. The feature can be disabled or enabled and set to a specific level by using the Asynchronous Schedule Park Mode Enable and Asynchronous Schedule Park Mode Count fields in the USBCMD register.</p>
1	R	0x1	<p>Programmable Frame List Flag</p> <p>If this bit is set to a zero, then system software must use a frame list length of 1024 elements with this host controller. The USBCMD register Frame List Size field is a read-only register and should be set to zero.</p> <p>If set to 1, then system software can specify and use the frame list in the USBCMD register Frame List Size field to configure the host controller.</p> <p>The frame list must always aligned on a 4K page boundary. This requirement ensures that the frame list is always physically contiguous.</p>
0	/	/	/

#### 8.6.5.5 0x000C EHCI Companion Port Route Description (Default Value:0x0000\_0000)

Offset: 0x000C			Register Name: HCSP-PORTROUTE
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	<p>HCSP-PORTROUTE</p> <p>This optional field is valid only if Port Routing Rules field in HCSPARAMS register is set to a one.</p> <p>This field is used to allow a host controller implementation to explicitly describe to which companion host controller each implemented port is mapped. This field is a 15-element nibble array (each 4 bit is one array element). Each array location corresponds one-to-one with a physical port provided by the host controller (e.g. PORTROUTE [0] corresponds to the first PORTSC port, PORTROUTE [1] to the second PORTSC port, etc.). The value of each element indicates to which the companion host controllers this port is routed. Only the first N_PORTS elements have valid information. A value of zero indicates that the port is routed to the lowest numbered function companion host controller. A value of one indicates that the port is routed to the next lowest numbered function companion host controller, and so on.</p>

#### 8.6.5.6 0x0010 EHCI USB Command Register (Default Value:0x0008\_0B00)

Offset: 0x0010			Register Name: USBCMD																		
Bit	Read/Write	Default/Hex	Description																		
31:24	/	/	/																		
23:16	R/W	0x08	<p>Interrupt Threshold Control</p> <p>The value in this field is used by system software to select the maximum rate at which the host controller will issue interrupts. The only valid values are defined below:</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Minimum Interrupt Interval</th> </tr> </thead> <tbody> <tr> <td>0x00</td> <td>Reserved</td> </tr> <tr> <td>0x01</td> <td>1 micro-frame</td> </tr> <tr> <td>0x02</td> <td>2 micro-frame</td> </tr> <tr> <td>0x04</td> <td>4 micro-frame</td> </tr> <tr> <td>0x08</td> <td>8 micro-frame(default, equates to 1 ms)</td> </tr> <tr> <td>0x10</td> <td>16 micro-frame(2ms)</td> </tr> <tr> <td>0x20</td> <td>32 micro-frame(4ms)</td> </tr> <tr> <td>0x40</td> <td>64 micro-frame(8ms)</td> </tr> </tbody> </table>	Value	Minimum Interrupt Interval	0x00	Reserved	0x01	1 micro-frame	0x02	2 micro-frame	0x04	4 micro-frame	0x08	8 micro-frame(default, equates to 1 ms)	0x10	16 micro-frame(2ms)	0x20	32 micro-frame(4ms)	0x40	64 micro-frame(8ms)
Value	Minimum Interrupt Interval																				
0x00	Reserved																				
0x01	1 micro-frame																				
0x02	2 micro-frame																				
0x04	4 micro-frame																				
0x08	8 micro-frame(default, equates to 1 ms)																				
0x10	16 micro-frame(2ms)																				
0x20	32 micro-frame(4ms)																				
0x40	64 micro-frame(8ms)																				

Offset: 0x0010			Register Name: USBCMD
Bit	Read/Write	Default/Hex	Description
			<p>Any other value in this register yields undefined results.</p> <p>The default value in this field is 0x08 .</p> <p>Software modifications to this bit while HC Halted bit is equal to zero results in undefined behavior.</p>
15:12	/	/	/
11	R	0x1	<p>Asynchronous Schedule Park Mode Enable (OPTIONAL)</p> <p>If the Asynchronous Park Capability bit in the HCCPARAMS register is a one, then this bit defaults to a 1 and is R/W. Otherwise the bit must be a zero and is Read Only. Software uses this bit to enable or disable Park mode. When this bit is one, Park mode is enabled. When this bit is zero, Park mode is disabled.</p>
10	/	/	/
9:8	R	0x3	<p>Asynchronous Schedule Park Mode Count (OPTIONAL)</p> <p>Asynchronous Park Capability bit in the HCCPARAMS register is a one, then this field defaults to 0x3 and is W/R. Otherwise it defaults to zero and is R. It contains a count of the number of successive transactions the host controller is allowed to execute from a high-speed queue head on the Asynchronous schedule before continuing traversal of the Asynchronous schedule.</p> <p>Valid value are 0x1 to 0x3. Software must not write a zero to this bit when Park Mode Enable is a one as it will result in undefined behavior.</p>
7	R/W	0x0	<p>Light Host Controller Reset (OPTIONAL)</p> <p>This control bit is not required.</p> <p>If implemented, it allows the driver to reset the EHCI controller without affecting the state of the ports or relationship to the companion host controllers. For example, the PORSTC registers should not be reset to their default values and the CF bit setting should not go to zero (retaining port ownership relationships).</p> <p>A host software read of this bit as zero indicates the Light Host Controller Reset has completed and it is safe for software to re-initialize the host controller. A host software read of this bit as a one indicates the Light Host</p>
6	R/W	0x0	<p>Interrupt on Async Advance Doorbell</p> <p>This bit is used as a doorbell by software to tell the host controller to issue an interrupt the next time it advances asynchronous schedule. Software must write a 1 to this bit to ring the doorbell.</p> <p>When the host controller has evicted all appropriate cached schedule state, it sets the Interrupt on Async Advance status bit in the USBSTS. if the Interrupt on Async Advance Enable bit in the</p>

Offset: 0x0010			Register Name: USBCMD										
Bit	Read/Write	Default/Hex	Description										
			<p>USBINTR register is a one then the host controller will assert an interrupt at the next interrupt threshold.</p> <p>The host controller sets this bit to a zero after it has set the Interrupt on Async Advance status bit in the USBSTS register to a one.</p> <p>Software should not write a one to this bit when the asynchronous schedule is disabled. Doing so will yield undefined results.</p>										
5	R/W	0x0	<p>Asynchronous Schedule Enable</p> <p>This bit controls whether the host controller skips processing the Asynchronous Schedule. Values mean:</p> <table border="1"> <thead> <tr> <th>Bit Value</th><th>Meaning</th></tr> </thead> <tbody> <tr> <td>0</td><td>Do not process the Asynchronous Schedule.</td></tr> <tr> <td>1</td><td>Use the ASYNLISTADDR register to access the Asynchronous Schedule.</td></tr> </tbody> </table> <p>The default value of this field is '0b'.</p>	Bit Value	Meaning	0	Do not process the Asynchronous Schedule.	1	Use the ASYNLISTADDR register to access the Asynchronous Schedule.				
Bit Value	Meaning												
0	Do not process the Asynchronous Schedule.												
1	Use the ASYNLISTADDR register to access the Asynchronous Schedule.												
4	R/W	0x0	<p>Periodic Schedule Enable</p> <p>This bit controls whether the host controller skips processing the Periodic Schedule. Values mean:</p> <table border="1"> <thead> <tr> <th>Bit Value</th><th>Meaning</th></tr> </thead> <tbody> <tr> <td>0</td><td>Do not process the Periodic Schedule.</td></tr> <tr> <td>1</td><td>Use the PERIODICLISTBASE register to access the Periodic Schedule.</td></tr> </tbody> </table> <p>The default value of this field is '0b'.</p>	Bit Value	Meaning	0	Do not process the Periodic Schedule.	1	Use the PERIODICLISTBASE register to access the Periodic Schedule.				
Bit Value	Meaning												
0	Do not process the Periodic Schedule.												
1	Use the PERIODICLISTBASE register to access the Periodic Schedule.												
3:2	R/W	0x0	<p>Frame List Size</p> <p>This field is R/W only if Programmable Frame List Flag in the HCCPARAMS registers is set to a one. This field specifies the size of the</p> <p>Frame list. The size the frame list controls which bits in the Frame Index</p> <p>Register should be used for the Frame List Current index. Values mean:</p> <table border="1"> <thead> <tr> <th>Bits</th><th>Meaning</th></tr> </thead> <tbody> <tr> <td>00b</td><td>1024 elements(4096bytes)Default value</td></tr> <tr> <td>01b</td><td>512 elements(2048bytes)</td></tr> <tr> <td>10b</td><td>256 elements(1024bytes)For resource-constrained condition</td></tr> <tr> <td>11b</td><td>reserved</td></tr> </tbody> </table> <p>The default value is '00b'.</p>	Bits	Meaning	00b	1024 elements(4096bytes)Default value	01b	512 elements(2048bytes)	10b	256 elements(1024bytes)For resource-constrained condition	11b	reserved
Bits	Meaning												
00b	1024 elements(4096bytes)Default value												
01b	512 elements(2048bytes)												
10b	256 elements(1024bytes)For resource-constrained condition												
11b	reserved												

Offset: 0x0010			Register Name: USBCMD
Bit	Read/Write	Default/Hex	Description
1	R/W	0x0	<p>Host Controller Reset</p> <p>This control bit is used by software to reset the host controller. The effects of this on Root Hub registers are similar to a Chip Hardware Reset.</p> <p>When software writes a one to this bit, the Host Controller resets its internal pipelines, timers, counters, state machines, etc. to their initial value. Any transaction currently in progress on USB is immediately terminated. A USB reset is not driven on downstream ports.</p> <p>All operational registers, including port registers and port state machines are set to their initial values. Port ownership reverts to the companion host controller(s). Software must reinitialize the host controller as described in Section 4.1 of the CHEI Specification in order to return the host controller to an operational state.</p> <p>This bit is set to zero by the Host Controller when the reset process is complete. Software cannot terminate the reset process early by writing a zero to this register.</p> <p>Software should not set this bit to a one when the HC Halted bit in the USBSTS register is a zero. Attempting to reset an actively running host controller will result in undefined behavior.</p>
0	R/W	0x0	<p>Run/Stop</p> <p>When set to a 1, the Host Controller proceeds with execution of the schedule. When set to 0, the Host Controller completes the current and any actively pipelined transactions on the USB and then halts. The Host Controller must halt within 16 micro-frames after software clears this bit.</p> <p>The HC Halted bit indicates when the Host Controller has finished its pending pipelined transactions and has entered the stopped state.</p> <p>Software must not write a one to this field unless the Host Controller is in the Halt State.</p> <p>The default value is 0x0.</p>

#### 8.6.5.7 0x0014 EHCI USB Status Register (Default Value:0x0000\_1000)

Offset: 0x0014			Register Name: USBSTS
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15	R	0x0	Asynchronous Schedule Status

Offset: 0x0014			Register Name: USBSTS
Bit	Read/Write	Default/Hex	Description
			The bit reports the current real status of Asynchronous Schedule. If this bit is a zero then the status of the Asynchronous Schedule is disabled. If this bit is a one then the status of the Asynchronous Schedule is enabled. The Host Controller is not required to immediately disable or enable the Asynchronous Schedule when software transitions the Asynchronous Schedule Enable bit in the USBCMD register. When this bit and the Asynchronous Schedule Enable bit are the same value, the Asynchronous Schedule is either enabled (1) or disabled (0).
14	R	0x0	<p>Periodic Schedule Status</p> <p>The bit reports the current real status of the Periodic Schedule. If this bit is a zero then the status of the Periodic Schedule is disabled. If this bit is a one then the status of the Periodic Schedule is enabled. The Host Controller is not required to <i>immediately</i> disable or enable the Periodic Schedule when software transitions the <i>Periodic Schedule Enable</i> bit in the USBCMD register. When this bit and the <i>Periodic Schedule Enable</i> bit are the same value, the Periodic Schedule is either enabled (1) or disabled (0).</p>
13	R	0x0	<p>Reclamation</p> <p>This is a read-only status bit, which is used to detect an empty asynchronous schedule.</p>
12	R	0x1	<p>HC Halted</p> <p>This bit is a zero whenever the Run/Stop bit is a one. The Host Controller Sets this bit to one after it has stopped executing as a result of the Run/Stop bit being set to 0, either by software or by the Host Controller Hardware (e.g. internal error).</p> <p>The default value is '1'.</p>
11:6	/	/	/
5	R/WC	0x0	<p>Interrupt on Async Advance</p> <p>System software can force the host controller to issue an interrupt the next time the host controller advances the asynchronous schedule by writing a one to the Interrupt on Async Advance Doorbell bit in the USBCMD register. This status bit indicates the assertion of that interrupt source.</p>
4	R/WC	0x0	<p>Host System Error</p> <p>The Host Controller set this bit to 1 when a serious error occurs during a host system access involving the Host Controller module. When this error occurs, the Host Controller clears the Run/Stop bit in the Command register to prevent further execution of the scheduled TDs.</p>
3	R/WC	0x0	Frame List Rollover

Offset: 0x0014			Register Name: USBSTS
Bit	Read/Write	Default/Hex	Description
			The Host Controller sets this bit to a one when the Frame List Index rolls over from its maximum value to zero. The exact value at which the rollover occurs depends on the frame list size. For example, if the frame list size is 1024, the Frame Index Register rolls over every time FRINDEX [13] toggles. Similarly, if the size is 512, the Host Controller sets this bit to a one every time FRINDEX [12] toggles.
2	R/WC	0x0	Port Change Detect The Host Controller sets this bit to a one when any port for which the Port Owner bit is set to zero has a change bit transition from a zero to a one or a Force Port Resume bit transition from a zero to a one as a result of a J-K transition detected on a suspended port. This bit will also be set as a result of the Connect Status Chang being set to a one after system software has relinquished ownership of a connected port by writing a one to a port's Port Owner bit.
1	R/WC	0x0	USB Error Interrupt(USBERRINT) The Host Controller sets this bit to 1 when completion of USB transaction results in an error condition(e.g. error counter underflow).If the TD on which the error interrupt occurred also had its IOC bit set, both. This bit and USBINT bit are set.
0	R/WC	0x0	USB Interrupt(USBINT) The Host Controller sets this bit to a one on the completion of a USB transaction, which results in the retirement of a Transfer Descriptor that had its IOC bit set. The Host Controller also sets this bit to 1 when a short packet is detected (actual number of bytes received was less than the expected number of bytes)

#### 8.6.5.8 0x0018 EHCI USB Interrupt Enable Register (Default Value:0x0000\_0000)

Offset: 0x0018			Register Name: USBINTR
Bit	Read/Write	Default/Hex	Description
31:6	/	/	/
5	R/W	0x0	Interrupt on Async Advance Enable When this bit is 1, and the Interrupt on Async Advance bit in the USBSTS register is 1, the host controller will issue an interrupt at the next interrupt threshold. The interrupt is acknowledged by software clearing the Interrupt on Async Advance bit.

Offset: 0x0018			Register Name: USBINTR
Bit	Read/Write	Default/Hex	Description
4	R/W	0x0	<p>Host System Error Enable</p> <p>When this bit is 1, and the Host System Error Status bit in the USBSTS register is 1, the host controller will issue an interrupt. The interrupt is acknowledged by software clearing the Host System Error bit.</p>
3	R/W	0x0	<p>Frame List Rollover Enable</p> <p>When this bit is 1, and the Frame List Rollover bit in the USBSTS register is 1, the host controller will issue an interrupt. The interrupt is acknowledged by software clearing the Frame List Rollover bit.</p>
2	R/W	0x0	<p>Port Change Interrupt Enable</p> <p>When this bit is 1, and the Port Chang Detect bit in the USBSTS register is 1, the host controller will issue an interrupt. The interrupt is acknowledged by software clearing the Port Chang Detect bit.</p>
1	R/W	0x0	<p>USB Error Interrupt Enable</p> <p>When this bit is 1, and the USBERRINT bit in the USBSTS register is 1, the host controller will issue an interrupt at the next interrupt threshold.</p> <p>The interrupt is acknowledged by software clearing the USBERRINT bit.</p>
0	R/W	0x0	<p>USB Interrupt Enable</p> <p>When this bit is 1, and the USBINT bit in the USBSTS register is 1, the host controller will issue an interrupt at the next interrupt threshold.</p> <p>The interrupt is acknowledged by software clearing the USBINT bit</p>

#### 8.6.5.9 0x001C EHCI Frame Index Register (Default Value:0x0000\_0000)

Offset: 0x001C			Register Name: FRINDEX
Bit	Read/Write	Default/Hex	Description
31:14	/	/	/
13:0	R/W	0	<p>Frame Index</p> <p>The value in this register increment at the end of each time frame (e.g. micro-frame). Bits[N:3] are used for the Frame List current index. It means that each location of the frame list is accessed 8 times (frames or Micro-frames) before moving to the next index.</p> <p>The following illustrates values of N based on the value of the Frame List Size field in the USBCMD register.</p>

Offset: 0x001C			Register Name: FRINDEX		
Bit	Read/Write	Default/Hex	Description		
			USBCMD[Frame List Size]	Number Elements	N
			00b	1024	12
			01b	512	11
			10b	256	10
			11b	Reserved	

 **NOTE**

This register must be written as a DWord. Byte writes produce undefined results.

#### 8.6.5.10 0x0024 EHCI Periodic Frame List Base Address Register (Default Value:0x0000\_0000)

Offset: 0x0024			Register Name: PERIODICLISTBASE		
Bit	Read/Write	Default/Hex	Description		
31:12	R/W	0x0	Base Address These bits correspond to memory address signals [31:12], respectively. This register contains the beginning address of the Periodic Frame List in the system memory. System software loads this register prior to starting the schedule execution by the Host Controller. The memory structure referenced by this physical memory pointer is assumed to be 4 Kbyte aligned. The contents of this register are combined with the Frame Index Register (FRINDEX) to enable the Host Controller to step through the Periodic Frame List in sequence.	/	/
11:0	/	/	/	/	/

 **NOTE**

Writes must be Dword Writes.

#### 8.6.5.11 0x0028 EHCI Current Asynchronous List Address Register (Default Value:0x0000\_0000)

Offset: 0x0028			Register Name: ASYNCLISTADDR
Bit	Read/Write	Default/Hex	Description
31:5	R/W	0x0	<p>Link Pointer (LP) This field contains the address of the next asynchronous queue head to be executed. These bits correspond to memory address signals [31:5], respectively.</p>
4:0	/	/	/



Write must be DWord Writes.

#### 8.6.5.12 0x0050 EHCI Configure Flag Register (Default Value:0x0000\_0000)

Offset: 0x0050			Register Name: CONFIGFLAG						
Bit	Read/Write	Default/Hex	Description						
31:1	/	/	/						
0	R/W	0x0	<p>Configure Flag (CF) Host software sets this bit as the last action in its process of configuring the Host Controller. This bit controls the default port-routing control logic as follow:</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Port routing control logic default-routs each port to an implementation dependent classic host controller.</td> </tr> <tr> <td>1</td> <td>Port routing control logic default-routs all ports to this host controller.</td> </tr> </tbody> </table> <p>The default value of this field is '0'.</p>	Value	Meaning	0	Port routing control logic default-routs each port to an implementation dependent classic host controller.	1	Port routing control logic default-routs all ports to this host controller.
Value	Meaning								
0	Port routing control logic default-routs each port to an implementation dependent classic host controller.								
1	Port routing control logic default-routs all ports to this host controller.								



This register is not used in the normal implementation.

## 8.6.5.13 0x0054 EHCI Port Status and Control Register (Default Value:0x0000\_2000)

Offset: 0x0054			Register Name: PORTSC																
Bit	Read/Write	Default/Hex	Description																
31:22	/	/	/																
21	R/W	0x0	<p>Wake on Disconnect Enable (WKDSCNNT_E) Writing this bit to a one enables the port to be sensitive to device disconnects as wake-up events. This field is zero if Port Power is zero. The default value in this field is '0'.</p>																
20	R/W	0x0	<p>Wake on Connect Enable (WKCNNT_E) Writing this bit to a one enable the port to be sensitive to device connects as wake-up events. This field is zero if Port Power is zero. The default value in this field is '0'.</p>																
19:16	R/W	0x0	<p>Port Test Control The value in this field specifies the test mode of the port. The encoding of the test mode bits are as follows:</p> <table border="1"> <thead> <tr> <th>Bits</th><th>Test Mode</th></tr> </thead> <tbody> <tr> <td>0000b</td><td>The port is NOT operating in a test mode.</td></tr> <tr> <td>0001b</td><td>Test J_STATE</td></tr> <tr> <td>0010b</td><td>Test K_STATE</td></tr> <tr> <td>0011b</td><td>Test SEO_NAK</td></tr> <tr> <td>0100b</td><td>Test Packet</td></tr> <tr> <td>0101b</td><td>Test FORCE_ENABLE</td></tr> <tr> <td>0110b-1111b</td><td>Reserved</td></tr> </tbody> </table> <p>The default value in this field is '0000b'.</p>	Bits	Test Mode	0000b	The port is NOT operating in a test mode.	0001b	Test J_STATE	0010b	Test K_STATE	0011b	Test SEO_NAK	0100b	Test Packet	0101b	Test FORCE_ENABLE	0110b-1111b	Reserved
Bits	Test Mode																		
0000b	The port is NOT operating in a test mode.																		
0001b	Test J_STATE																		
0010b	Test K_STATE																		
0011b	Test SEO_NAK																		
0100b	Test Packet																		
0101b	Test FORCE_ENABLE																		
0110b-1111b	Reserved																		
15:14	/	/	/																
13	R/W	0x1	<p>Port Owner This bit unconditionally goes to a 0b when the Configured bit in the CONFIGFLAG register makes a 0b to 1b transition. This bit unconditionally goes to 1b whenever the Configured bit is zero. System software uses this field to release ownership of the port to selected host controller (in the event that the attached device is not a high-speed device). Software writes a one to this bit when the attached device is not a high-speed device. A one in this bit means that a companion host controller owns and controls the port. Default Value = 1b.</p>																
12	/	/	/																

Offset: 0x0054			Register Name: PORTSC															
Bit	Read/Write	Default/Hex	Description															
11:10	R	0x0	<p>Line Status</p> <p>These bits reflect the current logical levels of the D+ (bit11) and D- (bit10) signal lines. These bits are used for detection of low-speed USB devices prior to port reset and enable sequence. This read only field is valid only when the port enable bit is zero and the current connect status bit is set to a one.</p> <p>The encoding of the bits are:</p> <table border="1"> <thead> <tr> <th>Bit[11:10]</th><th>USB State</th><th>Interpretation</th></tr> </thead> <tbody> <tr> <td>00b</td><td>SE0</td><td>Not Low-speed device, perform EHCI reset.</td></tr> <tr> <td>10b</td><td>J-state</td><td>Not Low-speed device, perform EHCI reset.</td></tr> <tr> <td>01b</td><td>K-state</td><td>Low-speed device, release ownership of port.</td></tr> <tr> <td>11b</td><td>Undefined</td><td>Not Low-speed device, perform EHCI reset.</td></tr> </tbody> </table> <p>This value of this field is undefined if Port Power is zero.</p>	Bit[11:10]	USB State	Interpretation	00b	SE0	Not Low-speed device, perform EHCI reset.	10b	J-state	Not Low-speed device, perform EHCI reset.	01b	K-state	Low-speed device, release ownership of port.	11b	Undefined	Not Low-speed device, perform EHCI reset.
Bit[11:10]	USB State	Interpretation																
00b	SE0	Not Low-speed device, perform EHCI reset.																
10b	J-state	Not Low-speed device, perform EHCI reset.																
01b	K-state	Low-speed device, release ownership of port.																
11b	Undefined	Not Low-speed device, perform EHCI reset.																
9	/	/	/															
8	R/W	0x0	<p>Port Reset</p> <p>1=Port is in Reset. 0=Port is not in Reset. Default value = 0.</p> <p>When software writes a one to this bit (from a zero), the bus reset sequence as defined in the USB Specification Revision 2.0 is started. Software writes a zero to this bit to terminate the bus reset sequence. Software must keep this bit at a one long enough to ensure the reset sequence, as specified in the USB Specification Revision 2.0, completes.</p> <p><b>Note: When software writes this bit to a one , it must also write a zero to the Port Enable bit.</b></p> <p>Note that when software writes a zero to this bit there may be a delay before the bit status changes to a zero. The bit status will not read as a zero until after the reset has completed. If the port is in high-speed mode after reset is complete, the host controller will automatically enable this port (e.g. set the Port Enable bit to a one). A host controller must terminate the reset and stabilize the state of the port within 2 milliseconds of software transitioning this bit from a one to a zero. For example: if the port detects that the attached device is high-speed during reset, then the host controller must have the port in the enabled state with 2ms of software writing this bit to a zero.</p>															

Offset: 0x0054			Register Name: PORTSC								
Bit	Read/Write	Default/Hex	Description								
			<p>The HC Halted bit in the USBSTS register should be a zero before software attempts to use this bit. The host controller may hold Port Reset asserted to a one when the HC Halted bit is a one.</p> <p>This field is zero if Port Power is zero.</p>								
7	R/W	0x0	<p>Suspend</p> <p>Port Enabled Bit and Suspend bit of this register define the port states as follows:</p> <table border="1"> <thead> <tr> <th>Bits[Port Enables, Suspend]</th><th>Port State</th></tr> </thead> <tbody> <tr> <td>0x</td><td>Disable</td></tr> <tr> <td>10</td><td>Enable</td></tr> <tr> <td>11</td><td>Suspend</td></tr> </tbody> </table> <p>When in suspend state, downstream propagation of data is blocked on this port, except for port reset. The blocking occurs at the end of the current transaction, if a transaction was in progress when this bit was written to 1. In the suspend state, the port is sensitive to resume detection. Note that the bit status does not change until the port is suspend and that there may be a delay in suspending a port if there is a transaction currently in progress on the USB.</p> <p>A write of zero to this bit is ignored by the host controller. The host controller will unconditionally set this bit to a zero when:</p> <ul style="list-style-type: none"> <li>① Software sets the Force Port Resume bit to a zero(from a one).</li> <li>② Software sets the Port Reset bit to a one(from a zero).</li> </ul> <p>If host software sets this bit to a one when the port is not enabled(i.e. Port enabled bit is a zero), the results are undefined.</p> <p>This field is zero if Port Power is zero.</p> <p>The default value in this field is '0'.</p>	Bits[Port Enables, Suspend]	Port State	0x	Disable	10	Enable	11	Suspend
Bits[Port Enables, Suspend]	Port State										
0x	Disable										
10	Enable										
11	Suspend										
6	R/W	0x0	<p>Force Port Resume</p> <p>1 = Resume detected/driven on port. 0 = No resume (K-state) detected/driven on port. Default value = 0.</p> <p>This functionality defined for manipulating this bit depends on the value of the Suspend bit. For example, if the port is not suspend and software transitions this bit to a one, then the effects on the bus are undefined.</p> <p>Software sets this bit to 1 to drive resume signaling. The Host Controller sets this bit to a 1 if a J-to-K transition is detected while the port is in the Suspend state. When this bit transitions to a one because a J-to-K transition is detected, the Port Change Detect bit in the USBSTS register is also set to a one. If software sets this bit</p>								

Offset: 0x0054			Register Name: PORTSC
Bit	Read/Write	Default/Hex	Description
			<p>to a one, the host controller must not set the Port Change Detect bit.</p> <p>Note that when the EHCI controller owns the port, the resume sequence follows the defined sequence documented in the USB Specification Revision 2.0. The resume signaling (Full-speed 'K') is driven on the port as long as this remains a one. Software must appropriately time the Resume and set this bit to a zero when the appropriate amount of time has elapsed. Writing a zero (from one) causes the port to return high-speed mode (forcing the bus below the port into a high-speed idle). This bit will remain a one until the port has switched to high-speed idle. The host controller must complete this transition within 2 milliseconds of software setting this bit to a zero.</p> <p>This field is zero if Port Power is zero.</p>
5	R/WC	0x0	<p>Over-current Change</p> <p>This bit gets set to a one when there is a change to Over-current Active. Software clears this bit by writing a one to this bit position.</p>
4	R	0x0	<p>Over-current Active</p> <p>0 = This port does not have an over-current condition</p> <p>1 = This port currently has an over-current condition</p> <p>This bit will automatically transition from a one to a zero when the over current condition is removed.</p> <p>The default value of this bit is '0'.</p>
3	R/WC	0x0	<p>Port Enable/Disable Change</p> <p>1 = Port enabled/disabled status has changed</p> <p>0 = No change</p> <p>For the root hub, this bit gets set to a one only when a port is disabled due to the appropriate conditions existing at the EOF2 point (See Chapter 11 of the USB Specification for the definition of a Port Error). Software clears this bit by writing a 1 to it.</p> <p>This field is zero if Port Power is zero.</p>
2	R/W	0x0	<p>Port Enabled/Disabled</p> <p>1=Enable</p> <p>0=Disable</p> <p>Ports can only be enabled by the host controller as a part of the reset and enable. Software cannot enable a port by writing a one to this field. The host controller will only set this bit to a one when the reset sequence determines that the attached device is a high-speed device.</p>

Offset: 0x0054			Register Name: PORTSC
Bit	Read/Write	Default/Hex	Description
			<p>Ports can be disabled by either a fault condition(disconnect event or other fault condition) or by host software. Note that the bit status does not change until the port state actually changes. There may be a delay in disabling or enabling a port due to other host controller and bus events.</p> <p>When the port is disabled, downstream propagation of data is blocked on this port except for reset.</p> <p>The default value of this field is '0'.</p> <p>This field is zero if Port Power is zero.</p>
1	R/WC	0x0	<p>Connect Status Change 1=Change in Current Connect Status 0=No change</p> <p>Indicates a change has occurred in the current connect status of the port. The host controller sets this bit for all changes to the port device connect status, even if system software has not cleared an existing connect status change. For example, the insertion status changes twice before system software has cleared the changed condition, hub hardware will be “setting” an already-set bit. Software sets this bit to 0 by writing a 1 to it.</p> <p>This field is zero if Port Power is zero.</p>
0	R	0x0	<p>Current Connect Status</p> <p>Device is present on port when the value of this field is a one, and no device is present on port when the value of this field is a zero.</p> <p>This value reflects the current state of the port, and may not correspond directly to the event that caused the Connect Status Change(Bit 1) to be set.</p> <p>This field is zero if Port Power zero.</p>



This register is only reset by hardware or in response to a host controller reset.

## 8.6.6 OHCI Register Description

### 8.6.6.1 0x0404 OHCI Control Register (Default Value: 0x0000\_0000)

Offset: 0x0404				Register Name: HcRevision								
Bit	Read/Write		Default/Hex	Description								
	HCD	HC										
31:11	/	/	/	/								
10	R/W	R	0x0	<p><b>RemoteWakeupEnable</b>  This bit is used by HCD to enable or disable the remote wakeup feature upon the detection of upstream resume signaling. When this bit is set and the <b>ResumeDetected</b> bit in <b>HcInterruptStatus</b> is set, a remote wakeup is signaled to the host system. Setting this bit has no impact on the generation of hardware interrupt.</p>								
9	R/W	R/W	0x0	<p><b>RemoteWakeupConnected</b>  This bit indicates whether HC supports remote wakeup signaling. If remote wakeup is supported and used by the system, it is the responsibility of system firmware to set this bit during POST. HC clear the bit upon a hardware reset but does not alter it upon a software reset. Remote wakeup signaling of the host system is host-bus-specific and is not described in this specification.</p>								
8	R/W	R	0x0	<p><b>InterruptRouting</b>  This bit determines the routing of interrupts generated by events registered in <b>HcInterruptStatus</b>. If clear, all interrupt are routed to the normal host bus interrupt mechanism. If set interrupts are routed to the System Management Interrupt. HCD clears this bit upon a hardware reset, but it does not alter this bit upon a software reset. HCD uses this bit as a tag to indicate the ownership of HC.</p>								
7:6	R/W	R/W	0x0	<p>HostControllerFunctionalState for USB</p> <table border="1"> <tr> <td>00b</td><td>USBReset</td></tr> <tr> <td>01b</td><td>USBResume</td></tr> <tr> <td>10b</td><td>USBOperational</td></tr> <tr> <td>11b</td><td>USBSuspend</td></tr> </table> <p>A transition to USBOperational from another state causes SOF generation to begin 1 ms later. HCD may determine whether HC has begun sending SOFs by reading the StartOfFrame field of <b>HcInterruptStatus</b>.</p> <p>This field may be changed by HC only when in the USBSUSPEND state. HC may move from the USBSUSPEND state to the USBRESUME state after detecting the resume signaling from a downstream port.</p> <p>HC enters USBSUSPEND after a software reset, whereas it enters USBRESET after a hardware reset. The latter also resets the Root</p>	00b	USBReset	01b	USBResume	10b	USBOperational	11b	USBSuspend
00b	USBReset											
01b	USBResume											
10b	USBOperational											
11b	USBSuspend											

Offset: 0x0404				Register Name: HcRevision				
Bit	Read/Write		Default/Hex	Description				
	HCD	HC						
				Hub and asserts subsequent reset signaling to downstream ports.				
5	R/W	R	0x0	<p><b>BulkListEnable</b></p> <p>This bit is set to enable the processing of the Bulk list in the next Frame. If cleared by HCD, the processing of the Bulk list does not occur after the next SOF. HC checks this bit whenever it determines to process the list. When disabled, HCD may modify the list. If <i>HcBulkCurrentED</i> is pointing to an ED to be removed, HCD must advance the pointer by updating <i>HcBulkCurrentED</i> before re-enabling processing of the list.</p>				
4	R/W	R	0x0	<p><b>ControlListEnable</b></p> <p>This bit is set to enable the processing of the Control list in the next Frame. If cleared by HCD, the processing of the Control list does not occur after the next SOF. HC must check this bit whenever it determines to process the list. When disabled, HCD may modify the list. If <i>HcControlCurrentED</i> is pointing to an ED to be removed, HCD must advance the pointer by updating <i>HcControlCurrentED</i> before re-enabling processing of the list.</p>				
3	R/W	R	0x0	<p><b>IsochronousEnable</b></p> <p>This bit is used by HCD to enable/disable processing of isochronous EDs. While processing the periodic list in a Frame, HC checks the status of this bit when it finds an Isochronous ED (F=1). If set (enabled), HC continues processing the EDs. If cleared (disabled), HC halts processing of the periodic list (which now contains only isochronous EDs) and begins processing the Bulk/Control lists. Setting this bit is guaranteed to take effect in the next Frame (not the current Frame).</p>				
2	R/W	R	0x0	<p><b>PeriodicListEnable</b></p> <p>This bit is set to enable the processing of periodic list in the next Frame. If cleared by HCD, processing of the periodic list does not occur after the next SOF. HC must check this bit before it starts processing the list.</p>				
1:0	R/W	R	0x0	<p><b>ControlBulkServiceRatio</b></p> <p>This specifies the service ratio between Control and Bulk EDs. Before processing any of the nonperiodic lists, HC must compare the ratio specified with its internal count on how many nonempty Control EDs have been processed, in determining whether to continue serving another Control ED or switching to Bulk EDs. The internal count will be retained when crossing the frame boundary. In case of reset, HCD is responsible for restoring this value.</p> <table border="1"> <tr> <td>CBSR</td><td>No. of Control EDs Over Bulk EDs Served</td></tr> <tr> <td>0</td><td>1:1</td></tr> </table>	CBSR	No. of Control EDs Over Bulk EDs Served	0	1:1
CBSR	No. of Control EDs Over Bulk EDs Served							
0	1:1							

#### 8.6.6.2 0x0408 OHCI Command Status Register (Default Value: 0x0000\_0000)

Offset: 0x0408				Register Name: HcCommandStatus
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31:18	/	/	0x0	<b>Reserved</b>
17:16	R	R/W	0x0	<p><b>SchedulingOverrunCount</b></p> <p>These bits are incremented on each scheduling overrun error. It is initialized to 00b and wraps around at 11b. This will be incremented when a scheduling overrun is detected even if <b>SchedulingOverrun</b> in <i>HcInterruptStatus</i> has already been set. This is used by HCD to monitor any persistent scheduling problem.</p>
15:4	/	/	/	/
3	R/W	R/W	0x0	<p><b>OwnershipChangeRequest</b></p> <p>This bit is set by an OS HCD to request a change of control of the HC. When set HC will set the <b>OwnershipChange</b> field in <i>HcInterruptStatus</i>. After the changeover, this bit is cleared and remains so until the next request from OS HCD.</p>
2	R/W	R/W	0x0	<p><b>BulkListFilled</b></p> <p>This bit is used to indicate whether there are any TDs on the Bulk list. It is set by HCD whenever it adds a TD to an ED in the Bulk list. When HC begins to process the head of the Bulk list, it checks BLF. As long as <b>BulkListFilled</b> is 0, HC will not start processing the Bulk list. If <b>BulkListFilled</b> is 1, HC will start processing the Bulk list and will set BF to 0. If HC finds a TD on the list, then HC will set <b>BulkListFilled</b> to 1 causing the Bulk list processing to continue. If no TD is found on the Bulk list, and if HCD does not set <b>BulkListFilled</b>, then <b>BulkListFilled</b> will still be 0 when HC completes processing the Bulk list and Bulk list processing will stop.</p>
1	R/W	R/W	0x0	<p><b>ControlListFilled</b></p> <p>This bit is used to indicate whether there are any TDs on the Control list. It is set by HCD whenever it adds a TD to an ED in the Control list.</p> <p>When HC begins to process the head of the Control list, it checks</p>

Offset: 0x0408				Register Name: HcCommandStatus
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
				CLF. As long as <b>ControlListFilled</b> is 0, HC will not start processing the Control list. If CF is 1, HC will start processing the Control list and will set <b>ControlListFilled</b> to 0. If HC finds a TD on the list, then HC will set <b>ControlListFilled</b> to 1 causing the Control list processing to continue. If no TD is found on the Control list, and if the HCD does not set <b>ControlListFilled</b> , then <b>ControlListFilled</b> will still be 0 when HC completes processing the Control list and Control list processing will stop.
0	R/W	R/E	0x0	<p><b>HostControllerReset</b></p> <p>This bit is by HCD to initiate a software reset of HC. Regardless of the functional state of HC, it moves to the USBSuspend state in which most of the operational registers are reset except those stated otherwise; e.g, the InterruptRouting field of HcControl, and no Host bus accesses are allowed. This bit is cleared by HC upon the completion of the reset operation. The reset operation must be completed within 10 ms. This bit, when set, should not cause a reset to the Root Hub and no subsequent reset signaling should be asserted to its downstream ports.</p>

#### 8.6.6.3 0x040C OHCI Interrupt Status Register (Default Value: 0x0000\_0000)

Offset: 0x040C				Register Name: HcInterruptStatus
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31:7	/	/	/	/
6	R/W	R/W	0x0	<p><b>RootHubStatusChange</b></p> <p>This bit is set when the content of <i>HcRhStatus</i> or the content of any of <i>HcRhPortStatus</i>[<b>NumberofDownstreamPort</b>] has changed.</p>
5	R/W	R/W	0x0	<p><b>FrameNumberOverflow</b></p> <p>This bit is set when the MSb of <i>HcFmNumber</i> (bit 15) changes value, from 0 to 1 or from 1 to 0, and after <i>HccaFrameNumber</i> has been updated.</p>
4	R/W	R/W	0x0	<p><b>UnrecoverableError</b></p> <p>This bit is set when HC detects a system error not related to USB. HC should not proceed with any processing nor signaling before the system error has been corrected. HCD clears this bit after HC has been reset.</p>
3	R/W	R/W	0x0	<p><b>ResumeDetected</b></p> <p>This bit is set when HC detects that a device on the USB is asserting resume signaling. It is the transition from no resume signaling to</p>

Offset: 0x040C				Register Name: HcInterruptStatus
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
				resume signaling causing this bit to be set. This bit is not set when HCD sets the USBRseume state.
2	R/W	R/W	0x0	<p><b>StartofFrame</b></p> <p>This bit is set by HC at each start of frame and after the update of <i>HccaFrameNumber</i>. HC also generates a SOF token at the same time.</p>
1	R/W	R/W	0x0	<p><b>WritebackDoneHead</b></p> <p>This bit is set immediately after HC has written <i>HcDoneHead</i> to <i>HccaDoneHead</i>. Further updates of the <i>HccaDoneHead</i> will not occur until this bit has been cleared. HCD should only clear this bit after it has saved the content of <i>HccaDoneHead</i>.</p>
0	R/W	R/W	0x0	<p><b>SchedulingOverrun</b></p> <p>This bit is set when the USB schedule for the current Frame overruns and after the update of <i>HccaFrameNumber</i>. A scheduling overrun will also cause the <b>SchedulingOverrunCount</b> of <i>HcCommandStatus</i> to be incremented.</p>

#### 8.6.6.4 0x0410 OHCI Interrupt Enable Register (Default Value: 0x0000\_0000)

Offset: 0x0410				Register Name: HcInterruptEnable Register				
Bit	Read/Write		Default/Hex	Description				
	HCD	HC						
31	R/W	R	0x0	<p><b>MasterInterruptEnable</b></p> <p>A '0' written to this field is ignored by HC. A '1' written to this field enables interrupt generation due to events specified in the other bits of this register. This is used by HCD as Master Interrupt Enable.</p>				
30:7	/	/	/	/				
6	R/W	R	0x0	<p><b>RootHubStatusChange Interrupt Enable</b></p> <table border="1"> <tr> <td>0</td><td>Ignore;</td></tr> <tr> <td>1</td><td>Enable interrupt generation due to Root Hub Status Change;</td></tr> </table>	0	Ignore;	1	Enable interrupt generation due to Root Hub Status Change;
0	Ignore;							
1	Enable interrupt generation due to Root Hub Status Change;							
5	R/W	R	0x0	<p><b>FrameNumberOverflow Interrupt Enable</b></p> <table border="1"> <tr> <td>0</td><td>Ignore;</td></tr> <tr> <td>1</td><td>Enable interrupt generation due to Frame Number Over Flow;</td></tr> </table>	0	Ignore;	1	Enable interrupt generation due to Frame Number Over Flow;
0	Ignore;							
1	Enable interrupt generation due to Frame Number Over Flow;							
4	R/W	R	0x0	<p><b>UnrecoverableError Interrupt Enable</b></p> <table border="1"> <tr> <td>0</td><td>Ignore;</td></tr> <tr> <td>1</td><td>Enable interrupt generation due to Unrecoverable Error;</td></tr> </table>	0	Ignore;	1	Enable interrupt generation due to Unrecoverable Error;
0	Ignore;							
1	Enable interrupt generation due to Unrecoverable Error;							
3	R/W	R	0x0	<b>ResumeDetected Interrupt Enable</b>				

Offset: 0x0410				Register Name: HcInterruptEnable Register	
Bit	Read/Write		Default/Hex	Description	
	HCD	HC			
0				0	Ignore;
				1	Enable interrupt generation due to Resume Detected;
2	R/W	R	0x0	<b>StartofFrame Interrupt Enable</b>	
				0	Ignore;
				1	Enable interrupt generation due to Start of Flame;
1	R/W	R	0x0	<b>WritebackDoneHead Interrupt Enable</b>	
				0	Ignore;
				1	Enable interrupt generation due to Write back Done Head;
0	R/W	R	0x0	<b>SchedulingOverrun Interrupt Enable</b>	
				0	Ignore;
				1	Enable interrupt generation due to Scheduling Overrun;

#### 8.6.6.5 0x0414 OHCI Interrupt Disable Register (Default Value: 0x0000\_0000)

Offset: 0x0414				Register Name: HcInterruptDisable Register	
Bit	Read/Write		Default/Hex	Description	
	HCD	HC			
31	R/W	R	0x0	<b>MasterInterruptEnable</b>	A written ‘0’ to this field is ignored by HC. A ‘1’ written to this field disables interrupt generation due events specified in the other bits of this register. This field is set after a hardware or software reset.
30:7	/	/	/	/	
6	R/W	R	0x0	<b>RootHubStatusChange Interrupt Disable</b>	
				0	Ignore;
				1	Disable interrupt generation due to Root Hub Status Change;
5	R/W	R	0x0	<b>FrameNumberOverflow Interrupt Disable</b>	
				0	Ignore;
				1	Disable interrupt generation due to Frame Number Overflow;
4	R/W	R	0x0	<b>UnrecoverableError Interrupt Disable</b>	
				0	Ignore;
				1	Disable interrupt generation due to Unrecoverable Error;
3	R/W	R	0x0	<b>ResumeDetected Interrupt Disable</b>	
				0	Ignore;
				1	Disable interrupt generation due to Resume Detected;
2	R/W	R	0x0	<b>StartofFrame Interrupt Disable</b>	
				0	Ignore;
				1	Disable interrupt generation due to Start of Flame;

Offset: 0x0414				Register Name: HcInterruptDisable Register
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
1	R/W	R	0x0	<b>WritebackDoneHead Interrupt Disable</b>
				<table border="1"> <tr> <td>0</td><td>Ignore;</td></tr> <tr> <td>1</td><td>Disable interrupt generation due to Write back Done Head;</td></tr> </table>
0	Ignore;			
1	Disable interrupt generation due to Write back Done Head;			
0	R/w	R	0x0	<b>SchedulingOverrun Interrupt Disable</b>
				<table border="1"> <tr> <td>0</td><td>Ignore;</td></tr> <tr> <td>1</td><td>Disable interrupt generation due to Scheduling Overrun;</td></tr> </table>
0	Ignore;			
1	Disable interrupt generation due to Scheduling Overrun;			

#### 8.6.6.6 0x0418 OHCI HCCA Register (Default Value: 0x0000\_0000)

Offset: 0x0418				Register Name: HcHCCA
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31:8	R/W	R	0x0	<b>HCCA[31:8]</b> This is the base address of the Host Controller Communication Area. This area is used to hold the control structures and the Interrupt table that are accessed by both the Host Controller and the Host Controller Driver.
7:0	R	R	0x0	<b>HCCA[7:0]</b> The alignment restriction in HcHCCA register is evaluated by examining the number of zeros in the lower order bits. The minimum alignment is 256 bytes, therefore, bits 0 through 7 must always return 0 when read.

#### 8.6.6.7 0x041C OHCI Period Current ED Register (Default Value: 0x0000\_0000)

Offset: 0x041C				Register Name: HcPeriodCurrentED[PCED]
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31:4	R	R/W	0x0	<b>PCED[31:4]</b> This is used by HC to point to the head of one of the Periodic list which will be processed in the current Frame. The content of this register is updated by HC after a periodic ED has been processed. HCD may read the content in determining which ED is currently being processed at the time of reading.
3:0	R	R	0x0	<b>PCED[3:0]</b> Because the general TD length is 16 bytes, the memory structure for the TD must be aligned to a 16-byte boundary. So the lower bits in the PCED, through bit 0 to bit 3 must be zero in this field.

#### 8.6.6.8 0x0420 OHCI Control Head ED Register (Default Value: 0x0000\_0000)

Offset: 0x0420				Register Name: HcControlHeadED[CHED]
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31:4	R/W	R	0x0	<b>EHCD[31:4]</b> The HcControlHeadED register contains the physical address of the first Endpoint Descriptor of the Control list. HC traverse the Control list starting with the HcControlHeadED pointer. The content is loaded from HCCA during the initialization of HC.
3:0	R	R	0x0	<b>EHCD[3:0]</b> Because the general TD length is 16 bytes, the memory structure for the TD must be aligned to a 16-byte boundary. So the lower bits in the PCED, through bit 0 to bit 3 must be zero in this field.

#### 8.6.6.9 0x0424 OHCI Control Current ED Register (Default Value: 0x0000\_0000)

Offset: 0x0424				Register Name: HcControlCurrentED[CCED]
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31:4	R/W	R/W	0x0	<b>CCED[31:4]</b> The pointer is advanced to the next ED after serving the present one. HC will continue processing the list from where it left off in the last Frame. When it reaches the end of the Control list, HC checks the ControlListFilled of in HcCommandStatus. If set, it copies the content of HcControlHeadED to HcControlCurrentED and clears the bit. If not set, it does nothing. HCD is allowed to modify this register only when the ControlListEnable of HcControl is cleared. When set, HCD only reads the instantaneous value of this register. Initially, this is set to zero to indicate the end of the Control list.
3:0	R	R	0x0	<b>CCED[3:0]</b> Because the general TD length is 16 bytes, the memory structure for the TD must be aligned to a 16-byte boundary. So the lower bits in the PCED, through bit 0 to bit 3 must be zero in this field.

#### 8.6.6.10 0x0428 OHCI Bulk Head ED Register (Default Value: 0x0000\_0000)

Offset: 0x0428				Register Name: HcBulkHeadED[BHED]
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31:4	R/W	R	0x0	<b>BHED[31:4]</b> The HcBulkHeadED register contains the physical address of the first Endpoint Descriptor of the Bulk list. HC traverses the Bulk list starting with the HcBulkHeadED pointer. The content is loaded from HCCA during the initialization of HC.
3:0	R	R	0x0	<b>BHED[3:0]</b> Because the general TD length is 16 bytes, the memory structure for the TD must be aligned to a 16-byte boundary. So the lower bits in the PCED, through bit 0 to bit 3 must be zero in this field.

#### 8.6.6.11 0x042C OHCI Bulk Current ED Register (Default Value: 0x0000\_0000)

Offset: 0x042C				Register Name: HcBulkCurrentED[BCED]
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31:4	R/W	R/W	0x0	<b>BulkCurrentED[31:4]</b> This is advanced to the next ED after the HC has served the present one. HC continues processing the list from where it left off in the last Frame. When it reaches the end of the Bulk list, HC checks the <b>ControlListFilled</b> of HcControl. If set, it copies the content of <b>HcBulkHeadED</b> to <b>HcBulkCurrentED</b> and clears the bit. If it is not set, it does nothing. HCD is only allowed to modify this register when the <b>BulkListEnable</b> of <b>HcControl</b> is cleared. When set, the HCD only reads the instantaneous value of this register. This is initially set to zero to indicate the end of the Bulk list.
3:0	R	R	0x0	<b>BulkCurrentED [3:0]</b> Because the general TD length is 16 bytes, the memory structure for the TD must be aligned to a 16-byte boundary. So the lower bits in the PCED, through bit 0 to bit 3 must be zero in this field.

#### 8.6.6.12 0x0430 OHCI Done Head Register (Default Value: 0x0000\_0000)

Offset: 0x0430				Register Name: HcDoneHead
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31:4	R	R/W	0x0	<b>HcDoneHead[31:4]</b> When a TD is completed, HC writes the content of <b>HcDoneHead</b> to

Offset: 0x0430				Register Name: HcDoneHead
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
				the NextTD field of the TD. HC then overwrites the content of <i>HcDoneHead</i> with the address of this TD. This is set to zero whenever HC writes the content of this register to HCCA. It also sets the <b>WritebackDoneHead</b> of <i>HcInterruptStatus</i> .
3:0	R	R	0x0	<b>HcDoneHead[3:0]</b> Because the general TD length is 16 bytes, the memory structure for the TD must be aligned to a 16-byte boundary. So the lower bits in the PCED, through bit 0 to bit 3 must be zero in this field.

#### 8.6.6.13 0x0434 OHCI Frame Interval Register (Default Value: 0x0000\_2EDF)

Offset: 0x0434				Register Name: HcFmInterval Register
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31	R/W	R	0x0	<b>FrameIntervalToggler</b> HCD toggles this bit whenever it loads a new value to <b>FrameInterval</b> .
30:16	R/W	R	0x0	<b>FSLargestDataPacket</b> This field specifies a value which is loaded into the Largest Data Packet Counter at the beginning of each frame. The counter value represents the largest amount of data in bits which can be sent or received by the HC in a single transaction at any given time without causing scheduling overrun. The field value is calculated by the HCD.
15:14	/	/	/	/
13:0	R/W	R	0x2edf	<b>FrameInterval</b> This specifies the interval between two consecutive SOFs in bit times. The nominal value is set to be 11,999. HCD should store the current value of this field before resetting HC. By setting the <b>HostControllerReset</b> field of <i>HcCommandStatus</i> as this will cause the HC to reset this field to its nominal value. HCD may choose to restore the stored value upon the completion of the Reset sequence.

#### 8.6.6.14 0x0438 OHCI Frame Remaining Register (Default Value: 0x0000\_0000)

Offset: 0x0438				Register Name: HcFmRemaining
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31	R	R/W	0x0	<b>FrameRemaining Toggle</b> This bit is loaded from the <b>FrameIntervalToggle</b> field of <i>HcFmInterval</i> whenever <b>FrameRemaining</b> reaches 0. This bit is used by HCD for the synchronization between <b>FrameInterval</b> and <b>FrameRemaining</b> .
30:14	/	/	/	/
13:0	R	RW	0x0	<b>FramRemaining</b> This counter is decremented at each bit time. When it reaches zero, it is reset by loading the <b>FrameInterval</b> value specified in <i>HcFmInterval</i> at the next bit time boundary. When entering the <b>USBOPERATIONAL</b> state, HC re-loads the content with the <b>FrameInterval</b> of <i>HcFmInterval</i> and uses the updated value from the next SOF.

#### 8.6.6.15 0x043C OHCI Frame Number Register (Default Value: 0x0000\_0000)

Offset: 0x043C				Register Name: HcFmNumber
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31:16	/	/	/	/
15:0	R	R/W	0x0	<b>FrameNumber</b> This is incremented when <i>HcFmRemaining</i> is re-loaded. It will be rolled over to 0x0 after 0xffff. When entering the <b>USBOPERATIONAL</b> state, this will be incremented automatically. The content will be written to HCCA after HC has incremented the <b>FrameNumber</b> at each frame boundary and sent a SOF but before HC reads the first ED in that Frame. After writing to HCCA, HC will set the <b>StartofFrame</b> in <i>HcInterruptStatus</i> .

#### 8.6.6.16 0x0440 OHCI Periodic Start Register (Default Value: 0x0000\_0000)

Offset: 0x0440				Register Name: HcPeriodicStatus
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31:14	/	/	/	/
13:0	R/W	R	0x0	<b>PeriodicStart</b> After a hardware reset, this field is cleared. This is then set by HCD

Offset: 0x0440				Register Name: HcPeriodicStatus
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
				during the HC initialization. The value is calculated roughly as 10% off from <i>HcFmInterval</i> . A typical value will be 0x2A3F (or 0x3e67). When <i>HcFmRemaining</i> reaches the value specified, processing of the periodic lists will have priority over Control/Bulk processing. HC will therefore start processing the Interrupt list after completing the current Control or Bulk transaction that is in progress.

#### 8.6.6.17 0x0444 OHCI LS Threshold Register (Default Value: 0x0000\_0628)

Offset: 0x0444				Register Name: HcLSThreshold
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31:12	/	/	/	/
11:0	R/W	R	0x0628	<p><b>LSThreshold</b></p> <p>This field contains a value which is compared to the <b>FrameRemaining</b> field prior to initiating a Low Speed transaction. The transaction is started only if <b>FrameRemaining</b> is less than or equal to this field. The value is calculated by HCD with the consideration of transmission and setup overhead.</p>

#### 8.6.6.18 0x0448 OHCI Root Hub DescriptorA Register (Default Value: 0x0200\_1201)

Offset: 0x0448				Register Name: HcRhDescriptorA		
Bit	Read/Write		Default/Hex	Description		
	HCD	HC				
31:24	R/W	R	0x2	<p><b>PowerOnToPowerGoodTime[POTPGT]</b></p> <p>This byte specifies the duration HCD has to wait before accessing a powered-on port of the Root Hub. It is implementation-specific. The unit of time is 2 ms. The duration is calculated as POTPGT * 2ms.</p>		
23:13	/	/	/	/		
12	R/W	R	0x1	<p><b>NoOverCurrentProtection</b></p> <p>This bit describes how the overcurrent status for the Root Hub ports are reported. When this bit is cleared, the <b>OverCurrentProtectionMode</b> field specifies global or per-port reporting.</p> <table border="1"> <tr> <td><b>0</b></td><td>Over-current status is reported collectively for all downstream ports.</td></tr> </table>	<b>0</b>	Over-current status is reported collectively for all downstream ports.
<b>0</b>	Over-current status is reported collectively for all downstream ports.					

Offset: 0x0448				Register Name: HcRhDescriptorA	
Bit	Read/Write		Default/Hex	Description	
	HCD	HC			
				1	No overcurrent protection supported.
11	R/W	R	0x0	<b>OverCurrentProtectionMode</b> This bit describes how the overcurrent status for the Root Hub ports are reported. At reset, these fields should reflect the same mode as <b>PowerSwitchingMode</b> . This field is valid only if the <b>NoOverCurrentProtection</b> field is cleared.	
				0	Over-current status is reported collectively for all downstream ports.
				1	Over-current status is reported on per-port basis.
10	R	R	0x0	<b>Device Type</b> This bit specifies that the Root Hub is not a compound device. The Root Hub is not permitted to be a compound device. This field should always read/write 0.	
9	R/W	R	0x1	<b>PowerSwitchingMode</b> This bit is used to specify how the power switching of the Root Hub ports is controlled. It is implementation-specific. This field is only valid if the <b>NoPowerSwitching</b> field is cleared.	
				0	All ports are powered at the same time.
8	R/W	R	0x0	<b>NoPowerSwitching</b> These bits are used to specify whether power switching is supported or ports are always powered. It is implementation-specific. When this bit is cleared, the <b>PowerSwitchingMode</b> specifies global or per-port switching.	
				0 Ports are power switched.	
				1 Ports are always powered on when the HC is powered on.	
7:0	R	R	0x01	<b>NumberDownstreamPorts</b> These bits specify the number of downstream ports supported by the Root Hub. It is implementation-specific. The minimum number of ports is 1. The maximum number of ports supported.	

#### 8.6.6.19 0x044C HcRhDescriptorB Register (Default Value: 0x0000\_0000)

Offset: 0x044C				Register Name: HcRhDescriptorB Register										
Bit	Read/Write		Default/Hex	Description										
	HCD	HC												
31:16	R/W	R	0x0	<p><b>PortPowerControlMask</b></p> <p>Each bit indicates if a port is affected by a global power control command when <b>PowerSwitchingMode</b> is set. When set, the port's power state is only affected by per-port power control (<b>Set/ClearPortPower</b>). When cleared, the port is controlled by the global power switch (<b>Set/ClearGlobalPower</b>). If the device is configured to global switching mode (<b>PowerSwitchingMode</b> = 0 ), this field is not valid.</p> <table border="1"> <tr> <td>Bit0</td><td>Reserved</td></tr> <tr> <td>Bit1</td><td>Ganged-power mask on Port #1.</td></tr> <tr> <td>Bit2</td><td>Ganged-power mask on Port #2.</td></tr> <tr> <td>...</td><td></td></tr> <tr> <td>Bit15</td><td>Ganged-power mask on Port #15.</td></tr> </table>	Bit0	Reserved	Bit1	Ganged-power mask on Port #1.	Bit2	Ganged-power mask on Port #2.	...		Bit15	Ganged-power mask on Port #15.
Bit0	Reserved													
Bit1	Ganged-power mask on Port #1.													
Bit2	Ganged-power mask on Port #2.													
...														
Bit15	Ganged-power mask on Port #15.													
15:0	R/W	R	0x0	<p><b>DeviceRemovable</b></p> <p>Each bit is dedicated to a port of the Root Hub. When cleared, the attached device is removable. When set, the attached device is not removable.</p> <table border="1"> <tr> <td>Bit0</td><td>Reserved</td></tr> <tr> <td>Bit1</td><td>Device attached to Port #1.</td></tr> <tr> <td>Bit2</td><td>Device attached to Port #2.</td></tr> <tr> <td>...</td><td></td></tr> <tr> <td>Bit15</td><td>Device attached to Port #15.</td></tr> </table>	Bit0	Reserved	Bit1	Device attached to Port #1.	Bit2	Device attached to Port #2.	...		Bit15	Device attached to Port #15.
Bit0	Reserved													
Bit1	Device attached to Port #1.													
Bit2	Device attached to Port #2.													
...														
Bit15	Device attached to Port #15.													

#### 8.6.6.20 0x0450 HcRhStatus Register (Default Value: 0x0000\_0000)

Offset: 0x0450				Register Name: HcRhStatus Register
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31	W	R	0x0	<p><b>(write)ClearRemoteWakeupEnable</b></p> <p>Write a '1' clears <b>DeviceRemoteWakeupEnable</b>. Writing a '0' has no effect.</p>
30:18	/	/	/	/
17	R/W	R	0x0	<p><b>OverCurrentIndicatorChang</b></p> <p>This bit is set by hardware when a change has occurred to the <b>OverCurrentIndicator</b> field of this register. The HCD clears this bit by writing a '1'. Writing a '0' has no effect.</p>
16	R/W	R	0x0	<b>(read)LocalPowerStartusChange</b>

Offset: 0x0450			Register Name: HcRhStatus Register					
Bit	Read/Write		Default/Hex	Description				
	HCD	HC						
				<p>The Root Hub does not support the local power status features, thus, this bit is always read as '0'.</p> <p><b>(write)SetGlobalPower</b></p> <p>In global power mode (<b>PowerSwitchingMode=0</b>), This bit is written to '1' to turn on power to all ports (clear <b>PortPowerStatus</b>). In per-port power mode, it sets <b>PortPowerStatus</b> only on ports whose <b>PortPowerControlMask</b> bit is not set. Writing a '0' has no effect.</p>				
15	R/W	R	0x0	<p><b>(read)DeviceRemoteWakeupEnable</b></p> <p>This bit enables a <b>ConnectStatusChange</b> bit as a resume event, causing a USBSUSPEND to USBRESUME state transition and setting the <b>ResumeDetected</b> interrupt.</p> <table border="1"> <tr> <td>0</td><td><b>ConnectStatusChange</b> is not a remote wakeup event.</td></tr> <tr> <td>1</td><td><b>ConnectStatusChange</b> is a remote wakeup event.</td></tr> </table> <p><b>(write)SetRemoteWakeupEnable</b></p> <p>Writing a '1' sets DeviceRemoveWakeupEnable. Writing a '0' has no effect.</p>	0	<b>ConnectStatusChange</b> is not a remote wakeup event.	1	<b>ConnectStatusChange</b> is a remote wakeup event.
0	<b>ConnectStatusChange</b> is not a remote wakeup event.							
1	<b>ConnectStatusChange</b> is a remote wakeup event.							
14:2	/	/	/	/				
1	R	R/W	0x0	<p><b>OverCurrentIndicator</b></p> <p>This bit reports overcurrent conditions when the global reporting is implemented. When set, an overcurrent condition exists. When cleared, all power operations are normal.</p> <p>If per-port overcurrent protection is implemented this bit is always '0'</p>				
0	R/W	R	0x0	<p><b>(Read)LocalPowerStatus</b></p> <p>When read, this bit returns the LocalPowerStatus of the Root Hub. The Root Hub does not support the local power status feature; thus, this bit is always read as '0'.</p> <p><b>(Write)ClearGlobalPower</b></p> <p>When write, this bit is operated as the ClearGlobalPower. In global power mode (<b>PowerSwitchingMode=0</b>), This bit is written to '1' to turn off power to all ports (clear <b>PortPowerStatus</b>). In per-port power mode, it clears <b>PortPowerStatus</b> only on ports whose <b>PortPowerControlMask</b> bit is not set. Writing a '0' has no effect.</p>				

## 8.6.6.21 0x0454 HcRhPortStatus Register (Default Value: 0x0000\_0100)

Offset: 0x0454				Register Name: HcRhPortStatus				
Bit	Read/Write		Default/Hex	Description				
	HCD	HC						
31:21	/	/	/	/				
20	R/W	R/W	0x0	<p><b>PortResetStatusChange</b>  This bit is set at the end of the 10-ms port reset signal. The HCD writes a '1' to clear this bit. Writing a '0' has no effect.</p> <table border="1"> <tr> <td>0</td><td>port reset is not complete</td></tr> <tr> <td>1</td><td>port reset is complete</td></tr> </table>	0	port reset is not complete	1	port reset is complete
0	port reset is not complete							
1	port reset is complete							
19	R/W	R/W	0x0	<p><b>PortOverCurrentIndicatorChange</b>  This bit is valid only if overcurrent conditions are reported on a per-port basis. This bit is set when Root Hub changes the <b>PortOverCurrentIndicator</b> bit. The HCD writes a '1' to clear this bit. Writing a '0' has no effect.</p> <table border="1"> <tr> <td>0</td><td>no change in <b>PortOverCurrentIndicator</b></td></tr> <tr> <td>1</td><td><b>PortOverCurrentIndicator</b> has changed</td></tr> </table>	0	no change in <b>PortOverCurrentIndicator</b>	1	<b>PortOverCurrentIndicator</b> has changed
0	no change in <b>PortOverCurrentIndicator</b>							
1	<b>PortOverCurrentIndicator</b> has changed							
18	R/W	R/W	0x0	<p><b>PortSuspendStatusChange</b>  This bit is set when the full resume sequence has been completed. This sequence includes the 20-s resume pulse, LS EOP, and 3-ms resynchronization delay. The HCD writes a '1' to clear this bit. Writing a '0' has no effect. This bit is also cleared when <b>ResetStatusChange</b> is set.</p> <table border="1"> <tr> <td>0</td><td>resume is not completed</td></tr> <tr> <td>1</td><td>resume completed</td></tr> </table>	0	resume is not completed	1	resume completed
0	resume is not completed							
1	resume completed							
17	R/W	R/W	0x0	<p><b>PortEnableStatusChange</b>  This bit is set when hardware events cause the <b>PortEnableStatus</b> bit to be cleared. Changes from HCD writes do not set this bit. The HCD writes a '1' to clear this bit. Writing a '0' has no effect.</p> <table border="1"> <tr> <td>0</td><td>no change in <b>PortEnableStatus</b></td></tr> <tr> <td>1</td><td>change in <b>PortEnableStatus</b></td></tr> </table>	0	no change in <b>PortEnableStatus</b>	1	change in <b>PortEnableStatus</b>
0	no change in <b>PortEnableStatus</b>							
1	change in <b>PortEnableStatus</b>							
16	R/W	R/W	0x0	<p><b>ConnectStatusChange</b>  This bit is set whenever a connect or disconnect event occurs. The HCD writes a '1' to clear this bit. Writing a '0' has no effect. If <b>CurrentConnectStatus</b> is cleared when a <b>SetPortReset</b>, <b>SetPortEnable</b>, or <b>SetPortSuspend</b> write occurs, this bit is set to force the driver to re-evaluate the connection status since these writes should not occur if the port is disconnected.</p> <table border="1"> <tr> <td>0</td><td>no change in <b>PortEnableStatus</b></td></tr> <tr> <td>1</td><td>change in <b>PortEnableStatus</b></td></tr> </table> <p><b>Note:</b> If the DeviceRemovable[NDP] bit is set, this bit is set only</p>	0	no change in <b>PortEnableStatus</b>	1	change in <b>PortEnableStatus</b>
0	no change in <b>PortEnableStatus</b>							
1	change in <b>PortEnableStatus</b>							

Offset: 0x0454				Register Name: HcRhPortStatus				
Bit	Read/Write		Default/Hex	Description				
	HCD	HC						
				after a Root Hub reset to inform the system that the device is attached.				
15:10	/	/	/	/				
9	R/W	R/W	0x0	<p>(read)<b>LowSpeedDeviceAttached</b>  This bit indicates the speed of the device attached to this port. When set, a Low Speed device is attached to this port. When clear, a Full Speed device is attached to this port. This field is valid only when the <b>CurrentConnectStatus</b> is set.</p> <table border="1"> <tr> <td>0</td><td>full speed device attached</td></tr> <tr> <td>1</td><td>low speed device attached</td></tr> </table> <p>(write)<b>ClearPortPower</b>  The HCD clears the <b>PortPowerStatus</b> bit by writing a '1' to this bit. Writing a '0' has no effect.</p>	0	full speed device attached	1	low speed device attached
0	full speed device attached							
1	low speed device attached							
8	R/W	R/W	0x1	<p>(read)<b>PortPowerStatus</b>  This bit reflects the port's power status, regardless of the type of power switching implemented. This bit is cleared if an overcurrent condition is detected. HCD sets this bit by writing <b>SetPortPower</b> or <b>SetGlobalPower</b>. HCD clears this bit by writing <b>ClearPortPower</b> or <b>ClearGlobalPower</b>. Which power control switches are enabled is determined by <b>PowerSwitchingMode</b> and <b>PortPortControlMask[NumberDownstreamPort]</b>. In global switching mode(<b>PowerSwitchingMode=0</b>), only <b>Set/ClearGlobalPower</b> controls this bit. In per-port power switching (<b>PowerSwitchingMode=1</b>), if the <b>PortPowerControlMask[NDP]</b> bit for the port is set, only <b>Set/ClearPortPower</b> commands are enabled. If the mask is not set, only <b>Set/ClearGlobalPower</b> commands are enabled. When port power is disabled, <b>CurrentConnectStatus</b>, <b>PortEnableStatus</b>, <b>PortSuspendStatus</b>, and <b>PortResetStatus</b> should be reset.</p> <table border="1"> <tr> <td>0</td><td>port power is off</td></tr> <tr> <td>1</td><td>port power is on</td></tr> </table> <p>(write)<b>SetPortPower</b>  The HCD writes a '1' to set the <b>PortPowerStatus</b> bit. Writing a '0' has no effect.</p> <p><b>Note:</b> This bit is always reads '1b' if power switching is not supported.</p>	0	port power is off	1	port power is on
0	port power is off							
1	port power is on							
7:5	/	/	/	/				
4	R/W	R/W	0x0	(read) <b>PortResetStatus</b>				

Offset: 0x0454			Register Name: HcRhPortStatus					
Bit	Read/Write		Default/Hex	Description				
	HCD	HC						
				<p>When this bit is set by a write to <b>SetPortReset</b>, port reset signaling is asserted. When reset is completed, this bit is cleared when <b>PortResetStatusChange</b> is set. This bit cannot be set if <b>CurrentConnectStatus</b> is cleared.</p> <table border="1"> <tr> <td>0</td><td>port reset signal is not active</td></tr> <tr> <td>1</td><td>port reset signal is active</td></tr> </table> <p>(write)<b>SetPortReset</b>  The HCD sets the port reset signaling by writing a '1' to this bit. Writing a '0' has no effect. If <b>CurrentConnectStatus</b> is cleared, this write does not set <b>PortResetStatus</b>, but instead sets <b>ConnectStatusChange</b>. This informs the driver that it attempted to reset a disconnected port.</p>	0	port reset signal is not active	1	port reset signal is active
0	port reset signal is not active							
1	port reset signal is active							
3	R/W	R/W	0x0	<p>(read)<b>PortOverCurrentIndicator</b>  This bit is only valid when the Root Hub is configured in such a way that overcurrent conditions are reported on a per-port basis. If per-port overcurrent reporting is not supported, this bit is set to 0. If cleared, all power operations are normal for this port. If set, an overcurrent condition exists on this port. This bit always reflects the overcurrent input signal.</p> <table border="1"> <tr> <td>0</td><td>no overcurrent condition.</td></tr> <tr> <td>1</td><td>overcurrent condition detected.</td></tr> </table> <p>(write)<b>ClearSuspendStatus</b>  The HCD writes a '1' to initiate a resume. Writing a '0' has no effect. A resume is initiated only if <b>PortSuspendStatus</b> is set.</p>	0	no overcurrent condition.	1	overcurrent condition detected.
0	no overcurrent condition.							
1	overcurrent condition detected.							
2	R/W	R/W	0x0	<p>(read)<b>PortSuspendStatus</b>  This bit indicates the port is suspended or in the resume sequence. It is set by a <b>SetSuspendState</b> write and cleared when <b>PortSuspendStatusChange</b> is set at the end of the resume interval. This bit cannot be set if <b>CurrentConnectStatus</b> is cleared. This bit is also cleared when <b>PortResetStatusChange</b> is set at the end of the port reset or when the HC is placed in the USBRESUME state. If an upstream resume is in progress, it should propagate to the HC.</p> <table border="1"> <tr> <td>0</td><td>port is not suspended</td></tr> <tr> <td>1</td><td>port is suspended</td></tr> </table> <p>(write)<b>SetPortSuspend</b>  The HCD sets the <b>PortSuspendStatus</b> bit by writing a '1' to this bit. Writing a '0' has no effect. If <b>CurrentConnectStatus</b> is cleared, this</p>	0	port is not suspended	1	port is suspended
0	port is not suspended							
1	port is suspended							

Offset: 0x0454			Register Name: HcRhPortStatus					
Bit	Read/Write		Default/Hex	Description				
	HCD	HC						
				write does not set <b>PortSuspendStatus</b> ; instead it sets <b>ConnectStatusChange</b> . This informs the driver that it attempted to suspend a disconnected port.				
1	R/W	R/W	0x0	<p>(read)<b>PortEnableStatus</b></p> <p>This bit indicates whether the port is enabled or disabled. The Root Hub may clear this bit when an overcurrent condition, disconnect event, switched-off power, or operational bus error such as babble is detected. This change also causes <b>PortEnabledStatusChange</b> to be set. HCD sets this bit by writing <b>SetPortEnable</b> and clears it by writing <b>ClearPortEnable</b>. This bit cannot be set when <b>CurrentConnectStatus</b> is cleared. This bit is also set, if not already, at the completion of a port reset when <b>ResetStatusChange</b> is set or port suspend when <b>SuspendStatusChange</b> is set.</p> <table border="1"> <tr> <td>0</td><td>port is disabled</td></tr> <tr> <td>1</td><td>port is enabled</td></tr> </table> <p>(write)<b>SetPortEnable</b></p> <p>The HCD sets <b>PortEnableStatus</b> by writing a '1'. Writing a '0' has no effect. If <b>CurrentConnectStatus</b> is cleared, this write does not set <b>PortEnableStatus</b>, but instead sets <b>ConnectStatusChange</b>. This informs the driver that it attempted to enable a disconnected Port.</p>	0	port is disabled	1	port is enabled
0	port is disabled							
1	port is enabled							
0	R/W	R/W	0x0	<p>(read)<b>CurrentConnectStatus</b></p> <p>This bit reflects the current state of the downstream port.</p> <table border="1"> <tr> <td>0</td><td>No device connected</td></tr> <tr> <td>1</td><td>Device connected</td></tr> </table> <p>(write)<b>ClearPortEnable</b></p> <p>The HCD writes a '1' to clear the PortEnableStatus bit. Writing '0' to this bit has no effect. The CurrentConnectStatus is not affected by any write.</p> <p><b>Note:</b> This bit is always read '1' when the attached device is nonremovable (DviceRemoveable[NumberDownstreamPort]).</p>	0	No device connected	1	Device connected
0	No device connected							
1	Device connected							

#### 8.6.6.22 0x0800 HCI Interface Register (Default Value: 0x1000\_0000)

Offset: 0x0800			Register Name: USB_CTRL
Bit	Read/Write	Default/Hex	Description
31:29	/	/	Reserved

Offset: 0x0800			Register Name: USB_CTRL
Bit	Read/Write	Default/Hex	Description
28	R	1	DMA Transfer Status Enable 0: Disable 1: Enable
27:26	/	/	/
25	R/W	0	OHCI count select 1: Simulation mode. The counters will be much shorter than real time 0: Normal mode. The counters will count full time
24:19	/	/	/
18	R/W	0	1: Within 2 us of the resume-K to SEO transition 0: Random time value of the resume-K to SEO transition
17:13	/	/	/
12	R/W	0	PP2VBUS 1: ULPI wrapper interface will automatically set or clear DrvVbus register in ULPI PHY according to the port power status from the root hub 0: ULPI wrapper will ignore the difference between power status of root hub and ULPI PHY
11	R/W	0	AHB Master interface INCR16 enable 1: Use INCR16 when appropriate 0: Do not use INCR16, use other enabled INCRX or unspecified length burst INCR
10	R/W	0	AHB Master interface INCR8 enable 1: Use INCR8 when appropriate 0: Do not use INCR8, use other enabled INCRX or unspecified length burst INCR
9	R/W	0	AHB Master interface burst type INCR4 enable 1: Use INCR4 when appropriate 0: Do not use INCR4, use other enabled INCNX or unspecified length burst INCR
8	R/W	0	AHB Master interface INCNX align enable 1: Start INCNx burst only on burst x-align address 0: Start burst on any double word boundary Note: This bit must enable if any bit of bit[11:9] is enabled
7:1	/	/	/
0	R/W	0	ULPI bypass enable 1: Enable UTMI interface, disable ULPI interface 0: Enable ULPI interface, disable UTMI interface

#### 8.6.6.23 0x0808 HCI Control 3 Register (Default Value: 0x0001\_0000)

Offset: 0x0808			Register Name: HCI_CTRL3
Bit	Read/Write	Default/Hex	Description
31:17	/	/	Reserved
16	R/W1C	1	Linestate Change Detect 0: Linestate change not detected 1: Linestate change detected Write '1' to clear.
15:4	/	/	Reserved
3	R/W	0	Remote Wakeup Enable 1: Enable 0: Disable
2	/	/	Reserved
1	R/W	0	Linestate Change Interrupt Enable 1: Enable 0: Disable
0	R/W	0	Linestate Change Detect Enable 1: Enable 0: Disable

#### 8.6.6.24 0x0810 PHY Control Register (Default Value: 0x0000\_0008)

Offset: 0x0810			Register Name: PHY_CTRL
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0	bist_en_a
15:8	R/W	0	vc_addr
7	R/W	0	vc_di
6:4	/	/	/
3	R/W	0x1	SIDDQ 1: Write 1 to disable phy 0: Write 0 to enable phy
2:1	/	/	/
0	R/W	0x0	vc_clk

**8.6.6.25 0x0824 PHY Status Register (Default Value: 0x0000\_0000)**

Offset: 0x0824			Register Name: PHY_STATUS
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
17	R	0	Bist_error
16	R	0	bist_done
15:1	/	/	/
0	R	0	vc_do

**8.6.6.26 0x0828 HCI SIE Port Disable Control Register (Default Value: 0x0000\_0000)**

Offset: 0x0828			Register Name: USB_SPDCR
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0	SE0 Status This bit is set when no-se0 is detected before SOF when bit[1:0] is 10b or 11b
15:5	/	/	/
4	R/W	0	resume_sel When set k-se0 transition 2 us, setting this bit to 1, which is cooperated with ss_utmi_backward_enb_i.
3:2	/	/	/
1:0	R/W	0	Port Disable Control 00: Port Disable when no-se0 detect before SOF 01: Port Disable when no-se0 detect before SOF 10: No Port Disable when no-se0 detect before SOF 11: Port Disable when no-se0 3 time detect before SOF during 8 frames

## 8.7 GPIO

### 8.7.1 Overview

The general purpose input/output (GPIO) is one of the blocks controlling the chip multiplexing pins. The H133 supports 5 groups of GPIO pins. Each pin can be configured as input or output and these pins are used to generate input signals or output signals for special purposes.

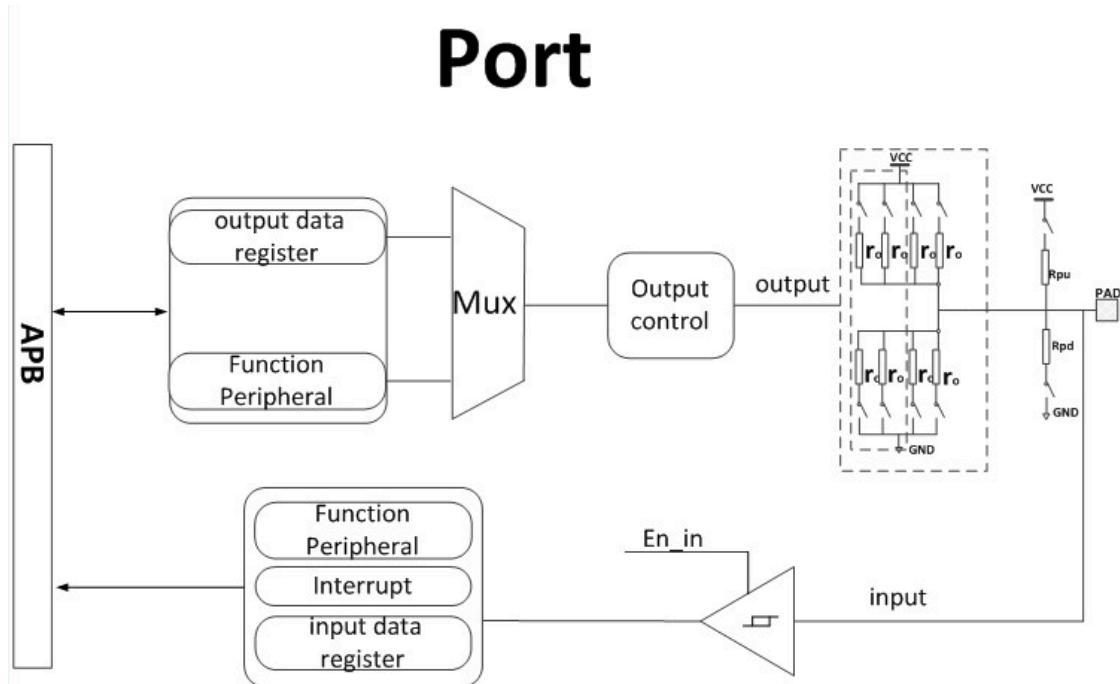
The Port Controller has the following features:

- 5 groups of ports (PB, PC, PD, PF, PG)
- Software control for each signal pin
- Data input (capture)/output (drive)
- Each GPIO peripheral can produce an interrupt
- Pull-up/Pull-down/no-Pull register control
- Control the direction of every signal
- 4 drive strengths in each operating mode
- Up to 49 interrupts
- Configurable interrupt edges

### 8.7.2 Block Diagram

The following figure shows the block diagram of the GPIO.

Figure 8-67 GPIO Block Diagram



The GPIO consists of the digital part (GPIO, external interface) and IO analog part (output buffer, dual pull down, pad). The digital part can select the output interface by the MUX switch; the analog part can configure pull up/down and buffer strength.

When executing GPIO read state, the GPIO reads the current level of the pin into the internal register bus.

When not executing GPIO read state, the external pin and the internal register bus are off-status, which is high-impedance.

### 8.7.3 Functional Description

#### 8.7.3.1 Multi-function Port

The H133 includes 49 multi-functional input/output port pins. There are 5 ports as listed below.

Table 8-20 Multi-function Port

Port Name	Number of Pins	Input Driver	Output Driver	Multiplex Pins	Power
PB	7	Schmitt	CMOS	DMIC/OWA/I2S/TWI/PWM/IR/SPI/DBI/UART/PB-EINT	3.3 V
PC	6	Schmitt	CMOS	SPI/SMHC/UART/BOOT/TWI/TCON/PC-EINT	3.3 V/ 1.8 V

Port Name	Number of Pins	Input Driver	Output Driver	Multiplex Pins	Power
PD	10	Schmitt	CMOS	LCD/LVDS /TWI/ DSI/ UART/PWM /PD-EINT	3.3 V/ 1.8 V
PF	7	Schmitt	CMOS	SMHC/JTAG/UART/OWA/TWI/IR/I2S/PWM /PF-EINT	3.3 V/ 1.8 V
PG	19	Schmitt	CMOS	SMHC/UART/PWM/I2S/TWI/EMAC/OWA/ IR/TCON/LEDC/SPI/PG-EINT	3.3 V/ 1.8 V

### 8.7.3.2 GPIO Multiplex Function

Table 8-21 to Table 8-26 show the multiplex function pins of the H133.



For each GPIO, Function0 is input function; Function1 is output function; Function9 to Function13 are reserved.

**Table 8-21 PB Multiplex Function**

GPIO Port	Function 2	Function 3	Function 4	Function 5	Function 6	Function 7	Function 8	Function 14
PB0	PWM3	IR-TX	TWI2-SCK	SPI1-WP/DBI-TE	UART0-TX	UART2-TX	OWA-OUT	PB-EINT0
PB1	PWM4	I2S2-DOUT3	TWI2-SDA	I2S2-DIN3	UART0-RX	UART2-RX	IR-RX	PB-EINT1
PB8	DMIC-DATA3	PWM5	TWI2-SCK	SPI1-HOLD/ DBI-DCX/ DBI-WRX	UART0-TX	UART1-TX		PB-EINT8
PB9	DMIC-DATA2	PWM6	TWI2-SDA	SPI1-MISO/ DBI-SDI/ DBI-TE/ DBI-DCX	UART0-RX	UART1-RX		PB-EINT9
PB10	DMIC-DATA1	PWM7	TWI0-SCK	SPI1-MOSI/ DBI-SDO	CLK-FANOUT0	UART1-RTS		PB-EINT10
PB11	DMIC-DATA0	PWM2	TWI0-SDA	SPI1-CLK/ DBI-SCLK	CLK-FANOUT1	UART1-CTS		PB-EINT11
PB12	DMIC-CLK	PWM0		SPI1-CS/ DBI-CSX	CLK-FANOUT2	IR-RX		PB-EINT12

**Table 8-22 PC Multiplex Function**

GPIO Port	Function 2	Function 3	Function 4	Function 5	Function 6	Function 7	Function 8	Function 14
PC2	SPI0-CLK	SDC2-CLK						PC-EINT2
PC3	SPI0-CS0	SDC2-CMD						PC-EINT3
PC4	SPI0-MOSI	SDC2-D2	BOOT-SEL0					PC-EINT4
PC5	SPI0-MISO	SDC2-D1	BOOT-SEL1					PC-EINT5
PC6	SPI0-WP	SDC2-D0	UART3-TX	TWI3-SCK	DBG-CLK			PC-EINT6
PC7	SPI0-HOLD	SDC2-D3	UART3-RX	TWI3-SDA	TCON-TRIG			PC-EINT7

**Table 8-23 PD Multiplex Function**

GPIO Port	Function 2	Function 3	Function 4	Function 5	Function 6	Function 7	Function 8	Function 14
PD0	LCD0-D2	LVDS0-V0P	DSI-D0P	TWI0-SCK				PD-EINT0
PD1	LCD0-D3	LVDS0-V0N	DSI-D0N	UART2-TX				PD-EINT1
PD2	LCD0-D4	LVDS0-V1P	DSI-D1P	UART2-RX				PD-EINT2
PD3	LCD0-D5	LVDS0-V1N	DSI-D1N	UART2-RTS				PD-EINT3
PD4	LCD0-D6	LVDS0-V2P	DSI-CKP	UART2-CTS				PD-EINT4

GPIO Port	Function 2	Function 3	Function 4	Function 5	Function 6	Function 7	Function 8	Function 14
PD5	LCD0-D7	LVDS0-V2N	DSI-CKN	UART5-TX				PD-EINT5
PD6	LCD0-D10	LVDS0-CKP	DSI-D2P	UART5-RX				PD-EINT6
PD7	LCD0-D11	LVDS0-CKN	DSI-D2N	UART4-TX				PD-EINT7
PD8	LCD0-D12	LVDS0-V3P	DSI-D3P	UART4-RX				PD-EINT8
PD9	LCD0-D13	LVDS0-V3N	DSI-D3N	PWM6				PD-EINT9

Table 8-24 PF Multiplex Function

GPIO Port	Function 2	Function 3	Function 4	Function 5	Function 6	Function 7	Function 8	Function 14
PF0	SDC0-D1	JTAG-MS		I2S2-DOUT1	I2S2-DIN0			PF-EINT0
PF1	SDC0-D0	JTAG-DI		I2S2-DOUT0	I2S2-DIN1			PF-EINT1
PF2	SDC0-CLK	UART0-TX	TWI0-SCK					PF-EINT2
PF3	SDC0-CMD	JTAG-DO		I2S2-BCLK				PF-EINT3
PF4	SDC0-D3	UART0-RX	TWI0-SDA	PWM6	IR-TX			PF-EINT4
PF5	SDC0-D2	JTAG-CK		I2S2-LRCK				PF-EINT5
PF6		OWA-OUT	IR-RX	I2S2-MCLK	PWM5			PF-EINT6

Table 8-25 PG Multiplex Function

GPIO Port	Function 2	Function 3	Function 4	Function 5	Function 6	Function 7	Function 8	Function 14
PG0	SDC1-CLK	UART3-TX	RGMII-RXCTRL/ RMII-CRS-DV	PWM7				PG-EINT0
PG1	SDC1-CMD	UART3-RX	RGMII-RXD0/ RMII-RXD0	PWM6				PG-EINT1
PG2	SDC1-D0	UART3-RTS	RGMII-RXD1/ RMII-RXD1	UART4-TX				PG-EINT2
PG3	SDC1-D1	UART3-CTS	RGMII-TXCK/ RMII-TXCK	UART4-RX				PG-EINT3
PG4	SDC1-D2	UART5-TX	RGMII-TXD0/ RMII-TXD0	PWM5				PG-EINT4
PG5	SDC1-D3	UART5-RX	RGMII-TXD1/ RMII-TXD1	PWM4				PG-EINT5
PG6	UART1-TX	TWI2-SCK	RGMII-TXD2	PWM1				PG-EINT6
PG7	UART1-RX	TWI2-SDA	RGMII-TXD3					PG-EINT7
PG8	UART1-RTS	TWI1-SCK	RGMII-RXD2	UART3-TX				PG-EINT8
PG9	UART1-CTS	TWI1-SDA	RGMII-RXD3	UART3-RX				PG-EINT9
PG10	PWM3	TWI3-SCK	RGMII-RXCK	CLK-FANOUT0	IR-RX			PG-EINT10
PG11	I2S1-MCLK	TWI3-SDA	EPHY-25M	CLK-FANOUT1	TCON-TRIG			PG-EINT11
PG12	I2S1-LRCK	TWI0-SCK	RGMII-TXCTRL/ RMII-TXEN	CLK-FANOUT2	PWM0	UART1-TX		PG-EINT12
PG13	I2S1-BCLK	TWI0-SDA	RGMII-CLKIN/ RMII-RXER	PWM2	LEDC-DO	UART1-RX		PG-EINT13
PG14	I2S1-DIN0	TWI2-SCK	MDC	I2S1-DOUT1	SPI0-WP	UART1-RTS		PG-EINT14
PG15	I2S1-DOUT0	TWI2-SDA	MDIO	I2S1-DIN1	SPI0-HOLD	UART1-CTS		PG-EINT15
PG16	IR-RX	TCON-TRIG	PWM5	CLK-FANOUT2		LEDC-DO		PG-EINT16
PG17	UART2-TX	TWI3-SCK	PWM7	CLK-FANOUT0	IR-TX	UART0-TX		PG-EINT17

GPIO Port	Function 2	Function 3	Function 4	Function 5	Function 6	Function 7	Function 8	Function 14
PG18	UART2-RX	TWI3-SDA	PWM6	CLK-FANOUT1	OWA-OUT	UART0-RX		PG-EINT18

### 8.7.3.3 Port Function

The Port Controller supports 5 GPIOs, every GPIO can configure as Input, Output, Function Peripheral, IO disable or Interrupt function. The configuration instruction of every function is as follows.

**Table 8-26 Port Function**

	Function	Buffer Strength	Pull Up	Pull Down
Input	GPIO/Multiplexing Input	/	X	X
Output	GPIO/Multiplexing Output	Y	X	X
Disable	Pull Up	/	Y	N
	Pull Down	/	N	Y
Interrupt	Trigger	/	X	X

/: non-configure, configuration is invalid

Y: configure

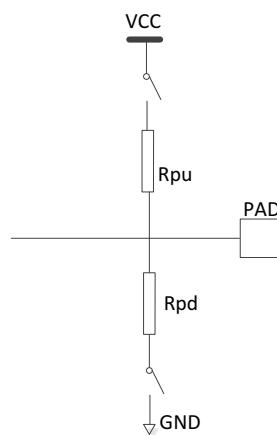
X: Select configuration according to the actual situation

N: Forbid to configure

### 8.7.3.4 Pull Up/Down and High-Impedance Logic

Each IO pin can configure the internal pull-up/down function or high-impedance.

**Figure 8-68 Pull up/down Logic**



**High-impedance**, the output is float state, all buffer is off, the level is decided by external high/low level. When high-impedance, the software configures the switch on Rpu and Rpd as off, and the multiplexing function of IO is set as IO disable or input by software.

**Pull-up**, an uncertain signal is pulled high by resistance, the resistance has a current-limiting function. When pulling up, the switch on Rpu is conducted by software configuration, the IO is pulled up to VCC by Rpu.

**Pull-down**, an uncertain signal is pulled low by a resistance. When pulling down, the switch on Rpd is conducted by software configuration, the IO is pulled down to GND by Rpd.

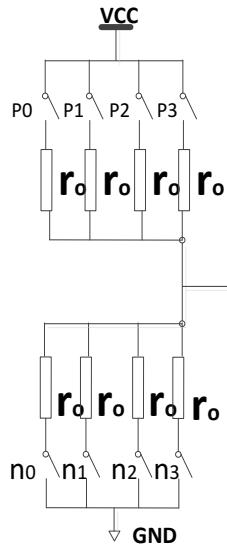
The pull-up/down of each IO is weak pull-up/down.

The setting of pull-down, pull-up, high-impedance is decided by the external circuit.

#### 8.7.3.5 Buffer Strength

Each IO can be set as different buffer strength. The IO buffer diagram is as follows.

Figure 8-69 IO Buffer Strength Diagram



When output high level, the n0, n1, n2, n3 of NMOS is off, the p0, p1, p2, p3 of PMOS is on. When the buffer strength is set to 0 (buffer strength is weakest), only the p0 is on, the output impedance is maximum, the impedance value is  $r_o$ . When the buffer strength is set to 1, only the p0 and p1 is on, the output impedance is equivalent to two  $r_o$  in parallel, the impedance value is  $r_o/2$ . When the buffer strength is 2, only the p0, p1, and p2 is on, the output impedance is equivalent to three  $r_o$  in parallel, the impedance value is  $r_o/3$ . When buffer strength is 3, the p0, p1, p2, and p3 is on, the output impedance is equivalent to four  $r_o$  in parallel, the impedance value is  $r_o/4$ .

When output low level, the p0, p1, p2, p3 of PMOS is off, the n0, n1, n2, n3 of NMOS is on. When the buffer strength is set to 0 (buffer strength is weakest), only the n0 is on, the output impedance is maximum, the

impedance value is  $r_0$ . When the buffer strength is set to 1, only the n0 and n1 is on, the output impedance is equivalent to two  $r_0$  in parallel, the impedance value is  $r_0/2$ . When the buffer strength is 2, only the n0, n1, and n2 is on, the output impedance is equivalent to three  $r_0$  in parallel, the impedance value is  $r_0/3$ . When the buffer strength is 3, the n0, n1, n2, and n3 is on, the output impedance is equivalent to four  $r_0$  in parallel, the impedance value is  $r_0/4$ .

When GPIO is set to input or interrupt function, between the output driver circuit and the port is unconnected, the driver configuration is invalid.



The typical value of  $r_0$  is  $180\Omega$ .

#### 8.7.3.6 Interrupt

Each group IO has an independent interrupt number. The IO within-group uses one interrupt number when one IO generates interrupt, the GPIO pins sent interrupt request to GIC. External Interrupt Status Register is used to query which IO generates interrupt.

The interrupt trigger of GPIO supports the following trigger types.

- Positive Edge: When a low level changes to a high level, the interrupt will generate. No matter how long a high level keeps, the interrupt generates only once.
- Negative Edge: When a high level changes to a low level, the interrupt will generate. No matter how long a low level keeps, the interrupt generates only once.
- High Level: Just keep a high level and the interrupt will always generate.
- Low Level: Just keep a low level and the interrupt will always generate.
- Double Edge: Positive and negative edge.

External Interrupt Configure Register is used to configure the trigger type.

The GPIO interrupt supports hardware debounce function by setting External Interrupt Debounce Register. Sample trigger signal using a lower sample clock, to reach the debounce effect because the dither frequency of the signal is higher than the sample frequency.

Set the sample clock source by PIO\_INT\_CLK\_SELECT and the prescale factor by DEB\_CLK\_PRE\_SCALE.

#### 8.7.4 Register List

Module Name	Base Address
GPIO	0x02000000

Register Name	Offset	Description
PB_CFG0	0x0030	PB Configure Register 0
PB_CFG1	0x0034	PB Configure Register 1
PB_DAT	0x0040	PB Data Register
PB_DRV0	0x0044	PB Multi_Driving Register 0
PB_DRV1	0x0048	PB Multi_Driving Register 1
PB_PULL0	0x0054	PB Pull Register 0
PC_CFG0	0x0060	PC Configure Register 0
PC_DAT	0x0070	PC Data Register
PC_DRV0	0x0074	PC Multi_Driving Register 0
PC_PULL0	0x0084	PC Pull Register 0
PD_CFG0	0x0090	PD Configure Register 0
PD_CFG1	0x0094	PD Configure Register 1
PD_CFG2	0x0098	PD Configure Register 2
PD_DAT	0x00A0	PD Data Register
PD_DRV0	0x00A4	PD Multi_Driving Register 0
PD_DRV1	0x00A8	PD Multi_Driving Register 1
PD_DRV2	0x00AC	PD Multi_Driving Register 2
PD_PULL0	0x00B4	PD Pull Register 0
PD_PULL1	0x00B8	PD Pull Register 1
PF_CFG0	0x00F0	PF Configure Register 0
PF_DAT	0x0100	PF Data Register
PF_DRV0	0x0104	PF Multi_Driving Register 0
PF_PULL0	0x0114	PF Pull Register 0
PG_CFG0	0x0120	PG Configure Register 0
PG_CFG1	0x0124	PG Configure Register 1
PG_DAT	0x0130	PG Data Register
PG_DRV0	0x0134	PG Multi_Driving Register 0
PG_DRV1	0x0138	PG Multi_Driving Register 1
PG_DRV3	0x0140	PG Multi_Driving Register 3
PG_PULL0	0x0144	PG Pull Register 0
PB_EINT_CFG0	0x0220	PB External Interrupt Configure Register 0
PB_EINT_CTL	0x0230	PB External Interrupt Control Register

Register Name	Offset	Description
PB_EINT_STATUS	0x0234	PB External Interrupt Status Register
PB_EINT_DEB	0x0238	PB External Interrupt Debounce Register
PC_EINT_CFG0	0x0240	PC External Interrupt Configure Register 0
PC_EINT_CTL	0x0250	PC External Interrupt Control Register
PC_EINT_STATUS	0x0254	PC External Interrupt Status Register
PC_EINT_DEB	0x0258	PC External Interrupt Debounce Register
PD_EINT_CFG0	0x0260	PD External Interrupt Configure Register 0
PD_EINT_CFG1	0x0264	PD External Interrupt Configure Register 1
PD_EINT_CFG2	0x0268	PD External Interrupt Configure Register 2
PD_EINT_CTL	0x0270	PD External Interrupt Control Register
PD_EINT_STATUS	0x0274	PD External Interrupt Status Register
PD_EINT_DEB	0x0278	PD External Interrupt Debounce Register
PF_EINT_CFG0	0x02A0	PF External Interrupt Configure Register 0
PF_EINT_CTL	0x02B0	PF External Interrupt Control Register
PF_EINT_STATUS	0x02B4	PF External Interrupt Status Register
PF_EINT_DEB	0x02B8	PF External Interrupt Debounce Register
PG_EINT_CFG0	0x02C0	PG External Interrupt Configure Register 0
PG_EINT_CFG1	0x02C4	PG External Interrupt Configure Register 1
PG_EINT_CTL	0x02D0	PG External Interrupt Control Register
PG_EINT_STATUS	0x02D4	PG External Interrupt Status Register
PG_EINT_DEB	0x02D8	PG External Interrupt Debounce Register
PIO_POW_MOD_SEL	0x0340	PIO Group Withstand Voltage Mode Select Register
PIO_POW_MS_CTL	0x0344	PIO Group Withstand Voltage Mode Select Control Register
PIO_POW_VAL	0x0348	PIO Group Power Value Register
PIO_POW_VOL_SEL_CTL	0x0350	PIO Group Power Voltage Select Control Register
PB_SECURE	0x0404	PB SECURE Register
PC_SECURE	0x0408	PC SECURE Register
PD_SECURE	0x040C	PD SECURE Register
PF_SECURE	0x0414	PF SECURE Register
PG_SECURE	0x0418	PG SECURE Register

### 8.7.5 Register Description

#### 8.7.5.1 0x0030 PB Configure Register 0 (Default Value: 0xFFFF\_FFFF)

Offset: 0x0030			Register Name: PB_CFG0
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0xF	PB7_SELECT PB7 Select 0000:Input 0001:Output 0010:Reserved 0011:I2S2-MCLK 0100:TWI3-SDA 0101:IR-RX 0110:Reserved 0111:UART3-RX 1000:CPUBIST1 1001:Reserved 1110:PB-EINT7 1111:IO Disable
27:24	R/W	0xF	PB6_SELECT PB6 Select 0000:Input 0001:Output 0010:Reserved 0011:I2S2-LRCK 0100:TWI3-SCK 0101:PWM1 0110:Reserved 0111:UART3-TX 1000:CPUBIST0 1001:Reserved 1110:PB-EINT6 1111:IO Disable
23:20	R/W	0xF	Reserved
19:16	R/W	0xF	Reserved
15:12	R/W	0xF	Reserved
11:8	R/W	0xF	Reserved
7:4	R/W	0xF	PB1_SELECT PB1 Select 0000:Input 0001:Output 0010:PWM4 0011:I2S2-DOUT3 0100:TWI2-SDA 0101:I2S2-DIN3 0110:UART0-RX 0111:UART2-RX 1000:IR-RX 1001:Reserved 1110:PB-EINT1 1111:IO Disable

Offset: 0x0030			Register Name: PB_CFG0
Bit	Read/Write	Default/Hex	Description
3:0	R/W	0xF	PB0_SELECT PB0 Select 0000:Input                  0001:Output 0010:PWM3                  0011:IR-TX 0100:TWI2-SCK              0101:SPI1-WP/DBI-TE 0110:UART0-TX              0111:UART2-TX 1000:OWA-OUT              1001:Reserved 1110:PB-EINT0              1111:IO Disable

#### 8.7.5.2 0x0034 PB Configure Register 1 (Default Value: 0x000F\_FFFF)

Offset: 0x0034			Register Name: PB_CFG1
Bit	Read/Write	Default/Hex	Description
31:20	/	/	/
19:16	R/W	0xF	PB12_SELECT PB12 Select 0000:Input                  0001:Output 0010:DMIC-CLK              0011:PWM0 0100: Reserved              0101:SPI1-CS/DBI-CSX 0110:CLK-FANOUT2           0111:IR-RX 1000:Reserved              1001:Reserved 1110:PB-EINT12              1111:IO Disable
15:12	R/W	0xF	PB11_SELECT PB11 Select 0000:Input                  0001:Output 0010:DMIC-DATA0            0011:PWM2 0100:TWI0-SDA              0101:SPI1-CLK/DBI-SCLK 0110:CLK-FANOUT1           0111:UART1-CTS 1000:Reserved              1001:Reserved 1110:PB-EINT11              1111:IO Disable

Offset: 0x0034			Register Name: PB_CFG1
Bit	Read/Write	Default/Hex	Description
11:8	R/W	0xF	PB10_SELECT PB10 Select 0000:Input 0001:Output 0010:DMIC-DATA1 0011:PWM7 0100:TWI0-SCK 0101:SPI1-MOSI/DBI-SDO 0110:CLK-FANOUT0 0111:UART1-RTS 1000:Reserved 1001:Reserved 1110:PB-EINT10 1111:IO Disable
7:4	R/W	0xF	PB9_SELECT PB9 Select 0000:Input 0001:Output 0010:DMIC-DATA2 0011:PWM6 0100:TWI2-SDA 0101:SPI1-MISO/DBI-SDI/DBI-TE/DBI-DCX 0110:UART0-RX 0111:UART1-RX 1000:Reserved 1001:Reserved 1110:PB-EINT9 1111:IO Disable
3:0	R/W	0xF	PB8_SELECT PB8 Select 0000:Input 0001:Output 0010:DMIC-DATA3 0011:PWM5 0100:TWI2-SCK 0101:SPI1-HOLD/DBI-DCX/DBI-WRX 0110:UART0-TX 0111:UART1-TX 1000:Reserved 1001:Reserved 1110:PB-EINT8 1111:IO Disable

#### 8.7.5.3 0x0040 PB Data Register (Default Value: 0x0000\_0000)

Offset: 0x0040			Register Name: PB_DAT
Bit	Read/Write	Default/Hex	Description
31:13	/	/	/
12:0	R/W	0x0	PB_DAT If the port is configured as the input function, the corresponding bit is the pin state. If the port is configured as the output function, the pin state is the same as the corresponding bit. The read bit value is the value set up by software. If the port is configured as a functional pin, the undefined value will be read.

#### 8.7.5.4 0x0044 PB Multi\_Driving Register 0 (Default Value: 0x1111\_1111)

Offset: 0x0044			Register Name: PB_DRV0
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29:28	R/W	0x1	Reserved
27:26	/	/	/
25:24	R/W	0x1	Reserved
23:22	/	/	/
21:20	R/W	0x1	Reserved
19:18	/	/	/
17:16	R/W	0x1	Reserved
15:14	/	/	/
13:12	R/W	0x1	Reserved
11:10	/	/	/
9:8	R/W	0x1	Reserved
7:6	/	/	/
5:4	R/W	0x1	PB1_DRV PB1 Multi_Driving Select 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3
3:2	/	/	/
1:0	R/W	0x1	PB0_DRV PB0 Multi_Driving Select 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3

#### 8.7.5.5 0x0048 PB Multi\_Driving Register 1 (Default Value: 0x0001\_1111)

Offset: 0x0048			Register Name: PB_DRV1
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/

Offset: 0x0048			Register Name: PB_DRV1
Bit	Read/Write	Default/Hex	Description
17:16	R/W	0x1	PB12_DRV PB12 Multi_Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
15:14	/	/	/
13:12	R/W	0x1	PB11_DRV PB11 Multi_Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
11:10	/	/	/
9:8	R/W	0x1	PB10_DRV PB10 Multi_Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
7:6	/	/	/
5:4	R/W	0x1	PB9_DRV PB9 Multi_Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
3:2	/	/	/
1:0	R/W	0x1	PB8_DRV PB8 Multi_Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3

#### 8.7.5.6 0x0054 PB Pull Register 0 (Default Value: 0x0000\_0000)

Offset: 0x0054			Register Name: PB_PULL0
Bit	Read/Write	Default/Hex	Description
31:26	/	/	/
25:24	R/W	0x0	PB12_PULL PB12 Pull_up or down Select 00: Pull_up/down disable 01: Pull_up 10: Pull_down 11: Reserved

Offset: 0x0054			Register Name: PB_PULL0
Bit	Read/Write	Default/Hex	Description
23:22	R/W	0x0	PB11_PULL PB11 Pull_up or down Select 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
21:20	R/W	0x0	PB10_PULL PB10 Pull_up or down Select 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
19:18	R/W	0x0	PB9_PULL PB9 Pull_up or down Select 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
17:16	R/W	0x0	PB8_PULL PB8 Pull_up or down Select 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
15:14	R/W	0x0	Reserved
13:12	R/W	0x0	Reserved
11:10	R/W	0x0	Reserved
9:8	R/W	0x0	Reserved
7:6	R/W	0x0	Reserved
5:4	R/W	0x0	Reserved
3:2	R/W	0x0	PB1_PULL PB1 Pull_up or down Select 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
1:0	R/W	0x0	PB0_PULL PB0 Pull_up or down Select 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved

**8.7.5.7 0x0060 PC Configure Register 0 (Default Value: 0xFFFF\_FFFF)**

Offset: 0x0060			Register Name: PC_CFG0
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0xF	PC7_SELECT PC7 Select 0000:Input 0001:Output 0010:SPI0-HOLD 0011:SDC2-D3 0100:UART3-RX 0101:TWI3-SDA 0110:TCON-TRIG 0111:Reserved 1000:Reserved 1001:Reserved 1110:PC-EINT7 1111:IO Disable
27:24	R/W	0xF	PC6_SELECT PC6 Select 0000:Input 0001:Output 0010:SPI0-WP 0011:SDC2-D0 0100:UART3-TX 0101:TWI3-SCK 0110:DBG-CLK 0111:Reserved 1000:Reserved 1001:Reserved 1110:PC-EINT6 1111:IO Disable
23:20	R/W	0xF	PC5_SELECT PC5 Select 0000:Input 0001:Output 0010:SPI0-MISO 0011:SDC2-D1 0100:BOOT-SEL1 0101:Reserved 0110:Reserved 0111:Reserved 1000:Reserved 1001:Reserved 1110:PC-EINT5 1111:IO Disable
19:16	R/W	0xF	PC4_SELECT PC4 Select. 0000:Input 0001:Output 0010:SPI0-MOSI 0011:SDC2-D2 0100:BOOT-SEL0 0101:Reserved 0110:Reserved 0111:Reserved 1000:Reserved 1001:Reserved 1110:PC-EINT4 1111:IO Disable

Offset: 0x0060			Register Name: PC_CFG0
Bit	Read/Write	Default/Hex	Description
15:12	R/W	0xF	PC3_SELECT PC3 Select 0000:Input 0001:Output 0010:SPIO-CS0 0011:SDC2-CMD 0100:Reserved 0101:Reserved 0110:Reserved 0111:Reserved 1000:Reserved 1001:Reserved 1110:PC-EINT3 1111:IO Disable
7:4	R/W	0xF	Reserved
3:0	R/W	0xF	Reserved

#### 8.7.5.8 0x0070 PC Data Register (Default Value: 0x0000\_0000)

Offset: 0x0070			Register Name: PC_DAT
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R/W	0x0	PC_DAT If the port is configured as the input function, the corresponding bit is the pin state. If the port is configured as the output function, the pin state is the same as the corresponding bit. The read bit value is the value set up by software. If the port is configured as a functional pin, the undefined value will be read.

### 8.7.5.9 0x0074 PC Multi\_Driving Register 0 (Default Value: 0x1111\_1111)

Offset: 0x0074			Register Name: PC_DRV0
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29:28	R/W	0x1	PC7_DRV PC7 Multi_Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
27:26	/	/	/
25:24	R/W	0x1	PC6_DRV PC6 Multi_Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
23:22	/	/	/
21:20	R/W	0x1	PC5_DRV PC5 Multi_Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
19:18	/	/	/
17:16	R/W	0x1	PC4_DRV PC4 Multi_Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
15:14	/	/	/
13:12	R/W	0x1	PC3_DRV PC3 Multi_Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
11:10	/	/	/
9:8	R/W	0x1	PC2_DRV PC2 Multi_Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
7:6	/	/	/
5:4	R/W	0x1	Reserved
3:2	/	/	/
1:0	R/W	0x1	Reserved

**8.7.5.10 0x0084 PC Pull Register 0 (Default Value: 0x0000\_0540)**

Offset: 0x0084			Register Name: PC_PULL0
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:14	R/W	0x0	PC7_PULL PC7 Pull_up/down Select 00: Pull_up/down disable      01: Pull_up 10: Pull_down      11: Reserved
13:12	R/W	0x0	PC6_PULL PC6 Pull_up/down Select 00: Pull_up/down disable      01: Pull_up 10: Pull_down      11: Reserved
11:10	R/W	0x1	PC5_PULL PC5 Pull_up/down Select 00: Pull_up/down disable      01: Pull_up 10: Pull_down      11: Reserved
9:8	R/W	0x1	PC4_PULL PC4 Pull_up/down Select 00: Pull_up/down disable      01: Pull_up 10: Pull_down      11: Reserved
7:6	R/W	0x1	PC3_PULL PC3 Pull_up/down Select 00: Pull_up/down disable      01: Pull_up 10: Pull_down      11: Reserved
5:4	R/W	0x0	PC2_PULL PC2 Pull_up/down Select 00: Pull_up/down disable      01: Pull_up 10: Pull_down      11: Reserved
3:2	R/W	0x0	Reserved
1:0	R/W	0x01	Reserved

**8.7.5.11 0x0090 PD Configure Register 0 (Default Value: 0xFFFF\_FFFF)**

Offset: 0x0090			Register Name: PD_CFG0
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0xF	PD7_SELECT PD7 Select 0000:Input 0001:Output 0010:LCD0-D11 0011:LVDS0-CKN 0100:DSI-D2N 0101:UART4-TX 0110:Reserved 0111:Reserved 1000:Reserved 1001:Reserved 1110:PD-EINT7 1111:IO Disable
27:24	R/W	0xF	PD6_SELECT PD6 Select 0000:Input 0001:Output 0010:LCD0-D10 0011:LVDS0-CKP 0100:DSI-D2P 0101:UART5-RX 0110:Reserved 0111:Reserved 1000:Reserved 1001:Reserved 1110:PD-EINT6 1111:IO Disable
23:20	R/W	0xF	PD5_SELECT PD5 Select 0000:Input 0001:Output 0010:LCD0-D7 0011:LVDS0-V2N 0100:DSI-CKN 0101:UART5-TX 0110:Reserved 0111:Reserved 1000:Reserved 1001:Reserved 1110:PD-EINT5 1111:IO Disable
19:16	R/W	0xF	PD4_SELECT PD4 Select 0000:Input 0001:Output 0010:LCD0-D6 0011:LVDS0-V2P 0100:DSI-CKP 0101:UART2-CTS 0110:Reserved 0111:Reserved 1000:Reserved 1001:Reserved 1110:PD-EINT4 1111:IO Disable

Offset: 0x0090			Register Name: PD_CFG0
Bit	Read/Write	Default/Hex	Description
15:12	R/W	0xF	PD3_SELECT PD3 Select 0000:Input 0001:Output 0010:LCD0-D5 0011:LVDS0-V1N 0100:DSI-D1N 0101:UART2-RTS 0110:Reserved 0111:Reserved 1000:Reserved 1001:Reserved 1110:PD-EINT3 1111:IO Disable
			PD2_SELECT PD2 Select 0000:Input 0001:Output 0010:LCD0-D4 0011:LVDS0-V1P 0100:DSI-D1P 0101:UART2-RX 0110:Reserved 0111:Reserved 1000:Reserved 1001:Reserved 1110:PD-EINT2 1111:IO Disable
			PD1_SELECT PD1 Select 0000:Input 0001:Output 0010:LCD0-D3 0011:LVDS0-V0N 0100:DSI-D0N 0101:UART2-TX 0110:Reserved 0111:Reserved 1000:Reserved 1001:Reserved 1110:PD-EINT1 1111:IO Disable
			PDO_SELECT PDO Select 0000:Input 0001:Output 0010:LCD0-D2 0011:LVDS0-V0P 0100:DSI-D0P 0101:TWI0-SCK 0110:Reserved 0111:Reserved 1000:Reserved 1001:Reserved 1110:PD-EINT0 1111:IO Disable

#### 8.7.5.12 0x0094 PD Configure Register 1 (Default Value: 0xFFFF\_FFFF)

Offset: 0x0094			Register Name: PD_CFG1
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0xF	Reserved
27:24	R/W	0xF	Reserved
23:20	R/W	0xF	Reserved
19:16	R/W	0xF	Reserved
15:12	R/W	0xF	Reserved
11:8	R/W	0xF	Reserved
7:4	R/W	0xF	PD9_SELECT PD9 Select 0000:Input                  0001:Output 0010:LCD0-D13              0011:LVDS0-V3N 0100:DSI-D3N                0101:PWM6 0110:Reserved                0111:Reserved 1000:Reserved                1001:Reserved 1110:PD-EINT9              1111:IO Disable
3:0	R/W	0xF	PD8_SELECT PD8 Select 0000:Input                  0001:Output 0010:LCD0-D12              0011:LVDS0-V3P 0100:DSI-D3P                0101:UART4-RX 0110:Reserved                0111:Reserved 1000:Reserved                1001:Reserved 1110:PD-EINT8              1111:IO Disable

#### 8.7.5.13 0x00A0 PD Data Register (Default Value: 0x0000\_0000)

Offset: 0x00A0			Register Name: PD_DAT
Bit	Read/Write	Default/Hex	Description
31:23	/	/	/

Offset: 0x00A0			Register Name: PD_DAT
Bit	Read/Write	Default/Hex	Description
22:0	R/W	0x0	<p>PD_DAT</p> <p>PD Data</p> <p>If the port is configured as the input function, the corresponding bit is the pin state. If the port is configured as the output function, the pin state is the same as the corresponding bit. The read bit value is the value set up by software. If the port is configured as a functional pin, the undefined value will be read.</p>

#### 8.7.5.14 0x00A4 PD Multi\_Driving Register 0 (Default Value: 0x1111\_1111)

Offset: 0x00A4			Register Name: PD_DRV0				
Bit	Read/Write	Default/Hex	Description				
31:30	/	/	/				
29:28	R/W	0x1	<p>PD7_DRV</p> <p>PD7 Multi_Driving Select.</p> <table> <tr> <td>00: Level 0</td> <td>01: Level 1</td> </tr> <tr> <td>10: Level 2</td> <td>11: Level 3</td> </tr> </table>	00: Level 0	01: Level 1	10: Level 2	11: Level 3
00: Level 0	01: Level 1						
10: Level 2	11: Level 3						
27:26	/	/	/				
25:24	R/W	0x1	<p>PD6_DRV</p> <p>PD6 Multi_Driving Select.</p> <table> <tr> <td>00: Level 0</td> <td>01: Level 1</td> </tr> <tr> <td>10: Level 2</td> <td>11: Level 3</td> </tr> </table>	00: Level 0	01: Level 1	10: Level 2	11: Level 3
00: Level 0	01: Level 1						
10: Level 2	11: Level 3						
23:22	/	/	/				
21:20	R/W	0x1	<p>PD5_DRV</p> <p>PD5 Multi_Driving Select.</p> <table> <tr> <td>00: Level 0</td> <td>01: Level 1</td> </tr> <tr> <td>10: Level 2</td> <td>11: Level 3</td> </tr> </table>	00: Level 0	01: Level 1	10: Level 2	11: Level 3
00: Level 0	01: Level 1						
10: Level 2	11: Level 3						
19:18	/	/	/				
17:16	R/W	0x1	<p>PD4_DRV</p> <p>PD4 Multi_Driving Select.</p> <table> <tr> <td>00: Level 0</td> <td>01: Level 1</td> </tr> <tr> <td>10: Level 2</td> <td>11: Level 3</td> </tr> </table>	00: Level 0	01: Level 1	10: Level 2	11: Level 3
00: Level 0	01: Level 1						
10: Level 2	11: Level 3						
15:14	/	/	/				

Offset: 0x00A4			Register Name: PD_DRV0
Bit	Read/Write	Default/Hex	Description
13:12	R/W	0x1	PD3_DRV PD3 Multi_Driving Select. 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3
11:10	/	/	/
9:8	R/W	0x1	PD2_DRV PD2 Multi_Driving Select. 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3
7:6	/	/	/
5:4	R/W	0x1	PD1_DRV PD1 Multi_Driving Select. 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3
3:2	/	/	/
1:0	R/W	0x1	PD0_DRV PD0 Multi_Driving Select. 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3

#### 8.7.5.15 0x00A8 PD Multi\_Driving Register 1 (Default Value: 0x1111\_1111)

Offset: 0x00A8			Register Name: PD_DRV1
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29:28	R/W	0x1	Reserved
27:26	/	/	/
25:24	R/W	0x1	Reserved
23:22	/	/	/
21:20	R/W	0x1	Reserved
19:18	/	/	/
17:16	R/W	0x1	Reserved
15:14	/	/	/
13:12	R/W	0x1	Reserved
11:10	/	/	/

Offset: 0x00A8			Register Name: PD_DRV1
Bit	Read/Write	Default/Hex	Description
9:8	R/W	0x1	Reserved
31:6	/	/	/
5:4	R/W	0x1	PD9_DRV PD9 Multi_Driving Select. 00: Level 0 01: Level 1 10: Level 2 11: Level 3
3:2	/	/	/
1:0	R/W	0x1	PD8_DRV PD8 Multi_Driving Select. 00: Level 0 01: Level 1 10: Level 2 11: Level 3

#### 8.7.5.16 0x00B4 PD Pull Register 0 (Default Value: 0x0000\_0000)

Offset: 0x00B4			Register Name: PD_PULL0
Bit	Read/Write	Default/Hex	Description
31:30	R/W	0x0	Reserved
29:28	R/W	0x0	Reserved
27:26	R/W	0x0	Reserved
25:24	R/W	0x0	Reserved
23:22	R/W	0x0	Reserved
21:20	R/W	0x0	Reserved
19:18	R/W	0x0	PD9_PULL PD9 Pull_up or down Select. 00: Pull_up/down disable 01: Pull_up 10: Pull_down 11: Reserved
17:16	R/W	0x0	PD8_PULL PD8 Pull_up or down Select. 00: Pull_up/down disable 01: Pull_up 10: Pull_down 11: Reserved

Offset: 0x00B4			Register Name: PD_PULL0
Bit	Read/Write	Default/Hex	Description
15:14	R/W	0x0	PD7_PULL PD7 Pull_up or down Select. 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
13:12	R/W	0x0	PD6_PULL PD6 Pull_up or down Select. 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
11:10	R/W	0x0	PD5_PULL PD5 Pull_up or down Select. 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
9:8	R/W	0x0	PD4_PULL PD4 Pull_up or down Select. 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
7:6	R/W	0x0	PD3_PULL PD3 Pull_up or down Select. 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
5:4	R/W	0x0	PD2_PULL PD2 Pull_up or down Select. 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
3:2	R/W	0x0	PD1_PULL PD1 Pull_up or down Select. 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
1:0	R/W	0x0	PD0_PULL PD0 Pull_up or down Select. 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved

**8.7.5.17 0x00F0 PF Configure Register 0 (Default Value: 0xFFFF\_FFFF)**

Offset: 0x00F0			Register Name: PF_CFG0
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27:24	R/W	0xF	PF6_SELECT PF6 Select 0000:Input 0001:Output 0010:Reserved 0011:OWA-OUT 0100:IR-RX 0101:I2S2-MCLK 0110:PWM5 0111:Reserved 1000:Reserved 1001:Reserved 1110:PF-EINT6 1111:IO Disable
23:20	R/W	0xF	PF5_SELECT PF5 Select 0000:Input 0001:Output 0010:SDC0-D2 0011:JTAG-CK 0100:Reserved 0101:I2S2-LRCK 0110:Reserved 0111:Reserved 1000:Reserved 1001:Reserved 1110:PF-EINT5 1111:IO Disable
19:16	R/W	0xF	PF4_SELECT PF4 Select 0000:Input 0001:Output 0010:SDC0-D3 0011:UART0-RX 0100:TWI0-SDA 0101:PWM6 0110:IR-TX 0111:Reserved 1000:Reserved 1001:Reserved 1110:PF-EINT4 1111:IO Disable
15:12	R/W	0xF	PF3_SELECT PF3 Select 0000:Input 0001:Output 0010:SDC0-CMD 0011:JTAG-DO 0100:Reserved 0101:I2S2-BCLK 0110:Reserved 0111:Reserved 1000:Reserved 1001:Reserved 1110:PF-EINT3 1111:IO Disable

Offset: 0x00F0			Register Name: PF_CFG0
Bit	Read/Write	Default/Hex	Description
11:8	R/W	0xF	PF2_SELECT PF2 Select 0000:Input 0001: Output 0010:SDC0-CLK 0011: UART0-TX 0100:TWI0-SCK 0101: Reserved 0110: Reserved 0111: Reserved 1000:Reserved 1001: Reserved 1110:PF-EINT2 1111: IO Disable
7:4	R/W	0xF	PF1_SELECT PF1 Select 0000:Input 0001:Output 0010:SDC0-D0 0011:JTAG-DI 0100:Reserved 0101:I2S2-DOUT0 0110:I2S2-DIN1 0111:Reserved 1000:Reserved 1001:Reserved 1110:PF-EINT1 1111:IO Disable
3:0	R/W	0xF	PFO_SELECT PFO Select 0000:Input 0001:Output 0010:SDC0-D1 0011:JTAG-MS 0100:Reserved 0101:I2S2-DOUT1 0110:I2S2-DIN0 0111:Reserved 1000:Reserved 1001:Reserved 1110:PF-EINT0 1111:IO Disable

#### 8.7.5.18 0x0100 PF Data Register (Default Value: 0x0000\_0000)

Offset: 0x0100			Register Name: PF_DAT
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/

Offset: 0x0100			Register Name: PF_DAT
Bit	Read/Write	Default/Hex	Description
6:0	R/W	0	<p>PF_DAT</p> <p>PF Data</p> <p>If the port is configured as input, the corresponding bit is the pin state. If the port is configured as output, the pin state is the same as the corresponding bit. The read bit value is the value setup by software. If the port is configured as functional pin, the undefined value will be read.</p>

#### 8.7.5.19 0x0104 PF Multi\_Driving Register 0 (Default Value: 0x0111\_1111)

Offset: 0x0104			Register Name: PF_DRV0				
Bit	Read/Write	Default/Hex	Description				
31:26	/	/	/				
25:24	R/W	0x1	<p>PF6_DRV</p> <p>PF6 Multi_Driving Select</p> <table> <tr> <td>00: Level 0</td> <td>01: Level 1</td> </tr> <tr> <td>10: Level 2</td> <td>11: Level 3</td> </tr> </table>	00: Level 0	01: Level 1	10: Level 2	11: Level 3
00: Level 0	01: Level 1						
10: Level 2	11: Level 3						
23:22	/	/	/				
21:20	R/W	0x1	<p>PF5_DRV</p> <p>PF5 Multi_Driving Select</p> <table> <tr> <td>00: Level 0</td> <td>01: Level 1</td> </tr> <tr> <td>10: Level 2</td> <td>11: Level 3</td> </tr> </table>	00: Level 0	01: Level 1	10: Level 2	11: Level 3
00: Level 0	01: Level 1						
10: Level 2	11: Level 3						
19:18	/	/	/				
17:16	R/W	0x1	<p>PF4_DRV</p> <p>PF4 Multi_Driving Select</p> <table> <tr> <td>00: Level 0</td> <td>01: Level 1</td> </tr> <tr> <td>10: Level 2</td> <td>11: Level 3</td> </tr> </table>	00: Level 0	01: Level 1	10: Level 2	11: Level 3
00: Level 0	01: Level 1						
10: Level 2	11: Level 3						
15:14	/	/	/				
13:12	R/W	0x1	<p>PF3_DRV</p> <p>PF3 Multi_Driving Select</p> <table> <tr> <td>00: Level 0</td> <td>01: Level 1</td> </tr> <tr> <td>10: Level 2</td> <td>11: Level 3</td> </tr> </table>	00: Level 0	01: Level 1	10: Level 2	11: Level 3
00: Level 0	01: Level 1						
10: Level 2	11: Level 3						
11:10	/	/	/				

Offset: 0x0104			Register Name: PF_DRV0
Bit	Read/Write	Default/Hex	Description
9:8	R/W	0x1	PF2_DRV PF2 Multi_Driving Select 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3
7:6	/	/	/
5:4	R/W	0x1	PF1_DRV PF1 Multi_Driving Select 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3
3:2	/	/	/
1:0	R/W	0x1	PFO_DRV PFO Multi_Driving Select 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3

#### 8.7.5.20 0x0114 PF Pull Register 0 (Default Value: 0x0000\_0000)

Offset: 0x0114			Register Name: PF_PULL0
Bit	Read/Write	Default/Hex	Description
31:14	/	/	/
13:12	R/W	0x0	PF6_PULL PF6 Pull_up or down Select 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
11:10	R/W	0x0	PF5_PULL PF5 Pull_up or down Select 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
9:8	R/W	0x0	PF4_PULL PF4 Pull_up or down Select 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved

Offset: 0x0114			Register Name: PF_PULL0
Bit	Read/Write	Default/Hex	Description
7:6	R/W	0x0	PF3_PULL PF3 Pull_up or down Select 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
5:4	R/W	0x0	PF2_PULL PF2 Pull_up or down Select 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
3:2	R/W	0x0	PF1_PULL PF1 Pull_up or down Select 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
1:0	R/W	0x0	PFO_PULL PFO Pull_up or down Select 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved

#### 8.7.5.21 0x0120 PG Configure Register 0 (Default Value: 0xFFFF\_FFFF)

Offset: 0x0120			Register Name: PG_CFG0
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0xF	PG7_SELECT PG7 Select 0000:Input                    0001:Output 0010:UART1-RX                0011:TWI2-SDA 0100:RGMII-TXD3              0101: Reserved 0110:Reserved                 0111:Reserved 1000:Reserved                 1001:Reserved 1110:PG-EINT7                1111:IO Disable

Offset: 0x0120			Register Name: PG_CFG0
Bit	Read/Write	Default/Hex	Description
27:24	R/W	0xF	PG6_SELECT PG6 Select 0000:Input 0001:Output 0010:UART1-TX 0011:TWI2-SCK 0100:RGMII-TXD2 0101:PWM1 0110:Reserved 0111:Reserved 1000:Reserved 1001:Reserved 1110:PG-EINT6 1111:IO Disable
23:20	R/W	0xF	PG5_SELECT PG5 Select 0000:Input 0001:Output 0010:SDC1-D3 0011:UART5-RX 0100:RGMII-TXD1/RMII-TXD1 0101:PWM4 0110:Reserved 0111:Reserved 1000:Reserved 1001:Reserved 1110:PG-EINT5 1111:IO Disable
19:16	R/W	0xF	PG4_SELECT PG4 Select 0000:Input 0001:Output 0010:SDC1-D2 0011:UART5-TX 0100:RGMII-TXD0/RMII-TXD0 0101:PWM5 0110:Reserved 0111:Reserved 1000:Reserved 1001:Reserved 1110:PG-EINT4 1111:IO Disable
15:12	R/W	0xF	PG3_SELECT PG3 Select 0000:Input 0001:Output 0010:SDC1-D1 0011:UART3-CTS 0100:RGMII-TXCK/RMII-TXCK 0101:UART4-RX 0110:Reserved 0111:Reserved 1000:Reserved 1001:Reserved 1110:PG-EINT3 1111:IO Disable

Offset: 0x0120			Register Name: PG_CFG0
Bit	Read/Write	Default/Hex	Description
11:8	R/W	0xF	PG2_SELECT
			PG2 Select
			0000:Input 0001:Output
			0010:SDC1-D0 0011:UART3-RTS
			0100:RGMII-RXD1/RMII-RXD1 0101:UART4-TX
			0110:Reserved 0111:Reserved
			1000:Reserved 1001:Reserved
			1110:PG-EINT2 1111:IO Disable
7:4	R/W	0xF	PG1_SELECT
			PG1 Select
			0000:Input 0001:Output
			0010:SDC1-CMD 0011:UART3-RX
			0100:RGMII-RXD0/RMII-RXD0 0101:PWM6
			0110:Reserved 0111:Reserved
			1000:Reserved 1001:Reserved
			1110:PG-EINT1 1111:IO Disable
3:0	R/W	0xF	PG0_SELECT
			PG0 Select
			0000:Input 0001:Output
			0010:SDC1-CLK 0011:UART3-TX
			0100:RGMII-RXCTRL/RMII-CRS-DV 0101:PWM7
			0110:Reserved 0111:Reserved
			1000:Reserved 1001:Reserved
			1110:PG-EINT0 1111:IO Disable

**8.7.5.22 0x0124 PG Configure Register 1 (Default Value: 0xFFFF\_FFFF)**

Offset: 0x0124			Register Name: PG_CFG1
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0xF	PG15_SELECT PG15 Select 0000:Input 0001:Output 0010:I2S1-DOUT0 0011:TWI2-SDA 0100:MDIO 0101:I2S1-DIN1 0110:SPI0-HOLD 0111:UART1-CTS 1000:Reserved 1001:Reserved 1110:PG-EINT15 1111:IO Disable
27:24	R/W	0xF	PG14_SELECT PG14 Select 0000:Input 0001:Output 0010:I2S1-DINO 0011:TWI2-SCK 0100:MDC 0101:I2S1-DOUT1 0110:SPI0-WP 0111:UART1-RTS 1000:Reserved 1001:Reserved 1110:PG-EINT14 1111:IO Disable
23:20	R/W	0xF	PG13_SELECT PG13 Select. 0000:Input 0001:Output 0010:I2S1-BCLK 0011:TWI0-SDA 0100:RGMII-CLKIN/RMII-RXER 0101:PWM2 0110:LEDC-DO 0111:UART1-RX 1000:Reserved 1001:Reserved 1110:PG-EINT13 1111:IO Disable
19:16	R/W	0xF	PG12_SELECT PG12 Select 0000:Input 0001:Output 0010:I2S1-LRCK 0011:TWI0-SCK 0100:RGMII-TXCTRL/RMII-TXEN 0101:CLK-FANOUT2 0110:PWM0 0111:UART1-TX 1000:Reserved 1001:Reserved 1110:PG-EINT12 1111:IO Disable

Offset: 0x0124			Register Name: PG_CFG1
Bit	Read/Write	Default/Hex	Description
15:12	R/W	0xF	PG11_SELECT PG11 Select 0000:Input 0001:Output 0010:I2S1-MCLK 0011:TWI3-SDA 0100:EPHY-25M 0101:CLK-FANOUT1 0110:TCON-TRIG 0111:Reserved 1000:Reserved 1001:Reserved 1110:PG-EINT11 1111:IO Disable
11:8	R/W	0xF	PG10_SELECT PG10 Select 0000:Input 0001:Output 0010:PWM3 0011:TWI3-SCK 0100:RGMII-RXCK 0101:CLK-FANOUT0 0110:IR-RX 0111:Reserved 1000:Reserved 1001:Reserved 1110:PG-EINT10 1111:IO Disable
7:4	R/W	0xF	PG9_SELECT PG9 Select. 0000:Input 0001:Output 0010:UART1-CTS 0011:TWI1-SDA 0100:RGMII-RXD3 0101:UART3-RX 0110:Reserved 0111:Reserved 1000:Reserved 1001:Reserved 1110:PG-EINT9 1111:IO Disable
3:0	R/W	0xF	PG8_SELECT PG8 Select 0000:Input 0001:Output 0010:UART1-RTS 0011:TWI1-SCK 0100:RGMII-RXD2 0101:UART3-TX 0110:Reserved 0111:Reserved 1000:Reserved 1001:Reserved 1110:PG-EINT8 1111:IO Disable

**8.7.5.23 0x0128 PG Configure Register 2 (Default Value: 0x0000\_0FFF)**

Offset: 0x0128			Register Name: PG_CFG2
Bit	Read/Write	Default/Hex	Description
31:12	/	/	/
11:8	R/W	0xF	PG18_SELECT PG18 Select 0000:Input 0001:Output 0010:UART2-RX 0011:TWI3-SDA 0100:PWM6 0101:CLK-FANOUT1 0110:OWA-OUT 0111:UART0-RX 1000:Reserved 1001:Reserved 1110:PG-EINT18 1111:IO Disable
7:4	R/W	0xF	PG17_SELECT PG17 Select 0000:Input 0001:Output 0010:UART2-TX 0011:TWI3-SCK 0100:PWM7 0101:CLK-FANOUT0 0110:IR-TX 0111:UART0-TX 1000:Reserved 1001:Reserved 1110:PG-EINT17 1111:IO Disable
3:0	R/W	0xF	PG16_SELECT PG16 Select 0000:Input 0001:Output 0010:IR-RX 0011:TCON-TRIG 0100:PWM5 0101:CLK-FANOUT2 0110: Reserved 0111:LEDC-DO 1000:Reserved 1001:Reserved 1110:PG-EINT16 1111:IO Disable

**8.7.5.24 0x0130 PG Data Register (Default Value: 0x0000\_0000)**

Offset: 0x0130			Register Name: PG_DAT
Bit	Read/Write	Default/Hex	Description
31:19	/	/	/

Offset: 0x0130			Register Name: PG_DAT
Bit	Read/Write	Default/Hex	Description
18:0	R/W	0x0	<p>PG_DAT</p> <p>If the port is configured as the input function, the corresponding bit is the pin state. If the port is configured as the output function, the pin state is the same as the corresponding bit. The read bit value is the value set up by software. If the port is configured as a functional pin, the undefined value will be read.</p>

#### 8.7.5.25 0x0134 PG Multi\_Driving Register 0 (Default Value: 0x1111\_1111)

Offset: 0x0134			Register Name: PG_DRV0				
Bit	Read/Write	Default/Hex	Description				
31:30	/	/	/				
29:28	R/W	0x1	<p>PG7_DRV</p> <p>PG7 Multi_Driving Select</p> <table> <tr> <td>00: Level 0</td> <td>01: Level 1</td> </tr> <tr> <td>10: Level 2</td> <td>11: Level 3</td> </tr> </table>	00: Level 0	01: Level 1	10: Level 2	11: Level 3
00: Level 0	01: Level 1						
10: Level 2	11: Level 3						
27:26	/	/	/				
25:24	R/W	0x1	<p>PG6_DRV</p> <p>PG6 Multi_Driving Select</p> <table> <tr> <td>00: Level 0</td> <td>01: Level 1</td> </tr> <tr> <td>10: Level 2</td> <td>11: Level 3</td> </tr> </table>	00: Level 0	01: Level 1	10: Level 2	11: Level 3
00: Level 0	01: Level 1						
10: Level 2	11: Level 3						
23:22	/	/	/				
21:20	R/W	0x1	<p>PG5_DRV</p> <p>PG5 Multi_Driving Select</p> <table> <tr> <td>00: Level 0</td> <td>01: Level 1</td> </tr> <tr> <td>10: Level 2</td> <td>11: Level 3</td> </tr> </table>	00: Level 0	01: Level 1	10: Level 2	11: Level 3
00: Level 0	01: Level 1						
10: Level 2	11: Level 3						
19:18	/	/	/				
17:16	R/W	0x1	<p>PG4_DRV</p> <p>PG4 Multi_Driving Select</p> <table> <tr> <td>00: Level 0</td> <td>01: Level 1</td> </tr> <tr> <td>10: Level 2</td> <td>11: Level 3</td> </tr> </table>	00: Level 0	01: Level 1	10: Level 2	11: Level 3
00: Level 0	01: Level 1						
10: Level 2	11: Level 3						
15:14	/	/	/				
13:12	R/W	0x1	<p>PG3_DRV</p> <p>PG3 Multi_Driving Select</p> <table> <tr> <td>00: Level 0</td> <td>01: Level 1</td> </tr> <tr> <td>10: Level 2</td> <td>11: Level 3</td> </tr> </table>	00: Level 0	01: Level 1	10: Level 2	11: Level 3
00: Level 0	01: Level 1						
10: Level 2	11: Level 3						

Offset: 0x0134			Register Name: PG_DRV0
Bit	Read/Write	Default/Hex	Description
11:10	/	/	/
9:8	R/W	0x1	PG2_DRV PG2 Multi_Driving Select 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3
7:6	/	/	/
5:4	R/W	0x1	PG1_DRV PG1 Multi_Driving Select 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3
3:2	/	/	/
1:0	R/W	0x1	PG0_DRV PG0 Multi_Driving Select 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3

#### 8.7.5.26 0x0138 PG Multi\_Driving Register 1 (Default Value: 0x1111\_1111)

Offset: 0x0138			Register Name: PG_DRV1
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29:28	R/W	0x1	PG15_DRV PG15 Multi_Driving Select 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3
27:26	/	/	/
25:24	R/W	0x1	PG14_DRV PG14 Multi_Driving Select 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3
23:22	/	/	/
21:20	R/W	0x1	PG13_DRV PG13 Multi_Driving Select 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3

Offset: 0x0138			Register Name: PG_DRV1
Bit	Read/Write	Default/Hex	Description
19:18	/	/	/
17:16	R/W	0x1	PG12_DRV PG12 Multi_Driving Select 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3
15:14	/	/	/
13:12	R/W	0x1	PG11_DRV PG11 Multi_Driving Select 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3
11:10	/	/	/
9:8	R/W	0x1	PG10_DRV PG10 Multi_Driving Select 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3
7:6	/	/	/
5:4	R/W	0x1	PG9_DRV PG9 Multi_Driving Select 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3
3:2	/	/	/
1:0	R/W	0x1	PG8_DRV PG8 Multi_Driving Select 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3

#### 8.7.5.27 0x013C PG Multi\_Driving Register 2 (Default Value: 0x0000\_0111)

Offset: 0x013C			Register Name: PG_DRV2
Bit	Read/Write	Default/Hex	Description
31:10	/	/	/
9:8	R/W	0x1	PG18_DRV PG18 Multi_Driving Select 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3

Offset: 0x013C			Register Name: PG_DRV2
Bit	Read/Write	Default/Hex	Description
7:6	/	/	/
5:4	R/W	0x1	PG17_DRV PG17 Multi_Driving Select 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3
3:2	/	/	/
1:0	R/W	0x1	PG16_DRV PG16 Multi_Driving Select 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3

#### 8.7.5.28 0x0144 PG Pull Register 0 (Default Value: 0x0000\_0000)

Offset: 0x0144			Register Name: PG_PULL0
Bit	Read/Write	Default/Hex	Description
31:30	R/W	0x0	PG15_PULL PG15 Pull_up or down Select. 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
29:28	R/W	0x0	PG14_PULL PG14 Pull_up or down Select. 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
27:26	R/W	0x0	PG13_PULL PG13 Pull_up or down Select. 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
25:24	R/W	0x0	PG12_PULL PG12 Pull_up or down Select. 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
23:22	R/W	0x0	PG11_PULL PG11 Pull_up or down Select. 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved

Offset: 0x0144			Register Name: PG_PULL0
Bit	Read/Write	Default/Hex	Description
21:20	R/W	0x0	<b>PG10_PULL</b> PG10 Pull_up or down Select. 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
19:18	R/W	0x0	<b>PG9_PULL</b> PG9 Pull_up or down Select. 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
17:16	R/W	0x0	<b>PG8_PULL</b> PG8 Pull_up or down Select. 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
15:14	R/W	0x0	<b>PG7_PULL</b> PG7 Pull_up or down Select. 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
13:12	R/W	0x0	<b>PG6_PULL</b> PG6 Pull_up or down Select. 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
11:10	R/W	0x0	<b>PG5_PULL</b> PG5 Pull_up or down Select. 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
9:8	R/W	0x0	<b>PG4_PULL</b> PG4 Pull_up or down Select. 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
7:6	R/W	0x0	<b>PG3_PULL</b> PG3 Pull_up or down Select. 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
5:4	R/W	0x0	<b>PG2_PULL</b> PG2 Pull_up or down Select. 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved

Offset: 0x0144			Register Name: PG_PULL0
Bit	Read/Write	Default/Hex	Description
3:2	R/W	0x0	PG1_PULL PG1 Pull_up or down Select. 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
1:0	R/W	0x0	PGO_PULL PGO Pull_up or down Select. 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved

#### 8.7.5.29 0x0148 PG Pull Register 1 (Default Value: 0x0000\_0000)

Offset: 0x0148			Register Name: PG_PULL1
Bit	Read/Write	Default/Hex	Description
31:6	/	/	/
5:4	R/W	0x0	PG18_PULL PG18 Pull_up or down Select. 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
3:2	R/W	0x0	PG17_PULL PG17 Pull_up or down Select. 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved
1:0	R/W	0x0	PG16_PULL PG16 Pull_up or down Select. 00: Pull_up/down disable    01: Pull_up 10: Pull_down                11: Reserved

#### 8.7.5.30 0x0220 PB External Interrupt Configure Register 0 (Default Value: 0x0000\_0000)

Offset: 0x0220			Register Name: PB_EINT_CFG0
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	Reserved
27:24	R/W	0x0	Reserved
23:20	R/W	0x0	Reserved

Offset: 0x0220			Register Name:PB_EINT_CFG0
Bit	Read/Write	Default/Hex	Description
19:16	R/W	0x0	Reserved
15:12	R/W	0x0	Reserved
11:8	R/W	0x0	Reserved
7:4	R/W	0x0	EINT1_CFG External INT1 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
3:0	R/W	0x0	EINT0_CFG External INT0 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved

#### 8.7.5.31 0x0224 PB External Interrupt Configure Register 1 (Default Value: 0x0000\_0000)

Offset: 0x0224			Register Name:PB_EINT_CFG1
Bit	Read/Write	Default/Hex	Description
31:20	/	/	/
19:16	R/W	0x0	EINT12_CFG External INT12 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved

Offset: 0x0224			Register Name:PB_EINT_CFG1
Bit	Read/Write	Default/Hex	Description
15:12	R/W	0x0	<p>EINT11_CFG External INT11 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved</p>
11:8	R/W	0x0	<p>EINT10_CFG External INT10 Mode. 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved</p>
7:4	R/W	0x0	<p>EINT9_CFG External INT9 Mode. 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved</p>
3:0	R/W	0x0	<p>EINT8_CFG External INT8 Mode. 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved</p>

**8.7.5.32 0x0230 PB External Interrupt Control Register (Default Value: 0x0000\_0000)**

Offset: 0x0230			Register Name: PB_EINT_CTL
Bit	Read/Write	Default/Hex	Description
31:13	/	/	/
12	R/W	0x0	EINT12_CTL External INT12 Enable 0: Disable 1: Enable
11	R/W	0x0	EINT11_CTL External INT11 Enable 0: Disable 1: Enable
10	R/W	0x0	EINT10_CTL External INT10 Enable 0: Disable 1: Enable
9	R/W	0x0	EINT9_CTL External INT9 Enable 0: Disable 1: Enable
8	R/W	0x0	EINT8_CTL External INT8 Enable 0: Disable 1: Enable
7	R/W	0x0	Reserved
6	R/W	0x0	Reserved
5	R/W	0x0	Reserved
4	R/W	0x0	Reserved
3	R/W	0x0	Reserved
2	R/W	0x0	Reserved
1	R/W	0x0	EINT1_CTL External INT1 Enable 0: Disable 1: Enable

Offset: 0x0230			Register Name: PB_EINT_CTL
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	EINT0_CTL External INTO Enable 0: Disable 1: Enable

#### 8.7.5.33 0x0234 PB External Interrupt Status Register (Default Value: 0x0000\_0000)

Offset: 0x0234			Register Name: PB_EINT_STATUS
Bit	Read/Write	Default/Hex	Description
31:13	/	/	/
12	R/W	0x0	EINT12_STATUS External INT12 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
11	R/W	0x0	EINT11_STATUS External INT11 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
10	R/W	0x0	EINT10_STATUS External INT10 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
9	R/W	0x0	EINT9_STATUS External INT9 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
8	R/W	0x0	EINT8_STATUS External INT8 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear

Offset: 0x0234			Register Name: PB_EINT_STATUS
Bit	Read/Write	Default/Hex	Description
7	R/W1C	0x0	Reserved
6	R/W1C	0x0	Reserved
5	R/W1C	0x0	Reserved
4	R/W1C	0x0	Reserved
3	R/W1C	0x0	Reserved
2	R/W1C	0x0	Reserved
1	R/W	0x0	EINT1_STATUS External INT1 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
0	R/W	0x0	EINT0_STATUS External INTO Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear

#### 8.7.5.34 0x0238 PB External Interrupt Debounce Register (Default Value: 0x0000\_0000)

Offset: 0x0238			Register Name: PB_EINT_DEB
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6:4	R/W	0x0	DEB_CLK_PRE_SCALE Debounce Clock Pre_scale n The selected clock source is prescaled by $2^n$ .
3:1	/	/	/
0	R/W	0x0	PIO_INT_CLK_SELECT PIO Interrupt Clock Select 0: LOSC 32KHz 1: HOSC 24MHz

**8.7.5.35 0x0240 PC External Interrupt Configure Register 0 (Default Value: 0x0000\_0000)**

Offset: 0x0240			Register Name:PC_EINT_CFG0
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	EINT7_CFG External INT7 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
27:24	R/W	0x0	EINT6_CFG External INT6 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
23:20	R/W	0x0	EINT5_CFG External INT5 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
19:16	R/W	0x0	EINT4_CFG External INT4 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved

Offset: 0x0240			Register Name:PC_EINT_CFG0
Bit	Read/Write	Default/Hex	Description
15:12	R/W	0x0	EINT3_CFG External INT3 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
11:8	R/W	0x0	EINT2_CFG External INT2 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
7:4	R/W	0x0	Reserved
3:0	R/W	0x0	Reserved

#### 8.7.5.36 0x0250 PC External Interrupt Control Register (Default Value: 0x0000\_0000)

Offset: 0x0250			Register Name: PC_EINT_CTL
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R/W	0x0	EINT7_CTL External INT7 Enable 0: Disable 1: Enable
6	R/W	0x0	EINT6_CTL External INT6 Enable 0: Disable 1: Enable

Offset: 0x0250			Register Name: PC_EINT_CTL
Bit	Read/Write	Default/Hex	Description
5	R/W	0x0	EINT5_CTL External INT5 Enable 0: Disable 1: Enable
4	R/W	0x0	EINT4_CTL External INT4 Enable 0: Disable 1: Enable
3	R/W	0x0	EINT3_CTL External INT3 Enable 0: Disable 1: Enable
2	R/W	0x0	EINT2_CTL External INT2 Enable 0: Disable 1: Enable
1	R/W	0x0	Reserved
0	R/W	0x0	Reserved

#### 8.7.5.37 0x0254 PC External Interrupt Status Register (Default Value: 0x0000\_0000)

Offset: 0x0254			Register Name: PC_EINT_STATUS
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R/W1C	0x0	EINT7_STATUS External INT7 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
6	R/W1C	0x0	EINT6_STATUS External INT6 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear

Offset: 0x0254			Register Name: PC_EINT_STATUS
Bit	Read/Write	Default/Hex	Description
5	R/W1C	0x0	EINT5_STATUS External INT5 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
4	R/W1C	0x0	EINT4_STATUS External INT4 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
3	R/W1C	0x0	EINT3_STATUS External INT3 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
2	R/W1C	0x0	EINT2_STATUS External INT2 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
1	R/W	0x0	Reserved
0	R/W	0x0	Reserved

#### 8.7.5.38 0x0258 PC External Interrupt Debounce Register (Default Value: 0x0000\_0000)

Offset: 0x0258			Register Name: PC_EINT_DEB
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6:4	R/W	0x0	DEB_CLK_PRE_SCALE Debounce Clock Pre_scale n The selected clock source is prescaled by $2^n$ .
3:1	/	/	/

Offset: 0x0258			Register Name: PC_EINT_DEB
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	PIO_INT_CLK_SELECT PIO Interrupt Clock Select 0: LOSC 32KHz 1: HOSC 24MHz

#### 8.7.5.39 0x0260 PD External Interrupt Configure Register 0 (Default Value: 0x0000\_0000)

Offset: 0x0260			Register Name: PD_EINT_CFG0
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	EINT7_CFG External INT7 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
27:24	R/W	0x0	EINT6_CFG External INT6 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
23:20	R/W	0x0	EINT5_CFG External INT5 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved

Offset: 0x0260			Register Name:PD_EINT_CFG0
Bit	Read/Write	Default/Hex	Description
19:16	R/W	0x0	EINT4_CFG External INT4 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
15:12	R/W	0x0	EINT3_CFG External INT3 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
11:8	R/W	0x0	EINT2_CFG External INT2 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
7:4	R/W	0x0	EINT1_CFG External INT1 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved

Offset: 0x0260			Register Name:PD_EINT_CFG0
Bit	Read/Write	Default/Hex	Description
3:0	R/W	0x0	EINT0_CFG External INT0 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved

#### 8.7.5.40 0x0264 PD External Interrupt Configure Register 1 (Default Value: 0x0000\_0000)

Offset: 0x0264			Register Name:PD_EINT_CFG1
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	Reserved
27:24	R/W	0x0	Reserved
23:20	R/W	0x0	Reserved
19:16	R/W	0x0	Reserved
15:12	R/W	0x0	Reserved
11:8	R/W	0x0	Reserved
7:4	R/W	0x0	EINT9_CFG External INT9 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
3:0	R/W	0x0	EINT8_CFG External INT8 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved

**8.7.5.41 0x0270 PD External Interrupt Control Register (Default Value: 0x0000\_0000)**

Offset: 0x0270			Register Name: PD_EINT_CTL
Bit	Read/Write	Default/Hex	Description
31:23	/	/	/
22	R/W	0x0	Reserved
21	R/W	0x0	Reserved
20	R/W	0x0	Reserved
19	R/W	0x0	Reserved
18	R/W	0x0	Reserved
17	R/W	0x0	Reserved
16	R/W	0x0	Reserved
15	R/W	0x0	Reserved
14	R/W	0x0	Reserved
13	R/W	0x0	Reserved
12	R/W	0x0	Reserved
11	R/W	0x0	Reserved
10	R/W	0x0	Reserved
9	R/W	0x0	EINT9_CTL External INT9 Enable 0: Disable 1: Enable
8	R/W	0x0	EINT8_CTL External INT8 Enable 0: Disable 1: Enable
7	R/W	0x0	EINT7_CTL External INT7 Enable 0: Disable 1: Enable
6	R/W	0x0	EINT6_CTL External INT6 Enable 0: Disable 1: Enable

Offset: 0x0270			Register Name: PD_EINT_CTL
Bit	Read/Write	Default/Hex	Description
5	R/W	0x0	EINT5_CTL External INT5 Enable 0: Disable 1: Enable
4	R/W	0x0	EINT4_CTL External INT4 Enable 0: Disable 1: Enable
3	R/W	0x0	EINT3_CTL External INT3 Enable 0: Disable 1: Enable
2	R/W	0x0	EINT2_CTL External INT2 Enable 0: Disable 1: Enable
1	R/W	0x0	EINT1_CTL External INT1 Enable 0: Disable 1: Enable
0	R/W	0x0	EINT0_CTL External INT0 Enable 0: Disable 1: Enable

#### 8.7.5.42 0x0274 PD External Interrupt Status Register (Default Value: 0x0000\_0000)

Offset: 0x0274			Register Name: PD_EINT_STATUS
Bit	Read/Write	Default/Hex	Description
31:23	/	/	/
22	R/W1C	0x0	Reserved
21	R/W1C	0x0	Reserved
20	R/W1C	0x0	Reserved
19	R/W1C	0x0	Reserved
18	R/W1C	0x0	Reserved

Offset: 0x0274			Register Name: PD_EINT_STATUS
Bit	Read/Write	Default/Hex	Description
17	R/W1C	0x0	Reserved
16	R/W1C	0x0	Reserved
15	R/W1C	0x0	Reserved
14	R/W1C	0x0	Reserved
13	R/W1C	0x0	Reserved
12	R/W1C	0x0	Reserved
11	R/W1C	0x0	Reserved
10	R/W1C	0x0	Reserved
9	R/W1C	0x0	EINT9_STATUS External INT9 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
8	R/W1C	0x0	EINT8_STATUS External INT8 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
7	R/W1C	0x0	EINT7_STATUS External INT7 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
6	R/W1C	0x0	EINT6_STATUS External INT6 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
5	R/W1C	0x0	EINT5_STATUS External INT5 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear

Offset: 0x0274			Register Name: PD_EINT_STATUS
Bit	Read/Write	Default/Hex	Description
4	R/W1C	0x0	EINT4_STATUS External INT4 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
3	R/W1C	0x0	EINT3_STATUS External INT3 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
2	R/W1C	0x0	EINT2_STATUS External INT2 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
1	R/W1C	0x0	EINT1_STATUS External INT1 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
0	R/W1C	0x0	EINT0_STATUS External INT0 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear

#### 8.7.5.43 0x0278 PD External Interrupt Debounce Register (Default Value: 0x0000\_0000)

Offset: 0x0278			Register Name: PD_EINT_DEB
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6:4	R/W	0x0	DEB_CLK_PRE_SCALE Debounce Clock Pre_scale n The selected clock source is prescaled by $2^n$ .
3:1	/	/	/

Offset: 0x0278			Register Name: PD_EINT_DEB
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	PIO_INT_CLK_SELECT PIO Interrupt Clock Select 0: LOSC 32KHz 1: HOSC 24MHz

#### 8.7.5.44 0x02A0 PF External Interrupt Configure Register 0 (Default Value: 0x0000\_0000)

Offset: 0x02A0			Register Name: PF_EINT_CFG0
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27:24	R/W	0x0	EINT6_CFG External INT6 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
23:20	R/W	0x0	EINT5_CFG External INT5 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
19:16	R/W	0x0	EINT4_CFG External INT4 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved

Offset: 0x02A0			Register Name: PF_EINT_CFG0
Bit	Read/Write	Default/Hex	Description
15:12	R/W	0x0	EINT3_CFG External INT3 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
11:8	R/W	0x0	EINT2_CFG External INT2 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
7:4	R/W	0x0	EINT1_CFG External INT1 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
3:0	R/W	0x0	EINT0_CFG External INT0 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved

**8.7.5.45 0x02B0 PF External Interrupt Control Register (Default Value: 0x0000\_0000)**

Offset: 0x02B0			Register Name: PF_EINT_CTL
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6	R/W	0x0	EINT6_CTL External INT6 Enable 0: Disable 1: Enable
5	R/W	0x0	EINT5_CTL External INT5 Enable 0: Disable 1: Enable
4	R/W	0x0	EINT4_CTL External INT4 Enable 0: Disable 1: Enable
3	R/W	0x0	EINT3_CTL External INT3 Enable 0: Disable 1: Enable
2	R/W	0x0	EINT2_CTL External INT2 Enable 0: Disable 1: Enable
1	R/W	0x0	EINT1_CTL External INT1 Enable 0: Disable 1: Enable
0	R/W	0x0	EINT0_CTL External INT0 Enable 0: Disable 1: Enable

**8.7.5.46 0x02B4 PF External Interrupt Status Register (Default Value: 0x0000\_0000)**

Offset: 0x02B4			Register Name: PF_EINT_STATUS
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6	R/W1C	0x0	EINT6_STATUS External INT6 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
5	R/W1C	0x0	EINT5_STATUS External INT5 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
4	R/W1C	0x0	EINT4_STATUS External INT4 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
3	R/W1C	0x0	EINT3_STATUS External INT3 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
2	R/W1C	0x0	EINT2_STATUS External INT2 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
1	R/W1C	0x0	EINT1_STATUS External INT1 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear

Offset: 0x02B4			Register Name: PF_EINT_STATUS
Bit	Read/Write	Default/Hex	Description
0	R/W1C	0x0	EINT0_STATUS External INTO Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear

#### 8.7.5.47 0x02B8 PF External Interrupt Debounce Register (Default Value: 0x0000\_0000)

Offset: 0x02B8			Register Name: PF_EINT_DEB
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6:4	R/W	0x0	DEB_CLK_PRE_SCALE Debounce Clock Pre_scale n The selected clock source is prescaled by $2^n$ .
3:1	/	/	/
0	R/W	0x0	PIO_INT_CLK_SELECT PIO Interrupt Clock Select 0: LOSC 32KHz 1: HOSC 24MHz

#### 8.7.5.48 0x02C0 PG External Interrupt Configure Register 0 (Default Value: 0x0000\_0000)

Offset: 0x02C0			Register Name: PG_EINT_CFG0
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	EINT7_CFG External INT7 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved

Offset: 0x02C0			Register Name:PG_EINT_CFG0
Bit	Read/Write	Default/Hex	Description
27:24	R/W	0x0	EINT6_CFG External INT6 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
23:20	R/W	0x0	EINT5_CFG External INT5 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
19:16	R/W	0x0	EINT4_CFG External INT4 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
15:12	R/W	0x0	EINT3_CFG External INT3 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved

Offset: 0x02C0			Register Name:PG_EINT_CFG0
Bit	Read/Write	Default/Hex	Description
11:8	R/W	0x0	EINT2_CFG External INT2 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
7:4	R/W	0x0	EINT1_CFG External INT1 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
3:0	R/W	0x0	EINT0_CFG External INT0 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved

**8.7.5.49 0x02C4 PG External Interrupt Configure Register 1 (Default Value: 0x0000\_0000)**

Offset: 0x02C4			Register Name: PG_EINT_CFG1
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	EINT15_CFG External INT15 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
27:24	R/W	0x0	EINT14_CFG External INT14 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
23:20	R/W	0x0	EINT13_CFG External INT13 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
19:16	R/W	0x0	EINT12_CFG External INT12 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved

Offset: 0x02C4			Register Name: PG_EINT_CFG1
Bit	Read/Write	Default/Hex	Description
15:12	R/W	0x0	EINT11_CFG External INT11 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
11:8	R/W	0x0	EINT10_CFG External INT10 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
7:4	R/W	0x0	EINT9_CFG External INT9 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
3:0	R/W	0x0	EINT8_CFG External INT8 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved

**8.7.5.50 0x02C8 PG External Interrupt Configure Register 2 (Default Value: 0x0000\_0000)**

Offset: 0x02C8			Register Name: PG_EINT_CFG2
Bit	Read/Write	Default/Hex	Description
31:12	/	/	/
11:8	R/W	0x0	EINT18_CFG External INT18 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
7:4	R/W	0x0	EINT17_CFG External INT17 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved
3:0	R/W	0x0	EINT16_CFG External INT16 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/Negative) Others: Reserved

**8.7.5.51 0x02D0 PG External Interrupt Control Register (Default Value: 0x0000\_0000)**

Offset: 0x02D0			Register Name: PG_EINT_CTL
Bit	Read/Write	Default/Hex	Description
31:19	/	/	/

Offset: 0x02D0			Register Name: PG_EINT_CTL
Bit	Read/Write	Default/Hex	Description
18	R/W	0x0	EINT18_CTL External INT18 Enable 0: Disable 1: Enable
17	R/W	0x0	EINT17_CTL External INT17 Enable 0: Disable 1: Enable
16	R/W	0x0	EINT16_CTL External INT16 Enable 0: Disable 1: Enable
15	R/W	0x0	EINT15_CTL External INT15 Enable 0: Disable 1: Enable
14	R/W	0x0	EINT14_CTL External INT14 Enable 0: Disable 1: Enable
13	R/W	0x0	EINT13_CTL External INT13 Enable 0: Disable 1: Enable
12	R/W	0x0	EINT12_CTL External INT12 Enable 0: Disable 1: Enable
11	R/W	0x0	EINT11_CTL External INT11 Enable 0: Disable 1: Enable
10	R/W	0x0	EINT10_CTL External INT10 Enable 0: Disable 1: Enable

Offset: 0x02D0			Register Name: PG_EINT_CTL
Bit	Read/Write	Default/Hex	Description
9	R/W	0x0	EINT9_CTL External INT9 Enable 0: Disable 1: Enable
8	R/W	0x0	EINT8_CTL External INT8 Enable 0: Disable 1: Enable
7	R/W	0x0	EINT7_CTL External INT7 Enable 0: Disable 1: Enable
6	R/W	0x0	EINT6_CTL External INT6 Enable 0: Disable 1: Enable
5	R/W	0x0	EINT5_CTL External INT5 Enable 0: Disable 1: Enable
4	R/W	0x0	EINT4_CTL External INT4 Enable 0: Disable 1: Enable
3	R/W	0x0	EINT3_CTL External INT3 Enable 0: Disable 1: Enable
2	R/W	0x0	EINT2_CTL External INT2 Enable 0: Disable 1: Enable
1	R/W	0x0	EINT1_CTL External INT1 Enable 0: Disable 1: Enable

Offset: 0x02D0			Register Name: PG_EINT_CTL
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	EINT0_CTL External INTO Enable 0: Disable 1: Enable

#### 8.7.5.52 0x02D4 PG External Interrupt Status Register (Default Value: 0x0000\_0000)

Offset: 0x02D4			Register Name: PG_EINT_STATUS
Bit	Read/Write	Default/Hex	Description
31:19	/	/	/
18	R/W	0x0	EINT18_STATUS External INT18 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
17	R/W	0x0	EINT17_STATUS External INT17 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
16	R/W	0x0	EINT16_STATUS External INT16 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
15	R/W1C	0x0	EINT15_STATUS External INT15 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
14	R/W1C	0x0	EINT14_STATUS External INT14 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear

Offset: 0x02D4			Register Name: PG_EINT_STATUS
Bit	Read/Write	Default/Hex	Description
13	R/W1C	0x0	EINT13_STATUS External INT13 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
12	R/W1C	0x0	EINT12_STATUS External INT12 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
11	R/W1C	0x0	EINT11_STATUS External INT11 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
10	R/W1C	0x0	EINT10_STATUS External INT10 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
9	R/W1C	0x0	EINT9_STATUS External INT9 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
8	R/W1C	0x0	EINT8_STATUS External INT8 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
7	R/W1C	0x0	EINT7_STATUS External INT7 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear

Offset: 0x02D4			Register Name: PG_EINT_STATUS
Bit	Read/Write	Default/Hex	Description
6	R/W1C	0x0	EINT6_STATUS External INT6 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
5	R/W1C	0x0	EINT5_STATUS External INT5 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
4	R/W1C	0x0	EINT4_STATUS External INT4 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
3	R/W1C	0x0	EINT3_STATUS External INT3 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
2	R/W1C	0x0	EINT2_STATUS External INT2 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
1	R/W1C	0x0	EINT1_STATUS External INT1 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
0	R/W1C	0x0	EINT0_STATUS External INT0 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear

#### 8.7.5.53 0x02D8 PG External Interrupt Debounce Register (Default Value: 0x0000\_0000)

Offset: 0x02D8			Register Name: PG_EINT_DEB
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6:4	R/W	0x0	DEB_CLK_PRE_SCALE Debounce Clock Pre_scale n The selected clock source is prescaled by $2^n$ .
3:1	/	/	/
0	R/W	0x0	PIO_INT_CLK_SELECT PIO Interrupt Clock Select 0: LOSC 32KHz 1: HOSC 24MHz

#### 8.7.5.54 0x0340 PIO Group Withstand Voltage Mode Select Register (Default Value: 0x0000\_0000)

When the power domain of GPIO is larger than 1.8 V, the withstand voltage is set to 3.3 V mode, the corresponding value in the 0x0340 register is set to 0.

When the power domain of GPIO is 1.8 V, the withstand voltage is set to 1.8 V mode, the corresponding value in the 0x0340 register is set to 1.

Offset: 0x0340			Register Name: PIO_POW_MOD_SEL
Bit	Read/Write	Default/Hex	Description
31:13	/	/	/
12	R/W	0x0	VCCIO_PWR_MOD_SEL VCC_IO POWER MODE Select 0: 3.3 V 1: 1.8 V
11:7	/	/	/
6	R/W	0x0	PG_PWR_MOD_SEL PG_POWER MODE Select 0: 3.3 V 1: 1.8 V If PG_Port Power Source selects VCC_IO, this bit is invalid.

Offset: 0x0340			Register Name: PIO_POW_MOD_SEL
Bit	Read/Write	Default/Hex	Description
5	R/W	0x0	PF_PWR_MOD_SEL PF_POWER MODE Select 0: 3.3 V 1: 1.8 V If PF_Port Power Source selects VCC_IO, this bit is invalid.
4	R/W	0x0	Reserved
3	R/W	0x0	PD_PWR_MOD_SEL PD_POWER MODE Select 0: 3.3 V 1: 1.8 V If PD_Port Power Source selects VCC_IO, this bit is invalid.
2	R/W	0x0	PC_PWR_MOD_SEL PC_POWER MODE Select 0: 3.3 V 1: 1.8 V If PC_Port Power Source selects VCC_IO, this bit is invalid.
1:0	/	/	/

#### 8.7.5.55 0x0344 PIO Group Withstand Voltage Mode Select Control Register (Default Value: 0x0000\_0000)

For 1.8 V and 3.3 V power, the withstand function is enabled by default, the corresponding bit in the 0x0344 register is set to 0.

For 2.5 V power, the withstand function is disabled, the corresponding bit in the 0x0344 register is set to 1, and the corresponding withstand voltage in the 0x0340 register needs to be set to 3.3 V.

Offset: 0x0344			Register Name: PIO_POW_MS_CTL
Bit	Read/Write	Default/Hex	Description
31:13	/	/	/
12	R/W	0x0	VCCIO_WS_VOL_MOD_SEL VCC_IO Withstand Voltage Mode Select Control 0: Enable 1: Disable
11:7	/	/	/

Offset: 0x0344			Register Name: PIO_POW_MS_CTL
Bit	Read/Write	Default/Hex	Description
6	R/W	0x0	VCC_PG_WS_VOL_MOD_SEL VCC_PG Withstand Voltage Mode Select Control 0: Enable 1: Disable
5	R/W	0x0	VCC_PF_WS_VOL_MOD_SEL VCC_PF Withstand Voltage Mode Select Control 0: Enable 1: Disable
4	R/W	0x0	Reserved
3	R/W	0x0	VCC_PD_WS_VOL_MOD_SEL VCC_PD Withstand Voltage Mode Select Control 0: Enable 1: Disable
2	R/W	0x0	VCC_PC_WS_VOL_MOD_SEL VCC_PC Withstand Voltage Mode Select Control 0: Enable 1: Disable
1:0	/	/	/

#### 8.7.5.56 0x0348 PIO Group Power Value Register (Default Value: 0x0000\_0000)

When the reading value of the 0x0348 register is 0, it indicates that the IO power voltage is greater than 2.5 V.  
When the reading value of the 0x0348 register is 1, it indicates that the IO power voltage is less than 2.0 V.

Offset: 0x0348			Register Name: PIO_POW_VAL
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R	0x0	VCCIO_PWR_VAL VCC_IO Power Value
15:7	/	/	/
6	R	0x0	PG_PWR_VAL PG_Port Power Value If PG_Port power source selects VCC_IO, this bit is invalid.
5	R	0x0	PF_PWR_VAL PF_Port Power Value If PF_Port power source selects VCC_IO, this bit is invalid.

Offset: 0x0348			Register Name: PIO_POW_VAL
Bit	Read/Write	Default/Hex	Description
4	R	0x0	Reserved
3	R	0x0	PD_PWR_VAL PD_Port Power Value If PD_Port power source selects VCC_IO, this bit is invalid.
2	R	0x0	PC_PWR_VAL PC_Port Power Value If PC_Port power source selects VCC_IO, this bit is invalid.
1:0	/	/	/

#### 8.7.5.57 0x0350 PIO Group Power Voltage Select Control Register (Default Value: 0x0000\_0001)

Offset: 0x0350			Register Name: PIO_POW_VOL_SEL_CTL
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x1	VCC-PF Power Voltage Select Control 0: 1.8 V 1: 3.3 V

#### 8.7.5.58 0x0404 PB Secure Register (Default Value: 0x0000\_0000)

Offset: 0x0404			Register Name: PB_SECURE
Bit	Read/Write	Default/Hex	Description
31:13	R/W	0x0	Reserved
12	R/W	0x0	POR12_SEC_SEL Port12 secure select 0: Not secure 1: Secure
11	R/W	0x0	POR11_SEC_SEL Port11 secure select 0: Not secure 1: Secure

Offset: 0x0404			Register Name: PB_SECURE
Bit	Read/Write	Default/Hex	Description
10	R/W	0x0	PORT10_SEC_SEL Port10 secure select 0: Not secure 1: Secure
9	R/W	0x0	PORT9_SEC_SEL Port9 secure select 0: Not secure 1: Secure
8	R/W	0x0	PORT8_SEC_SEL Port8 secure select 0: Not secure 1: Secure
7	R/W	0x0	Reserved
6	R/W	0x0	Reserved
5	R/W	0x0	Reserved
4	R/W	0x0	Reserved
3	R/W	0x0	Reserved
2	R/W	0x0	Reserved
1	R/W	0x0	PORT1_SEC_SEL Port1 secure select 0: Not secure 1: Secure
0	R/W	0x0	PORT0_SEC_SEL Port0 secure select 0: Not secure 1: Secure

#### 8.7.5.59 0x0408 PC Secure Register (Default Value: 0x0000\_0000)

Offset: 0x0408			Register Name: PC_SECURE
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/

Offset: 0x0408			Register Name: PC_SECURE
Bit	Read/Write	Default/Hex	Description
7	R/W	0x0	PORT7_SEC_SEL Port7 secure select 0: Not secure 1: Secure
6	R/W	0x0	PORT6_SEC_SEL Port6 secure select 0: Not secure 1: Secure
5	R/W	0x0	PORT5_SEC_SEL Port5 secure select 0: Not secure 1: Secure
4	R/W	0x0	PORT4_SEC_SEL Port4 secure select 0: Not secure 1: Secure
3	R/W	0x0	PORT3_SEC_SEL Port3 secure select 0: Not secure 1: Secure
2	R/W	0x0	PORT2_SEC_SEL Port2 secure select 0: Not secure 1: Secure
1	R/W	0x0	Reserved
0	R/W	0x0	Reserved

#### 8.7.5.60 0x040C PD Secure Register (Default Value: 0x0000\_0000)

Offset: 0x040C			Register Name: PD_SECURE
Bit	Read/Write	Default/Hex	Description
31:10	R/W	0x0	Reserved

Offset: 0x040C			Register Name: PD_SECURE
Bit	Read/Write	Default/Hex	Description
9	R/W	0x0	PORT9_SEC_SEL Port9 secure select 0: Not secure 1: Secure
8	R/W	0x0	PORT8_SEC_SEL Port8 secure select 0: Not secure 1: Secure
7	R/W	0x0	PORT7_SEC_SEL Port7 secure select 0: Not secure 1: Secure
6	R/W	0x0	PORT6_SEC_SEL Port6 secure select. 0: Not secure 1: Secure
5	R/W	0x0	PORT5_SEC_SEL Port5 secure select 0: Not secure 1: Secure
4	R/W	0x0	PORT4_SEC_SEL Port4 secure select 0: Not secure 1: Secure
3	R/W	0x0	PORT3_SEC_SEL Port3 secure select 0: Not secure 1: Secure
2	R/W	0x0	PORT2_SEC_SEL Port2 secure select 0: Not secure 1: Secure
1	R/W	0x0	PORT1_SEC_SEL Port1 secure select 0: Not secure 1: Secure

Offset: 0x040C			Register Name: PD_SECURE
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	PORT0_SEC_SEL Port0 secure select 0: Not secure 1: Secure

#### 8.7.5.61 0x0414 PF Secure Register (Default Value: 0x0000\_0000)

Offset: 0x0414			Register Name: PF_SECURE
Bit	Read/Write	Default/Hex	Description
31:7	R/W	0x0	Reserved
6	R/W	0x0	PORT6_SEC_SEL Port6 secure select 0: Not secure 1: Secure
5	R/W	0x0	PORT5_SEC_SEL Port5 secure select 0: Not secure 1: Secure
4	R/W	0x0	PORT4_SEC_SEL Port4 secure select 0: Not secure 1: Secure
3	R/W	0x0	PORT3_SEC_SEL Port3 secure select 0: Not secure 1: Secure
2	R/W	0x0	PORT2_SEC_SEL Port2 secure select 0: Not secure 1: Secure
1	R/W	0x0	PORT1_SEC_SEL Port1 secure select 0: Not secure 1: Secure

Offset: 0x0414			Register Name: PF_SECURE
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	PORT0_SEC_SEL Port0 secure select 0: Not secure 1: Secure

#### 8.7.5.62 0x0418 PG Secure Register (Default Value: 0x0000\_0000)

Offset: 0x0418			Register Name: PG_SECURE
Bit	Read/Write	Default/Hex	Description
31:19	R/W	0x0	Reserved
18	R/W	0x0	PORT18_SEC_SEL Port18 secure select 0: Not secure 1: Secure
17	R/W	0x0	PORT17_SEC_SEL Port17 secure select 0: Not secure 1: Secure
16	R/W	0x0	PORT16_SEC_SEL Port16 secure select 0: Not secure 1: Secure
15	R/W	0x0	PORT15_SEC_SEL Port15 secure select 0: Not secure 1: Secure
14	R/W	0x0	PORT14_SEC_SEL Port14 secure select 0: Not secure 1: Secure
13	R/W	0x0	PORT13_SEC_SEL Port13 secure select 0: Not secure 1: Secure

Offset: 0x0418			Register Name: PG_SECURE
Bit	Read/Write	Default/Hex	Description
12	R/W	0x0	PORT12_SEC_SEL Port12 secure select 0: Not secure 1: Secure
11	R/W	0x0	PORT11_SEC_SEL Port11 secure select 0: Not secure 1: Secure
10	R/W	0x0	PORT10_SEC_SEL Port10 secure select 0: Not secure 1: Secure
9	R/W	0x0	PORT9_SEC_SEL Port9 secure select 0: Not secure 1: Secure
8	R/W	0x0	PORT8_SEC_SEL Port8 secure select 0: Not secure 1: Secure
7	R/W	0x0	PORT7_SEC_SEL Port7 secure select 0: Not secure 1: Secure
6	R/W	0x0	PORT6_SEC_SEL Port6 secure select. 0: Not secure 1: Secure
5	R/W	0x0	PORT5_SEC_SEL Port5 secure select 0: Not secure 1: Secure
4	R/W	0x0	PORT4_SEC_SEL Port4 secure select 0: Not secure 1: Secure

Offset: 0x0418			Register Name: PG_SECURE
Bit	Read/Write	Default/Hex	Description
3	R/W	0x0	PORT3_SEC_SEL Port3 secure select 0: Not secure 1: Secure
2	R/W	0x0	PORT2_SEC_SEL Port2 secure select 0: Not secure 1: Secure
1	R/W	0x0	PORT1_SEC_SEL Port1 secure select 0: Not secure 1: Secure
0	R/W	0x0	PORT0_SEC_SEL Port0 secure select 0: Not secure 1: Secure

## 8.8 GPADC

### 8.8.1 Overview

The General Purpose ADC (GPADC) can convert the external signal into a certain proportion of digital value, to realize the measurement of analog signal, which can be applied to power detection and key detection. This ADC is a type of successive approximation register (SAR) A/D converter.

The GPADC has the following features:

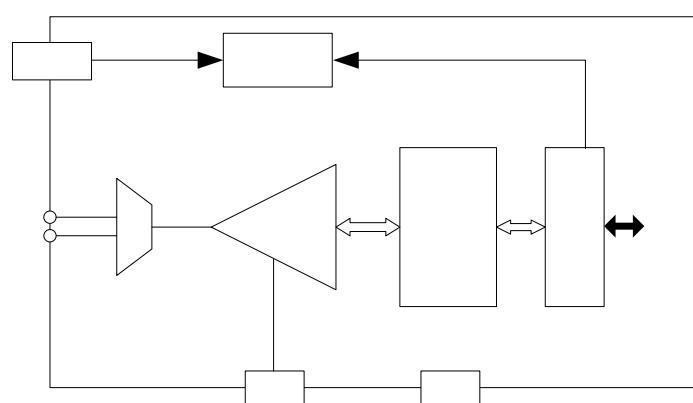
- One independent channel
- 12-bit sampling resolution and 8-bit precision
- 64 FIFO depth of data register
- Power reference voltage: AVCC, and analog input voltage range: 0 to AVCC
- Maximum sampling frequency: 1 MHz
- Supports data compare and interrupt
- Supports three operation modes

- Single conversion mode
- Continuous conversion mode
- Burst conversion mode

### 8.8.2 Block Diagram

Figure 8-70 shows the block diagram of the GPADC.

**Figure 8-70 GPADC Block Diagram**



### 8.8.3 Functional Description

#### 8.8.3.1 External Signals

The following table describes the external signals of the GPADC.

**Table 8-27 GPADC External Signals**

Signal	Description	Type
GPADCO	ADC Input Channel0	AI

#### 8.8.3.2 Clock Sources

The GPADC has one clock source. The following table describes the clock source for GPADC. Users can see section 3.3 “[CCU](#)” for clock setting, configuration, and gating information.

**Table 8-28 GPADC Clock Sources**

Clock Sources	Description
HOSC	24 MHz

### 8.8.3.3 GPADC Work Mode

- **Single conversion mode**

The GPADC completes one conversion in a specified channel, the converted data is updated at the data register of the corresponding channel.

- **Continuous conversion mode**

The GPADC has continuous conversion in a specified channel until the software stops, the converted data is updated at the data register of the corresponding channel.

- **Burst conversion mode**

The GPADC samples and converts in a specified channel, and sequentially stores the results in FIFO.

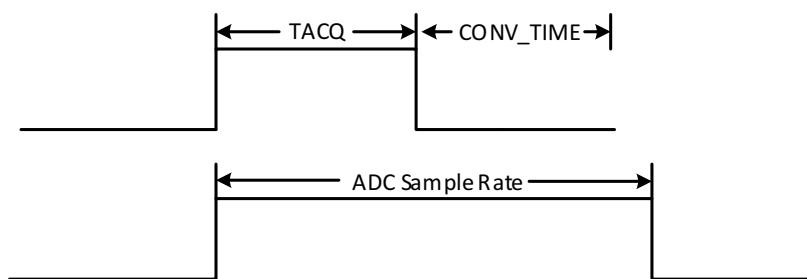
### 8.8.3.4 Clock and Timing Requirements

CLK\_IN = 24 MHz

CONV\_TIME(Conversion Time) =  $1/(24\text{MHz}/14\text{Cycles}) = 0.583 \text{ (us)}$

TACQ > 10RC (R is output impedance of ADC sample circuit, C = 6.4 pF)

ADC Sample Frequency > TACQ+CONV\_TIME

**Figure 8-71 GPADC Clock and Timing Requirement**

### 8.8.3.5 GPADC Calculate Formula

GPADC calculate formula: GPADC\_DATA = Vin/V<sub>REF</sub>\*4095

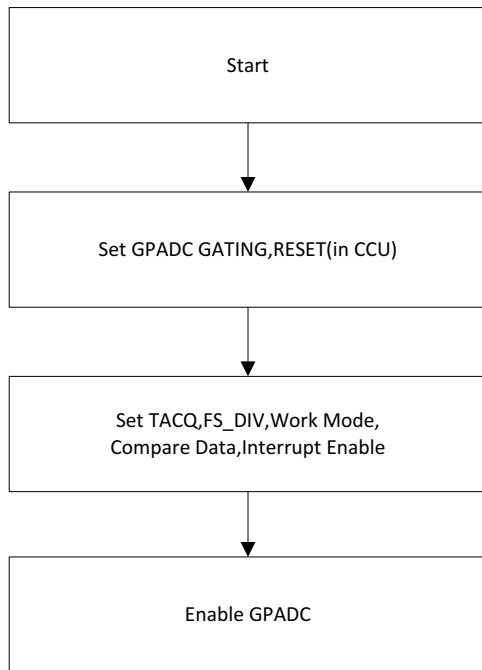
Where:

$$V_{REF} = 1.8 \text{ V}$$

### 8.8.4 Programming Guidelines

The GPADC initial process is as follows.

**Figure 8-72 GPADC Initial Process**



#### (1).Query Mode

- Step 1** Write 0x1 to the bit[16] of [GPADC\\_BGR\\_REG](#) to dessert reset.
- Step 2** Write 0x1 to the bit[0] of [GPADC\\_BGR\\_REG](#) to enable the GPADC clock.
- Step 3** Write 0x2F to the bit[15:0] of [GP\\_SR\\_CON](#) to set the *acquiring time of ADC*.
- Step 4** Write 0x1DF to the bit[31:16] of [GP\\_SR\\_CON](#) to set the *ADC sample frequency divider*.
- Step 5** Write 0x2 to the bit[19:18] of [GP\\_CTRL](#) to set the *continuous conversion mode*.
- Step 6** Write 0x1 to the bit[0] of [GP\\_CS\\_EN](#) to enable the *analog input channel*.
- Step 7** Write 0x1 to the bit[16] of [GP\\_CTRL](#) to enable the ADC function.
- Step 8** Read the bit[0] of [GP\\_DATA\\_INTS](#), if the bit is 1, then data conversion is complete.

**Step 9** Read the bit[11:0] of [GP\\_CHO\\_DATA](#), and calculate voltage value based on GPADC formula.

## (2).Interrupt Mode

**Step 1** Write 0x1 to the bit[16] of [GPADC\\_BGR\\_REG](#) to dessert reset.

**Step 2** Write 0x1 to the bit[0] of [GPADC\\_BGR\\_REG](#) to enable the GPADC clock.

**Step 3** Write 0x2F to the bit[15:0] of [GP\\_SR\\_CON](#) to set the *acquiring time of ADC*.

**Step 4** Write 0x1DF to the bit[31:16] of [GP\\_SR\\_CON](#) to set the *ADC sample frequency divider*.

**Step 5** Write 0x2 to the bit[19:18] of [GP\\_CTRL](#) to set the *continuous conversion mode*.

**Step 6** Write 0x1 to the bit[0] of [GP\\_CS\\_EN](#) to enable the *analog input channel*.

**Step 7** Write 0x1 to the bit[0] of [GP\\_DATA\\_INTC](#) to enable the GPADC data interrupt.

**Step 8** Set the GIC module based on the IRQ 89.

**Step 9** Put interrupt handler address into interrupt vector table based on the IRQ 148.

**Step 10** Write 0x1 to the bit16 of [GP\\_CTRL](#) to enable the ADC function.

**Step 11** Read the bit[11:0] of [GP\\_CHO\\_DATA](#) from the interrupt handler, calculate voltage value based on GPADC formula.

## 8.8.5 Register List

Module Name	Base Address
GPADC	0x02009000

Register Name	Offset	Description
GP_SR_CON	0x0000	GPADC Sample Rate Configure Register
GP_CTRL	0x0004	GPADC Control Register
GP_CS_EN	0x0008	GPADC Compare and Select Enable Register
GP_FIFO_INTC	0x000C	GPADC FIFO Interrupt Control Register
GP_FIFO_INTS	0x0010	GPADC FIFO Interrupt Status Register
GP_FIFO_DATA	0x0014	GPADC FIFO Data Register
GP_CDATA	0x0018	GPADC Calibration Data Register
GP_DATA_L_INTC	0x0020	GPADC Data Low Interrupt Configure Register
GP_DATAH_INTC	0x0024	GPADC Data High Interrupt Configure Register
GP_DATA_INTC	0x0028	GPADC Data Interrupt Configure Register

Register Name	Offset	Description
GP_DATAL_INTS	0x0030	GPADC Data Low Interrupt Status Register
GP_DATAH_INTS	0x0034	GPADC Data High Interrupt Status Register
GP_DATA_INTS	0x0038	GPADC Data Interrupt Status Register
GP_CHO_CMP_DATA	0x0040	GPADC CH0 Compare Data Register
GP_CHO_DATA	0x0080	GPADC CH0 Data Register

## 8.8.6 Register Description

### 8.8.6.1 0x0000 GPADC Sample Rate Configure Register (Default Value: 0x01DF\_002F)

Offset: 0x0000			Register Name: GP_SR_CON
Bit	Read/Write	Default/Hex	Description
31: 16	R/W	0x1DF	FS_DIV ADC sample frequency divider CLK_IN/(n+1) Default value: 50K
15:0	R/W	0x2F	TACQ ADC acquire time (n+1)/CLK_IN Default value: 2 us

### 8.8.6.2 0x0004 GPADC Control Register (Default Value: 0x0080\_0000)

Offset: 0x0004			Register Name: GP_CTRL
Bit	Read/Write	Default/Hex	Description
31:24	R/ W	0x0	ADC_FIRST_DLY ADC First Convert Delay Setting ADC conversion of each channel is delayed by N samples.
23	R/ W	0x1	ADC_AUTOCALI_EN ADC Auto Calibration
22	/	/	/
21:20	R/W	0x0	ADC_OP_BIAS ADC OP Bias Adjust the bandwidth of the ADC amplifier

Offset: 0x0004			Register Name: GP_CTRL
Bit	Read/Write	Default/Hex	Description
19:18	R/W	0x0	GPADC Work Mode 00: Single conversion mode 01: Reserved 10: Continuous conversion mode 11: Burst conversion mode
17	R/W	0x0	ADC_CALI_EN ADC Calibration 1: Start Calibration, it is cleared to 0 after calibration
16	R/W	0x0	ADC_EN ADC Function Enable Before the bit is enabled, configure ADC parameters including the work mode and channel number, etc. 0: Disable 1: Enable <b>Note: When selecting a single conversion mode, the bit can be cleared automatically after the switch is completed.</b>
15:0	/	/	/

#### 8.8.6.3 0x0008 GPADC Compare and Select Enable Register (Default Value: 0x0000\_0000)

Offset: 0x0008			Register Name: GP_CS_EN
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
17	R/W	0x0	Reserved
16	R/W	0x0	ADC_CHO_CMP_EN Channel 0 Compare Enable 0: Disable 1: Enable
15:2	/	/	/
1	R/W	0x0	Reserved
0	R/W	0x0	ADC_CHO_SELECT Analog Input Channel 0 Select 0: Disable 1: Enable

#### 8.8.6.4 0x000C GPADC FIFO Interrupt Control Register (Default Value: 0x0000\_1F00)

Offset: 0x000C			Register Name: GP_FIFO_INTC
Bit	Read/Write	Default/Hex	Description
31:19	/	/	/
18	R/W	0x0	FIFO_DATA_DRQ_EN ADC FIFO Date DRQ Enable 0: Disable 1: Enable
17	R/W	0x0	FIFO_OVERRUN_IRQ_EN ADC FIFO Overrun IRQ Enable 0: Disable 1: Enable
16	R/W	0x0	FIFO_DATA_IRQ_EN ADC FIFO Data Available IRQ Enable 0: Disable 1: Enable
15:14	/	/	/
13:8	R/W	0x1F	FIFO_TRIG_LEVEL Interrupt trigger level for ADC Trigger Level = TXTL + 1
7:5	/	/	/
4	R/WAC	0x0	FIFO_FLUSH ADC FIFO Flush Write '1' to flush TX FIFO, clear automatically to '0'.
3:0	/	/	/

#### 8.8.6.5 0x0010 GPADC FIFO Interrupt Status Register (Default Value: 0x0000\_0000)

Offset: 0x0010			Register Name: GP_FIFO_INTS
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/

Offset: 0x0010			Register Name: GP_FIFO_INTS
Bit	Read/Write	Default/Hex	Description
17	R/W1C	0x0	FIFO_OVERRUN_PENDING ADC FIFO Overrun IRQ Pending 0: No Pending IRQ 1: FIFO Overrun Pending IRQ Write '1' to clear this interrupt or automatically clear if the interrupt condition fails.
16	R/W1C	0x0	FIFO_DATA_PENDING ADC FIFO Data Available Pending Bit 0: NO Pending IRQ 1: FIFO Available Pending IRQ Write '1' to clear this interrupt or automatically clear if the interrupt condition fails.
15:14	/	/	/
13:8	R	0x0	RXA_CNT ADC FIFO available sample word counter
7:0	/	/	/

#### 8.8.6.6 0x0014 GPADC FIFO Data Register (Default Value: 0x0000\_0000)

Offset: 0x0014			Register Name: GP_FIFO_DATA
Bit	Read/Write	Default/Hex	Description
31:12	/	/	/
11:0	R	0x0	GP_FIFO_DATA GPADC Data in FIFO

#### 8.8.6.7 0x0018 GPADC Calibration Data Register (Default Value: 0x0000\_0000)

Offset: 0x0018			Register Name: GP_CDATA
Bit	Read/Write	Default/Hex	Description
31:12	/	/	/
11:0	R/W	0x0	GP_CDATA GPADC Calibration Data

**8.8.6.8 0x0020 GPADC Low Interrupt Configure Register (Default Value: 0x0000\_0000)**

Offset: 0x0020			Register Name: GP_DATA_L_INTC
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1	R/W	0x0	Reserved
0	R/W	0x0	CH0_LOW_IRQ_EN Channel 0 Voltage Low Available Interrupt Enable 0: Disable 1: Enable

**8.8.6.9 0x0024 GPADC High Interrupt Configure Register (Default Value: 0x0000\_0000)**

Offset: 0x0024			Register Name: GP_DATAH_INTC
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1	R/W	0x0	Reserved
0	R/W	0x0	CH0_HIG_IRQ_EN Channel 0 Voltage High Available Interrupt Enable 0: Disable 1: Enable

**8.8.6.10 0x0028 GPADC DATA Interrupt Configure Register (Default Value: 0x0000\_0000)**

Offset: 0x0028			Register Name: GP_DATA_INTC
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1	R/W	0x0	CH1_DATA_IRQ_EN 0: Disable 1: Enable
0	R/W	0x0	CH0_DATA_IRQ_EN 0: Disable 1: Enable

#### 8.8.6.11 0x0030 GPADC Low Interrupt Status Register (Default Value: 0x0000\_0000)

Offset: 0x0030			Register Name: GP_DATA_L_INTS
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1	R/W1C	0x0	Reserved
0	R/W1C	0x0	CH0_LOW_PENGDING Channel 0 Voltage Low Available Interrupt Status 0: NO Pending IRQ 1: Channel 0 Voltage Low Available Pending IRQ Write '1' to clear this interrupt or automatically clear if the interrupt condition fails.

#### 8.8.6.12 0x0034 GPADC High Interrupt Status Register (Default Value: 0x0000\_0000)

Offset: 0x0034			Register Name: GP_DATAH_INTS
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1	R/W1C	0x0	Reserved
0	R/W1C	0x0	CH0_HIG_PENGDING 0: No Pending IRQ 1: Channel 0 Voltage High Available Pending IRQ Write '1' to clear this interrupt or automatically clear if the interrupt condition fails.

#### 8.8.6.13 0x0038 GPADC Data Interrupt Status Register (Default Value: 0x0000\_0000)

Offset: 0x0038			Register Name: GP_DATA_INTS
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1	R/W1C	0x0	Reserved

Offset: 0x0038			Register Name: GP_DATA_INTS
Bit	Read/Write	Default/Hex	Description
0	R/W1C	0x0	<p>CH0_DATA_PENGDING Channel 0 Data Available Interrupt Status 0: No Pending IRQ 1: Channel 0 Data Available Pending IRQ Write ‘1’ to clear this interrupt or automatically clear if the interrupt condition fails.</p>

#### 8.8.6.14 0x0040 GPADC CH0 Compare Data Register (Default Value: 0x0BFF\_0400)

Offset: 0x0040			Register Name: GP_CH0_CMP_DATA
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27:16	R/W	0xBFF	<p>CH0_CMP_HIG_DATA Channel 0 Voltage High Value</p>
15:12	/	/	/
11:0	R/W	0x400	<p>CH0_CMP_LOW_DATA Channel 0 Voltage Low Value</p>

#### 8.8.6.15 0x0080 GPADC CH0 Data Register (Default Value: 0x0000\_0000)

Offset: 0x0080			Register Name: GP_CH0_DATA
Bit	Read/Write	Default/Hex	Description
31:12	/	/	/
11:0	R	0x000	<p>GP_CH0_DATA Channel 0 Data</p>

## 8.9 PWM

### 8.9.1 Overview

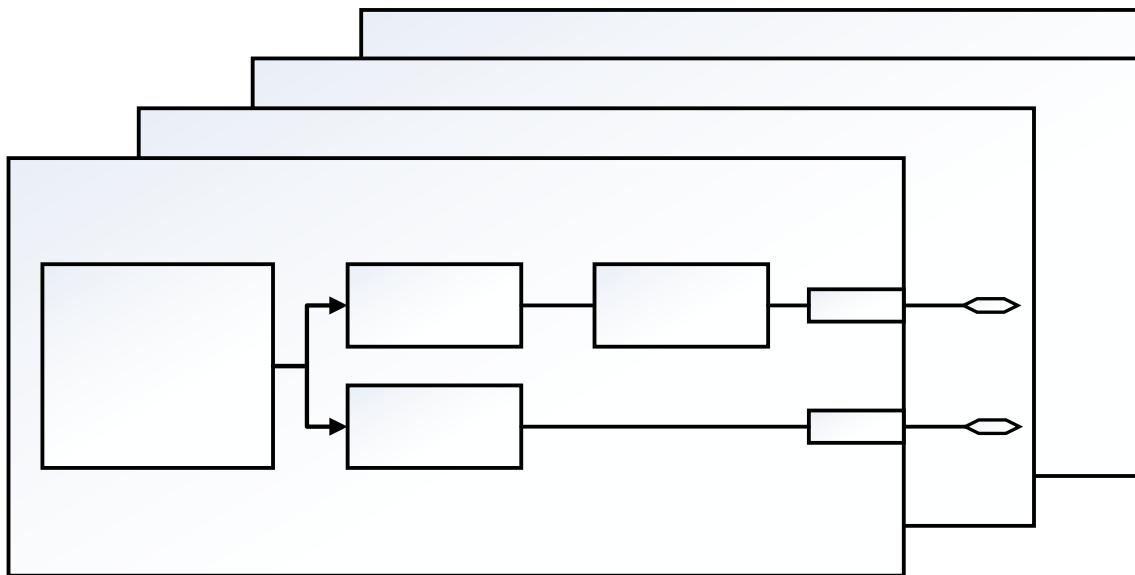
The Pulse Width Modulation (PWM) module can output the configurable PWM waveforms and measure the external input waveforms.

The PWM has the following features:

- Supports 8 independent PWM channels (PWM0 to PWM7)
  - Supports PWM continuous mode output
  - Supports PWM pulse mode output, and the pulse number is configurable
  - Output frequency range: 0 to 24 MHz or 100 MHz
  - Various duty-cycle: 0% to 100%
  - Minimum resolution: 1/65536
- Supports 4 complementary pairs output
  - PWM01 pair (PWM0 + PWM1), PWM23 pair (PWM2 + PWM3), PWM45 pair (PWM4 + PWM5), PWM67 pair (PWM6 + PWM7)
  - Supports dead-zone generator, and the dead-zone time is configurable
- Supports 4 group of PWM channel output for controlling stepping motors
  - Supports any plural channels to form a group, and output the same duty-cycle pulse
  - In group mode, the relative phase of the output waveform for each channel is configurable
- Supports 8 channels capture input
  - Supports rising edge detection and falling edge detection for input waveform pulse
  - Supports pulse-width measurement for input waveform pulse

### 8.9.2 Block Diagram

The PWM includes multi PWM channels. Each channel can generate different PWM waveform by the independent counter and duty-ratio configuration register. Each PWM pair shares one group of clock and dead-zone generator to generate PWM waveform.

**Figure 8-73 PWM Block Diagram**

Each PWM pair consists of 1 clock module, 2 timer logic module, and 1 programmable dead-zone generator.

### 8.9.3 Functional Description

#### 8.9.3.1 External Signals

The following table describes the external signals of the PWM.

**Table 8-29 PWM External Signals**

Signal	Description	Type
PWM0	Pulse Width Module Channel0	I/O
PWM1	Pulse Width Module Channel1	I/O
PWM2	Pulse Width Module Channel2	I/O
PWM3	Pulse Width Module Channel3	I/O
PWM4	Pulse Width Module Channel4	I/O
PWM5	Pulse Width Module Channel5	I/O
PWM6	Pulse Width Module Channel6	I/O
PWM7	Pulse Width Module Channel7	I/O

#### 8.9.3.2 Typical Application

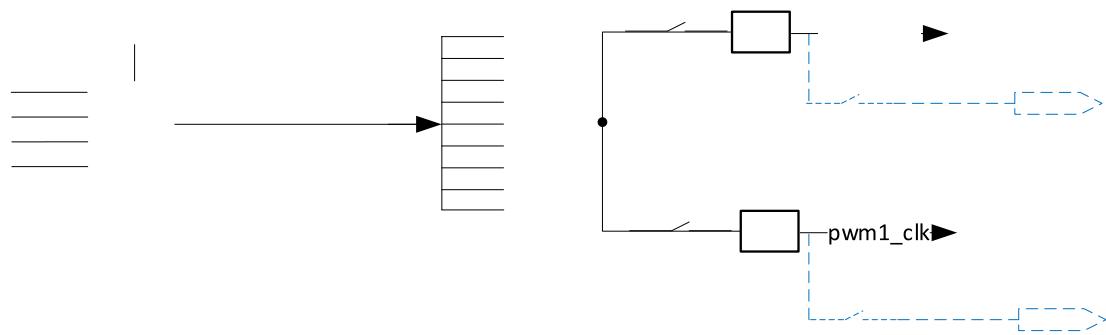
- Suitable for display device, such as LCD

- Suitable for electric motor control

### 8.9.3.3 Clock Controller

Using PWM01 as an example. The other PWM pairs are the same as PWM01.

Figure 8-74 PWM01 Clock Controller Diagram



The clock controller of each PWM pair includes clock source select ([PWM01\\_CLK\\_SRC](#)), 1~256 scaler ([PWM01\\_CLK\\_DIV\\_M](#)). Each PWM channel has the secondary frequency division ([PWM\\_PRESCAL\\_K](#)), clock source bypass ([PWMx\\_CLK\\_BYPASS](#)) and clock switch ([PWMx\\_CLK\\_GATING](#)).

The clock sources have HOSC and APBO. The HOSC comes from the external high-frequency oscillator; the APBO is APBO bus clock.

The bypass function of the clock source is that the clock source directly accesses PWM output, the PWM output waveform is the waveform of the clock controller output. The BYPASS gridlines in the above figure indicate the bypass function of the clock source, see Figure 9-85 for the details about implement. At last, the output clock of the clock controller is sent to the PWM logic module.

### 8.9.3.4 PWM Output

Taking PWM01 as an example, Figure 9-85 indicates the PWM01 output logic diagram. The logic diagrams of other PWM pairs are the same as PWM01.

The timer logic module of PWM consists of one 16-bit up-counter ([PCNTR](#)) and three 16-bit parameters ([PWM\\_ENTIRE\\_CYCLE](#), [PWM\\_ACT\\_CYCLE](#), [PWM\\_COUNTER\\_START](#)). The [PWM\\_ENTIRE\\_CYCLE](#) is used to

control the PWM cycle, the [PWM\\_ACT\\_CYCLE](#) is used to control the duty-cycle, the [PWM\\_COUNTER\\_START](#) is used to control the output phase (multi-channel synchronization work requirements).

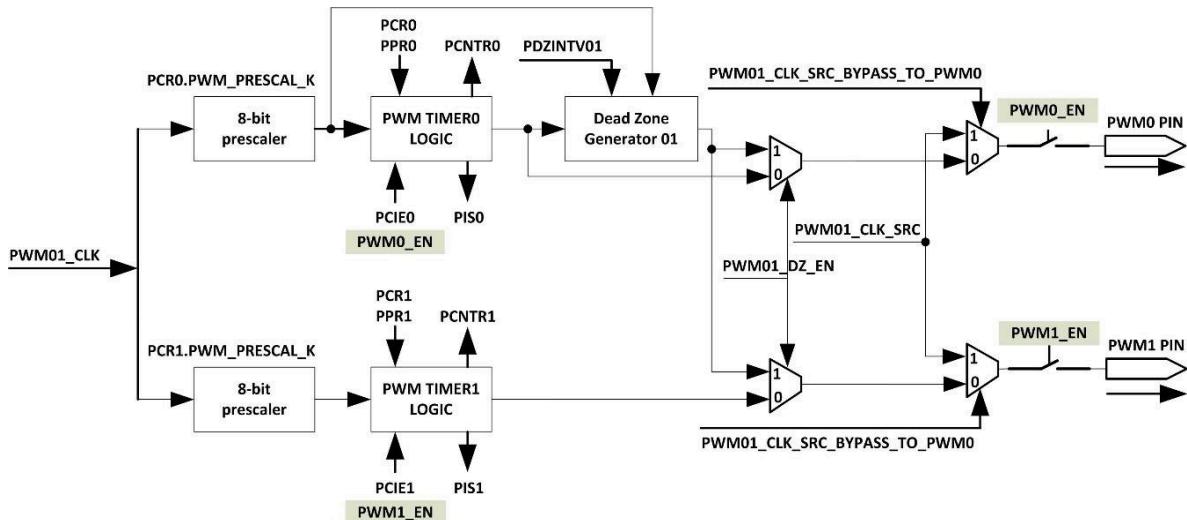
The [PWM\\_ENTIRE\\_CYCLE](#) and the [PWM\\_ACT\\_CYCLE](#) support the cache load, after PWM output is enabled, the register values of the [PWM\\_ENTIRE\\_CYCLE](#) and the [PWM\\_ACT\\_CYCLE](#) can be changed anytime, the changed value caches into the cache register. When the PCNTR counter outputs a period of PWM waveform, the value of the cache register can be updated for the PCNTR control. The purpose of the cache load is to avoid the unstable PWM output waveform with the burred feature when updating the values of the [PWM\\_ENTIRE\\_CYCLE](#) and [PWM\\_ACT\\_CYCLE](#).

The PWM supports cycle and pulse waveform output.

**Cycle mode:** The PWM outputs the setting PWM waveform continually, that is, the output waveform is a continuous PWM square wave.

**Pulse mode:** After setting the [PWM\\_PUL\\_NUM](#) parameter, the PWM outputs (PWM\_PULNUM+1) periods of PWM waveform, that is, the waveform with several pulses are output.

**Figure 8-75 PWM01 Output Logic Module Diagram**



### 8.9.3.5 Up-Counter and Comparator

The period, duty-cycle, and phase of PWM output waveform are decided by the [PCNTR](#), [PWM\\_ENTIRE\\_CYCLE](#), [PWM\\_ACT\\_CYCLE](#), and [PWM\\_COUNTER\\_START](#). The rules are as follows.

- $\text{PCNTR} = (\text{PCNTR} == \text{PWM\_ENTIRE\_CYCLE})?0 : \text{PCNTR} + 1$
- PCNTR starts to count by [PWM\\_COUNTER\\_START](#), the counter of a PWM period is ( $\text{PWM\_ENTIRE\_CYCLE} + 1$ ).
- $\text{PCNTR} > (\text{PWM\_ENTIRE\_CYCLE} - \text{PWM\_ACT\_CYCLE})$ , output “active state”

- PCNTR <= (PWM\_ENTIRE\_CYCLE - PWM\_ACT\_CYCLE), output “~ (active state)”

#### Active state of PWM0 channel is high level (PCRO. PWM\_ACT\_STA = 1)

When PCNTR0 > (PPR0. PWM\_ENTIRE\_CYCLE - PPR0.PWM\_ACT\_CYCLE), then PWM0 outputs 1 (high level).

When PCNTR0 <= (PPR0. PWM\_ENTIRE\_CYCLE - PPR0.PWM\_ACT\_CYCLE), then PWM0 outputs 0 (low level).

The formula of the output period and the duty-cycle for PWM are as follows.

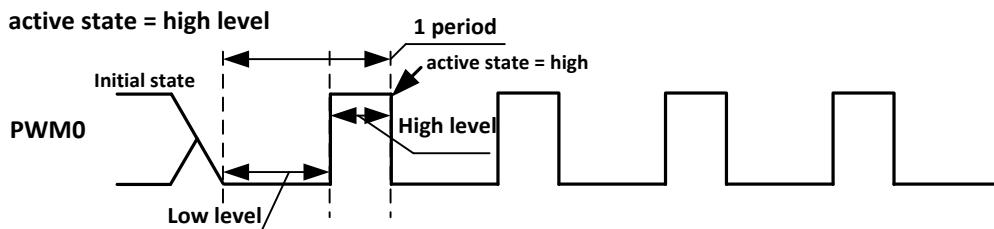
$$T_{\text{period}} = (\text{PWM01\_CLK} / \text{PWM0\_PREScale\_K})^{-1} * (\text{PPR0.PWM\_ENTIRE\_CYCLE} + 1)$$

$$T_{\text{high-level}} = (\text{PWM01\_CLK} / \text{PWM0\_PREScale\_K})^{-1} * \text{PPR0.PWM\_ACT\_CYCLE}$$

$$T_{\text{low-level}} = (\text{PWM01\_CLK} / \text{PWM0\_PREScale\_K})^{-1} * (\text{PPR0.PWM\_ENTIRE\_CYCLE} + 1 - \text{PPR0.PWM\_ACT\_CYCLE})$$

$$\text{Duty-cycle} = (\text{high level time}) / (\text{1 period time}) = T_{\text{high-level}} / T_{\text{period}}$$

**Figure 8-76 PWM0 High Level Active State**



#### Active state of PWM0 channel is low level (PCRO. PWM\_ACT\_STA = 0)

When PCNTR0 > (PPR0.PWM\_ENTIRE\_CYCLE - PPR0.PWM\_ACT\_CYCLE), then PWM0 outputs 0.

When PCNTR0 <= (PPR0.PWM\_ENTIRE\_CYCLE - PPR0.PWM\_ACT\_CYCLE), then PWM0 outputs 1.

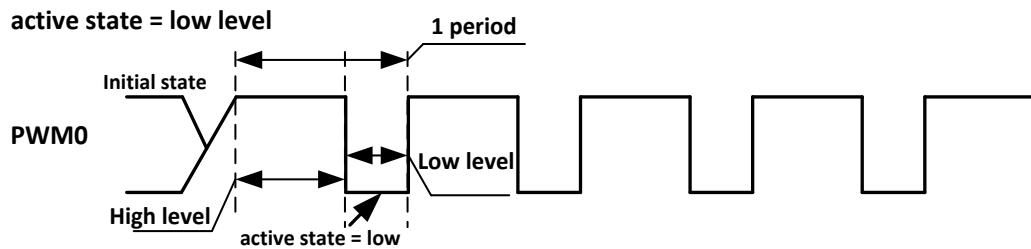
The formula of the output period and the duty-cycle for PWM are as follows.

$$T_{\text{period}} = (\text{PWM01\_CLK} / \text{PWM0\_PREScale\_K})^{-1} * (\text{PPR0.PWM\_ENTIRE\_CYCLE} + 1)$$

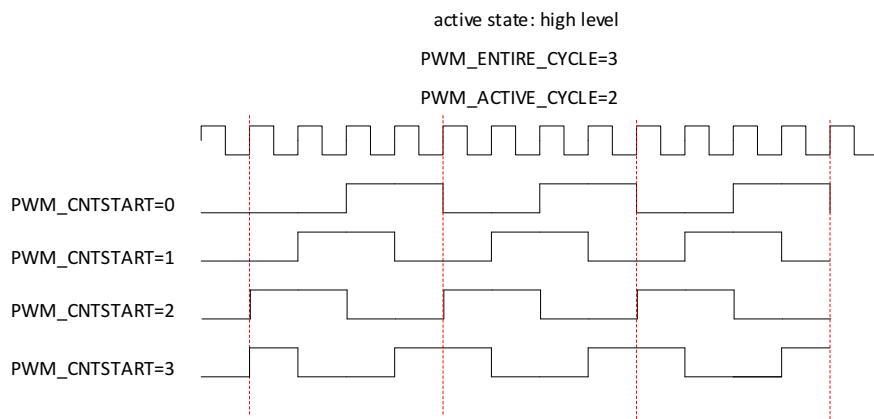
$$T_{\text{high-level}} = (\text{PWM01\_CLK} / \text{PWM0\_PREScale\_K})^{-1} * (\text{PPR0.PWM\_ENTIRE\_CYCLE} + 1 - \text{PPR0.PWM\_ACT\_CYCLE})$$

$$T_{\text{low-level}} = (\text{PWM01\_CLK} / \text{PWM0\_PREScale\_K})^{-1} * \text{PPR0.PWM\_ACT\_CYCLE}$$

$$\text{Duty-cycle} = (\text{low level time}) / (\text{1 period time}) = T_{\text{low-level}} / T_{\text{period}}$$

**Figure 8-77 PWM0 Low Level Active State**

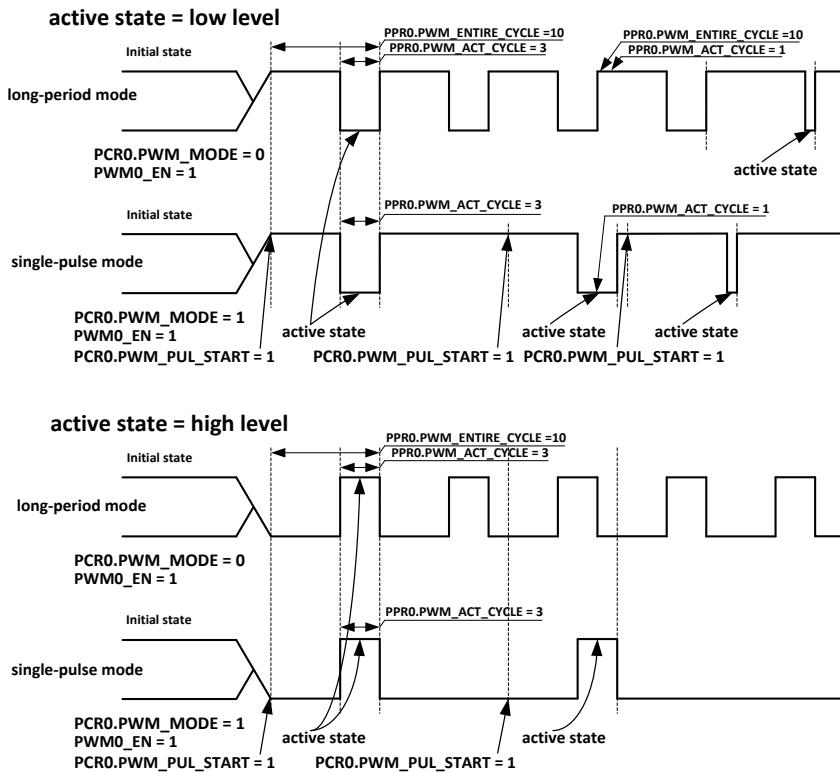
The counter of PCNTR starts from 0 by default, it can output the pulse control of the waveform by setting [PWM\\_COUNTER\\_START](#). The figure is as follows.

**Figure 9-78 Phase of PWM0 High Level Active State**

#### 8.9.3.6 Pulse Mode and Cycle Mode

The PWM output supports pulse mode and cycle mode. PWM in pulse mode outputs one pulse waveform, but PWM in cycle mode outputs continuous waveform. Figure 9-89 shows the PWM output waveform in pulse mode and cycle mode.

Figure 8-79 PWM0 Output Waveform in Pulse Mode and Cycle Mode



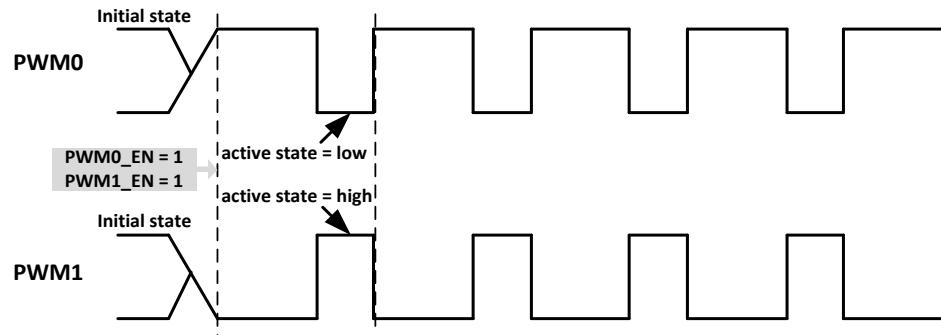
Each channel of the PWM module supports the PWM output of pulse mode and cycle mode, the active state of the PWM output waveform can be programmed to control.

When [PCR0\[PWM\\_MODE\]](#) is 0, the PWM0 outputs in cycle mode. When [PCR0\[PWM\\_MODE\]](#) is 1, the PWM0 outputs in pulse mode.

Specifically, in pulse mode, after the PWM0 channel enabled, [PCR0\[PWM\\_PUL\\_START\]](#) needs to be set to 1 when the PWM0 needs to output pulse waveform, after completed the output, [PCR0\[PWM\\_PUL\\_START\]](#) can be cleared to 0 by hardware. The next setting 1 can be operated after [PCR0\[PWM\\_PUL\\_START\]](#) is cleared.

#### 8.9.3.7 Complementary Pair Output

Every PWM pair supports complementary pair output and PWM pair with dead-time. Figure 9-90 shows the complementary pair output of PWM01.

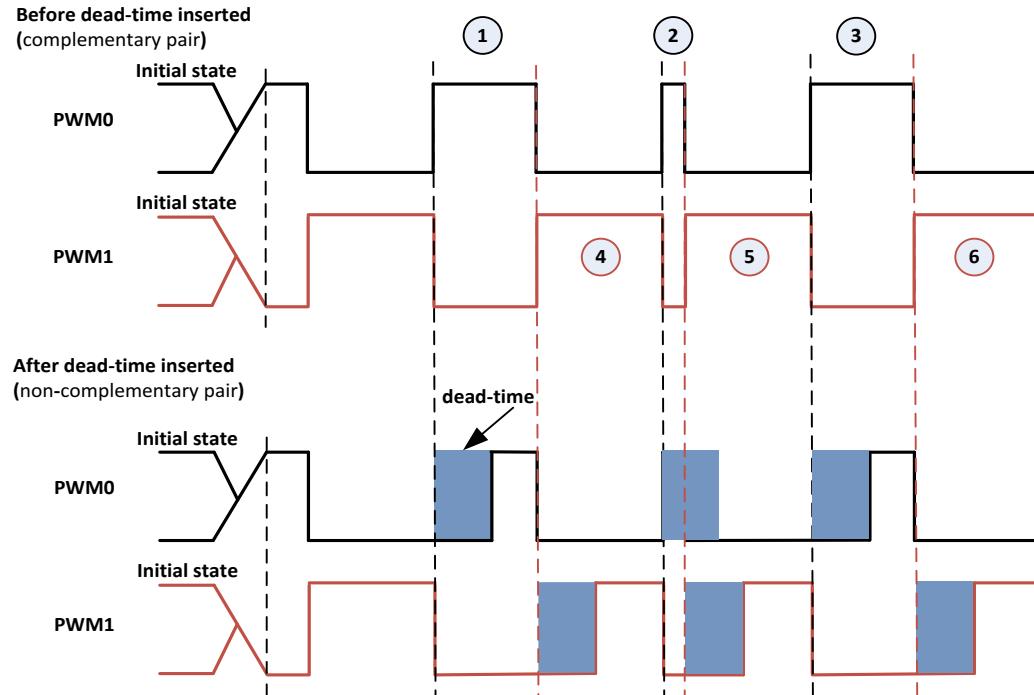
**Figure 8-80 PWM01 Complementary Pair Output**

The complementary pair output needs to satisfy the following conditions:

- PWM0 and PWM1 have the same clock divider, frequency, duty-cycle, and phase
- PWM0 and PWM1 have an opposite active state
- Enable the clock gating of PWM0 and PWM1 at the same time
- Enable the waveform output of PWM0 and PWM1 at the same time

#### 8.9.3.8 Dead-time Generator

Every PWM pair has a programmable dead-time generator. When the dead-time function of the PWM pair enabled, the PWM01 output waveform is decided by PWM timer logic and DeadZone Generator. Figure 9-91 shows the output waveform.

**Figure 8-81 Dead-time Output Waveform**

The PWM waveform before the insertion of dead-time indicates a complementary waveform pair of non-inserted dead-time in Dead Zone Generator 01.

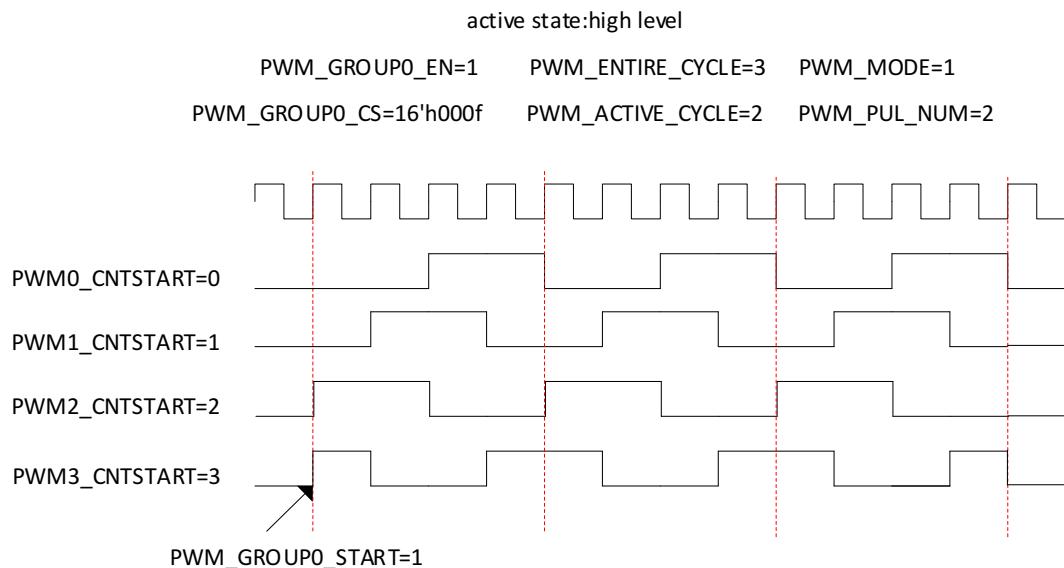
The PWM waveform after the insertion of dead-time indicates a non-complementary PWM waveform pair inserted dead-time in a complementary waveform pair of Dead Zone Generator 01. The PWM waveform pair at last outputs to PWM0 pin and PWM1 pin.

For the complementary pair of Dead Zone Generator 01, the principle of inserting dead-time is that to insert dead-time as soon as the rising edge came. If the high level time for mark② in the above figure is less than dead-time, then dead-time will override the high level. The setting of dead-time needs to consider the period and the duty-cycle of the output waveform. The dead-time formula is defined as follows:

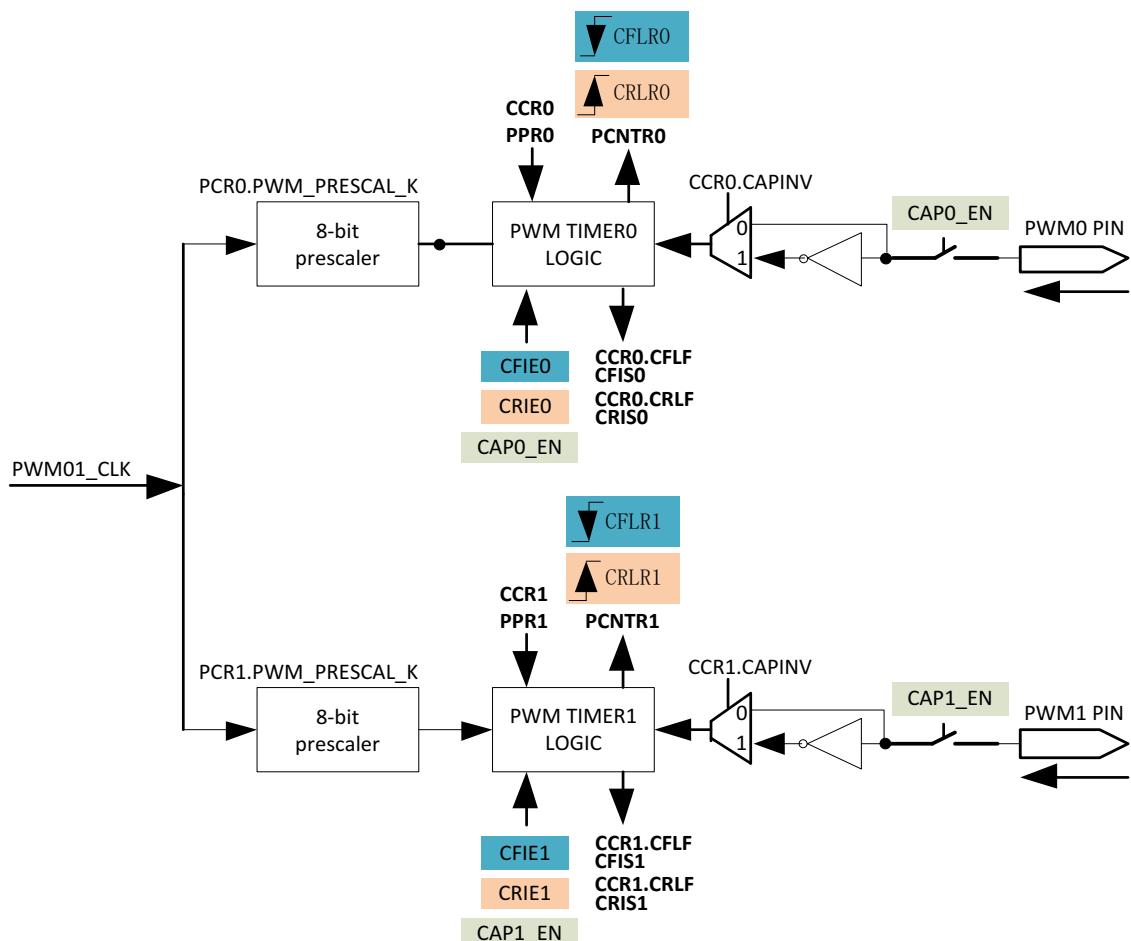
$$\text{Dead-time} = (\text{PWM01_CLK} / \text{PWM0_PRESCALE_K})^{-1} * \text{PDZINTV01}$$

#### 8.9.3.9 PWM Group Mode

Taking PWM Group0 as an example. The same group of PWM channel is selected to work by PGRO.CS; the same [PWM\\_ENTIRE\\_CYCLE](#), [PWM\\_ACT\\_CYCLE](#) are set by the same clock configuration; the different [PWM\\_COUNTER\\_START](#) can output PWM group signals with the same duty-cycle and the different phase.

**Figure 8-82 Group 0~3 PWM Signal Output**

### 8.9.3.10 Capture Input

**Figure 8-83 PWM01 Capture Logic Module Diagram**

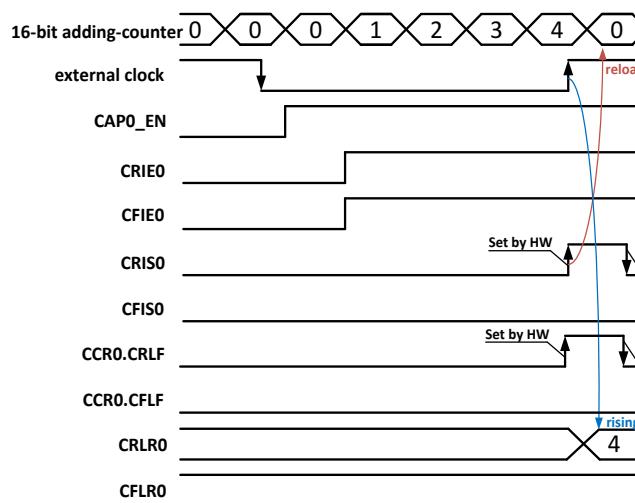
Besides the timer logic module of every PWM channel generates PWM output, it can be used to capture the rising edge and the falling edge of the external clock. Using the PWM0 channel as an example, the PWM0 channel has one [CFLR0](#) and one [CRLR0](#) for capturing up-counter value on the falling edge and rising edge, respectively. You can calculate the period of the external clock by [CFLR0](#) and [CRLR0](#).

$$T_{\text{high-level}} = (\text{PWM01\_CLK} / \text{PWMO\_PRESCALE\_K})^{-1} * \text{CRLR0}$$

$$T_{\text{low-level}} = (\text{PWM01\_CLK} / \text{PWMO\_PRESCALE\_K})^{-1} * \text{CFLR0}$$

$$T_{\text{period}} = T_{\text{high-level}} + T_{\text{low-level}}$$

**Figure 8-84 PWM0 Channel Capture Timing**



When the capture input function of the PWM0 channel is enabled, the PCNTR of the PWM0 channel starts to work.

When the timer logic module of PWM0 captures a rising edge, the current value of the up-counter is locked to [CRLR0](#) and [CCR0\[CRLF\]](#) is set to 1. If [CRIEO](#) is 1, then [CRISO](#) is set to 1, the PWM0 channel sends interrupt requests, and the up-counter is loaded to 0 and continues to count. If [CRIEO](#) is 0, the timer logic module of PWM0 captures a rising edge, [CRISO](#) cannot be set to 1, the up-counter is not loaded to 0.

When the timer logic module of PWM0 captures one falling edge, the current value of PCNTR is locked to [CFLR0](#) and [CCR0\[CFLF\]](#) is set to 1. If [CFIEO](#) is 1, then [CFISO](#) is set to 1, the PWM0 channel sends interrupt requests, and the up-counter is loaded to 0 and continues to count. If [CFIEO](#) is 0, the timer logic module of PWM0 captures a falling edge, [CFISO](#) cannot be set to 1, the up-counter is not loaded to 0.

### 8.9.3.11 Interrupt

The PWM supports an interrupt generation when configuring the PWM channel to PWM output or capturing input.

---

For PWM output function, when the controller outputs one period of PWM waveform in cycle mode, the PIS of the corresponding PWM channel is set to 1; when the controller outputs (PWM\_PULNUM+1) periods of PWM waveform in pulse mode, the PIS of the corresponding PWM channel is set to 1.

---

**NOTE**

The PIS bit is set to 1 automatically by hardware and cleared by software.

---

For capturing input function, when the timer logic module of the capture channel0 captures rising edge, and CRIEQ is 1, then CRISO is set to 1; when the timer logic module of the capture channel0 captures falling edge, and CFIEQ is 1, then CFISO is set to 1.

## 8.9.4 Programming Guidelines

The following working mode takes PWM01 as an example, other PWM pairs and PWM01 are consistent.

### 8.9.4.1 Configuring Clock

- Step 1** PWM gating: When using PWM, write 1 to PCGR[PWMx\_CLK\_GATING].
- Step 2** PWM clock source select: Set PCCR01[PWM01\_CLK\_SRC] to select HOSC or APB0 clock.
- Step 3** PWM clock divider: Set PCCR01[PWM01\_CLK\_DIV\_M] to select different frequency division coefficient (1/2/4/8/16/32/64/128/256).
- Step 4** PWM clock bypass: Set PCGR[PWM\_CLK\_SRC\_BYPASS\_TO\_PWM] to 1, output the PWM clock after the secondary frequency division to the corresponding PWM output pin.
- Step 5** PWM internal clock configuration: Set PCR[PWM\_PRESCAL\_K] to select any frequency division coefficient from 1 to 256.

**NOTE**

For the channel of complementary output and group mode, firstly, set the same clock configurations (clock source selects APB0, clock division configures the same division factor); secondly, open clock gating at the same time; thirdly, configure PWM parameters; finally, enable PWM output at the same time to ensure each channel sync.

We suggest that the two channels of the same PWM pair cannot subject to two groups because of they have the same first level clock division and gating. If must allocate based on this way, the first level of clock division of the channel used by all groups needs to set to the same coefficient and open gating at the same time. And the total module needs to be reset when the group mode regroups.

---

#### 8.9.4.2 Configuring PWM

- Step 1** PWM mode: Set [PCR\[PWM\\_MODE\]](#) to select cycle mode or pulse mode, if pulse mode, [PCR\[PWM\\_PUL\\_NUM\]](#) needs to be configured.
- Step 2** PWM active level: Set [PCR\[PWM\\_ACT\\_STA\]](#) to select a low level or high level.
- Step 3** PWM duty-cycle: Configure [PPR\[PWM\\_ENTIRE\\_CYCLE\]](#) and [PPR\[PWM\\_ACT\\_CYCLE\]](#) after clock gating is opened.
- Step 4** PWM starting/stopping phase: Configure [PCNTR\[PWM\\_COUNTER\\_START\]](#) after the clock gating is enabled and before the PWM is enabled. You can verify whether the configuration was successful by reading back [PCNTR\[PWM\\_COUNTER\\_STATUS\]](#).
- Step 5** Enable PWM: Configure PER to select the corresponding PWM enable bit; when selecting pulse mode, [PCR\[PWM\\_PUL\\_START\]](#) needs to be enabled.

#### 8.9.4.3 Configuring Deadzone

- Step 1** Set initial value: set [PWM01\\_Dead\\_Zone - Hlk49450444\\_PWM01\\_Dead\\_Zone - Hlk49450444](#)[PDZINTV01].
- Step 2** Enable Deadzone: set [PWM01\\_Dead\\_Zone - Hlk49450444\\_PWM01\\_Dead\\_Zone - Hlk49450444](#)[PWM01\_DZ\_CN].

#### 8.9.4.4 Configuring Capture Input

- Step 1** Enable capture: Configure [CCR](#) to enable the corresponding channel.
- Step 2** Capture mode: Configure [CCR\[CRLF\]](#) and [CCR\[CFLF\]](#) to select rising edge capture or falling edge capture, configure [CCR\[CAPINV\]](#) to select whether the input signal does reverse processing.

#### 8.9.5 Register List

Module Name	Base Address
PWM	0x02000C00

Register Name	Offset	Description
PIER	0x0000	PWM IRQ Enable Register
PISR	0x0004	PWM IRQ Status Register

<b>Register Name</b>	<b>Offset</b>	<b>Description</b>
CIER	0x0010	Capture IRQ Enable Register
CISR	0x0014	Capture IRQ Status Register
PCCR01	0x0020	PWM01 Clock Configuration Register
PCCR23	0x0024	PWM23 Clock Configuration Register
PCCR45	0x0028	PWM45 Clock Configuration Register
PCCR67	0x002C	PWM67 Clock Configuration Register
PCGR	0x0040	PWM Clock Gating Register
PDZCR01	0x0060	PWM01 Dead Zone Control Register
PDZCR23	0x0064	PWM23 Dead Zone Control Register
PDZCR45	0x0068	PWM45 Dead Zone Control Register
PDZCR67	0x006C	PWM67 Dead Zone Control Register
PER	0x0080	PWM Enable Register
PGR0	0x0090	PWM Group0 Register
PGR1	0x0094	PWM Group1 Register
PGR2	0x0098	PWM Group2 Register
PGR3	0x009C	PWM Group3 Register
CER	0x00C0	Capture Enable Register
PCR	0x0100+0x0000+N*0x0020 (N= 0~7)	PWM Control Register
PPR	0x0100+0x0004+N*0x0020 (N= 0~7)	PWM Period Register
PCNTR	0x0100+0x0008+N*0x0020 (N= 0~7)	PWM Count Register
PPCNTR	0x0100+0x000C+N*0x0020 (N= 0~7)	PWM Pulse Count Register
CCR	0x0100+0x0010+N*0x0020 (N= 0~7)	Capture Control Register
CRLR	0x0100+0x0014+N*0x0020 (N= 0~7)	Capture Rise Lock Register
CFLR	0x0100+0x0018+N*0x0020 (N= 0~7)	Capture Fall Lock Register

## 8.9.6 Register Description

### 8.9.6.1 0x0000 PWM IRQ Enable Register (Default Value: 0x0000\_0000)

Offset:0x0000			Register Name: PIER
Bit	Read/Write	Default/Hex	Description
31:20	/	/	/
19	R/W	0x0	PGIE3 PWM Group 3 Interrupt Enable 0: Disable 1: Enable
18	R/W	0x0	PGIE2 PWM Group 2 Interrupt Enable 0: Disable 1: Enable
17	R/W	0x0	PGIE1 PWM Group 1 Interrupt Enable 0: Disable 1: Enable
16	R/W	0x0	PGIE0 PWM Group 0 Interrupt Enable 0: Disable 1: Enable
15:8	/	/	/
7	R/W	0x0	PCIE7 PWM Channel 7 Interrupt Enable 0: PWM Channel 7 Interrupt Disable 1: PWM Channel 7 Interrupt Enable
6	R/W	0x0	PCIE6 PWM Channel 6 Interrupt Enable 0: PWM Channel 6 Interrupt Disable 1: PWM Channel 6 Interrupt Enable
5	R/W	0x0	PCIE5 PWM Channel 5 Interrupt Enable 0: PWM Channel 5 Interrupt Disable 1: PWM Channel 5 Interrupt Enable

Offset:0x0000			Register Name: PIER
Bit	Read/Write	Default/Hex	Description
4	R/W	0x0	PCIE4 PWM Channel 4 Interrupt Enable 0: PWM Channel 4 Interrupt Disable 1: PWM Channel 4 Interrupt Enable
3	R/W	0x0	PCIE3 PWM Channel 3 Interrupt Enable 0: PWM Channel 3 Interrupt Disable 1: PWM Channel 3 Interrupt Enable
2	R/W	0x0	PCIE2 PWM Channel 2 Interrupt Enable 0: PWM Channel 2 Interrupt Disable 1: PWM Channel 2 Interrupt Enable
1	R/W	0x0	PCIE1 PWM Channel 1 Interrupt Enable 0: PWM Channel 1 Interrupt Disable 1: PWM Channel 1 Interrupt Enable
0	R/W	0x0	PCIE0 PWM Channel 0 Interrupt Enable 0: PWM Channel 0 Interrupt Disable 1: PWM Channel 0 Interrupt Enable

#### 8.9.6.2 0x0004 PWM IRQ Status Register (Default Value: 0x0000\_0000)

Offset:0x0004			Register Name: PISR
Bit	Read/Write	Default/Hex	Description
31:20	/	/	/
19	R/W1C	0x0	PGIS3 PWM Group 3 Interrupt Status
18	R/W1C	0x0	PGIS2 PWM Group 2 Interrupt Status
17	R/W1C	0x0	PGIS1 PWM Group 1 Interrupt Status
16	R/W1C	0x0	PGIS0 PWM Group 0 Interrupt Status

Offset:0x0004			Register Name: PISR
Bit	Read/Write	Default/Hex	Description
15:8	/	/	/
7	R/W1C	0x0	<p>PIS7 PWM Channel 7 Interrupt Status When the PWM channel 7 counter reaches the Entire Cycle Value, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: PWM channel 7 interrupt is not pending. Reads 1: PWM channel 7 interrupt is pending. Writes 0: No effect. Writes 1: Clear PWM channel 7 interrupt status.</p>
6	R/W1C	0x0	<p>PIS6 PWM Channel 6 Interrupt Status When the PWM channel 6 counter reaches the Entire Cycle Value, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: PWM channel 6 interrupt is not pending. Reads 1: PWM channel 6 interrupt is pending. Writes 0: No effect. Writes 1: Clear PWM channel 6 interrupt status.</p>
5	R/W1C	0x0	<p>PIS5 PWM Channel 5 Interrupt Status When the PWM channel 5 counter reaches the Entire Cycle Value, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: PWM channel 5 interrupt is not pending. Reads 1: PWM channel 5 interrupt is pending. Writes 0: No effect. Writes 1: Clear PWM channel 5 interrupt status.</p>
4	R/W1C	0x0	<p>PIS4 PWM Channel 4 Interrupt Status When the PWM channel 4 counter reaches the Entire Cycle Value, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: PWM channel 4 interrupt is not pending. Reads 1: PWM channel 4 interrupt is pending. Writes 0: No effect. Writes 1: Clear PWM channel 4 interrupt status.</p>

Offset:0x0004			Register Name: PISR
Bit	Read/Write	Default/Hex	Description
3	R/W1C	0x0	<p>PIS3 PWM Channel 3 Interrupt Status When the PWM channel 3 counter reaches the Entire Cycle Value, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: PWM channel 3 interrupt is not pending. Reads 1: PWM channel 3 interrupt is pending. Writes 0: No effect. Writes 1: Clear PWM channel 3 interrupt status.</p>
2	R/W1C	0x0	<p>PIS2 PWM Channel 2 Interrupt Status When the PWM channel 2 counter reaches the Entire Cycle Value, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: PWM channel 2 interrupt is not pending. Reads 1: PWM channel 2 interrupt is pending. Writes 0: No effect. Writes 1: Clear PWM channel 2 interrupt status.</p>
1	R/W1C	0x0	<p>PIS1 PWM Channel 1 Interrupt Status When the PWM channel 1 counter reaches the Entire Cycle Value, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: PWM channel 1 interrupt is not pending. Reads 1: PWM channel 1 interrupt is pending. Writes 0: No effect. Writes 1: Clear PWM channel 1 interrupt status.</p>
0	R/W1C	0x0	<p>PISO PWM Channel 0 Interrupt Status When the PWM channel 0 counter reaches the Entire Cycle Value, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: PWM channel 0 interrupt is not pending. Reads 1: PWM channel 0 interrupt is pending. Writes 0: No effect. Writes 1: Clear PWM channel 0 interrupt status.</p>

## 8.9.6.3 0x0010 PWM Capture IRQ Enable Register (Default Value: 0x0000\_0000)

Offset:0x0010			Register Name: CIER
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15	R/W	0x0	<p>CFIE7</p> <p>If the enable bit is set to 1, when the capture channel 7 captures falling edge, it generates a capture channel 7 pending.</p> <p>0: Capture channel 7 fall lock interrupt disable 1: Capture channel 7 fall lock interrupt enable</p>
14	R/W	0x0	<p>CRIE7</p> <p>If the enable bit is set to 1, when the capture channel 7 captures rising edge, it generates a capture channel 7 pending.</p> <p>0: Capture channel 7 rise lock interrupt disable 1: Capture channel 7 rise lock interrupt enable</p>
13	R/W	0x0	<p>CFIE6</p> <p>If the enable bit is set to 1, when the capture channel 6 captures falling edge, it generates a capture channel 6 pending.</p> <p>0: Capture channel 6 fall lock interrupt disable 1: Capture channel 6 fall lock interrupt enable</p>
12	R/W	0x0	<p>CRIE6</p> <p>If the enable bit is set to 1, when the capture channel 6 captures rising edge, it generates a capture channel 6 pending.</p> <p>0: Capture channel 6 rise lock interrupt disable 1: Capture channel 6 rise lock interrupt enable</p>
11	R/W	0x0	<p>CFIE5</p> <p>If the enable bit is set to 1, when the capture channel 5 captures falling edge, it generates a capture channel 5 pending.</p> <p>0: Capture channel 5 fall lock interrupt disable 1: Capture channel 5 fall lock interrupt enable</p>
10	R/W	0x0	<p>CRIE5</p> <p>If the enable bit is set to 1, when the capture channel 5 captures rising edge, it generates a capture channel 5 pending.</p> <p>0: Capture channel 5 rise lock interrupt disable 1: Capture channel 5 rise lock interrupt enable</p>

Offset:0x0010			Register Name: CIER
Bit	Read/Write	Default/Hex	Description
9	R/W	0x0	<p>CFIE4</p> <p>If the enable bit is set to 1, when the capture channel 4 captures falling edge, it generates a capture channel 4 pending.</p> <p>0: Capture channel 4 fall lock interrupt disable 1: Capture channel 4 fall lock interrupt enable</p>
8	R/W	0x0	<p>CRIE4</p> <p>If the enable bit is set to 1, when the capture channel 4 captures rising edge, it generates a capture channel 4 pending.</p> <p>0: Capture channel 4 rise lock interrupt disable 1: Capture channel 4 rise lock interrupt enable</p>
7	R/W	0x0	<p>CFIE3</p> <p>If the enable bit is set to 1, when the capture channel 3 captures falling edge, it generates a capture channel 3 pending.</p> <p>0: Capture channel 3 fall lock interrupt disable 1: Capture channel 3 fall lock interrupt enable</p>
6	R/W	0x0	<p>CRIE3</p> <p>If the enable bit is set to 1, when the capture channel 3 captures rising edge, it generates a capture channel 3 pending.</p> <p>0: Capture channel 3 rise lock interrupt disable 1: Capture channel 3 rise lock interrupt enable</p>
5	R/W	0x0	<p>CFIE2</p> <p>If the enable bit is set to 1, when the capture channel 2 captures falling edge, it generates a capture channel 2 pending.</p> <p>0: Capture channel 2 fall lock interrupt disable 1: Capture channel 2 fall lock interrupt enable</p>
4	R/W	0x0	<p>CRIE2</p> <p>If the enable bit is set to 1, when the capture channel 2 captures rising edge, it generates a capture channel 2 pending.</p> <p>0: Capture channel 2 rise lock interrupt disable 1: Capture channel 2 rise lock interrupt enable</p>
3	R/W	0x0	<p>CFIE1</p> <p>If the enable bit is set to 1, when the capture channel 1 captures falling edge, it generates a capture channel 1 pending.</p> <p>0: Capture channel 1 fall lock interrupt disable 1: Capture channel 1 fall lock interrupt enable</p>

Offset:0x0010			Register Name: CIER
Bit	Read/Write	Default/Hex	Description
2	R/W	0x0	<p>CRIE1</p> <p>If the enable bit is set to 1, when the capture channel 1 captures rising edge, it generates a capture channel 1 pending.</p> <p>0: Capture channel 1 rise lock interrupt disable 1: Capture channel 1 rise lock interrupt enable</p>
1	R/W	0x0	<p>CFIE0</p> <p>If the enable bit is set to 1, when the capture channel 0 captures falling edge, it generates a capture channel 0 pending.</p> <p>0: Capture channel 0 fall lock interrupt disable 1: Capture channel 0 fall lock interrupt enable</p>
0	R/W	0x0	<p>CRIE0</p> <p>If the enable bit is set to 1, when the capture channel 0 captures rising edge, it generates a capture channel 0 pending.</p> <p>0: Capture channel 0 rise lock interrupt disable 1: Capture channel 0 rise lock interrupt enable</p>

#### 8.9.6.4 0x0014 PWM Capture IRQ Status Register (Default Value: 0x0000\_0000)

Offset:0x0014			Register Name: CISR
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
15	R/W1C	0x0	<p>CFIS7</p> <p>Status of the capture channel 7 falling lock interrupt</p> <p>When the capture channel 7 captures falling edge, if the fall lock interrupt (<a href="#">CFIE7</a>) is enabled, this bit is set to 1 by hardware.</p> <p>Writing 1 to clear this bit.</p> <p>Reads 0: The capture channel 7 interrupt is not pending.</p> <p>Reads 1: The capture channel 7 interrupt is pending.</p> <p>Writes 0: No effect.</p> <p>Writes 1: Clear the status of the capture channel 7 interrupt.</p>

Offset:0x0014			Register Name: CISR
Bit	Read/Write	Default/Hex	Description
14	R/W1C	0x0	<p>CRIS7</p> <p>Status of the capture channel 7 rising lock interrupt</p> <p>When the capture channel 7 captures rising edge, if the rise lock interrupt (<a href="#">CRIE7</a>) is enabled, this bit is set to 1 by hardware. Write 1 to clear this bit.</p> <p>Reads 0: The capture channel 7 interrupt is not pending.</p> <p>Reads 1: The capture channel 7 interrupt is pending.</p> <p>Writes 0: No effect.</p> <p>Writes 1: Clear the status of the capture channel 7 interrupt.</p>
13	R/W1C	0x0	<p>CFIS6</p> <p>Status of the capture channel 6 falling lock interrupt</p> <p>When the capture channel 6 captures falling edge, if the fall lock interrupt (<a href="#">CFIE6</a>) is enabled, this bit is set to 1 by hardware. Writing 1 to clear this bit.</p> <p>Reads 0: The capture channel 6 interrupt is not pending.</p> <p>Reads 1: The capture channel 6 interrupt is pending.</p> <p>Writes 0: No effect.</p> <p>Writes 1: Clear the status of the capture channel 6 interrupt.</p>
12	R/W1C	0x0	<p>CRIS6</p> <p>Status of the capture channel 6 rising lock interrupt.</p> <p>When the capture channel 6 captures rising edge, if the rise lock interrupt (<a href="#">CRIE6</a>) is enabled, this bit is set to 1 by hardware. Writing 1 to clear this bit.</p> <p>Reads 0: The capture channel 6 interrupt is not pending.</p> <p>Reads 1: The capture channel 6 interrupt is pending.</p> <p>Writes 0: No effect.</p> <p>Writes 1: Clear the status of the capture channel 6 interrupt.</p>
11	R/W1C	0x0	<p>CFIS5</p> <p>Status of the capture channel 5 falling lock interrupt</p> <p>When the capturing channel 5 captures falling edge, if the fall lock interrupt (<a href="#">CFIE5</a>) is enabled, this bit is set to 1 by hardware. Writing 1 to clear this bit.</p> <p>Reads 0: The capture channel 5 interrupt is not pending.</p> <p>Reads 1: The capture channel 5 interrupt is pending.</p> <p>Writes 0: No effect.</p> <p>Reads 1: Clear the status of the capture channel 5 interrupt.</p>

Offset:0x0014			Register Name: CISR
Bit	Read/Write	Default/Hex	Description
10	R/W1C	0x0	<p>CRIS5</p> <p>Status of the capture channel 5 rising lock interrupt</p> <p>When the capture channel 5 captures rising edge, if the rise lock interrupt (<a href="#">CRIES5</a>) is enabled, this bit is set to 1 by hardware.</p> <p>Writing 1 to clear this bit.</p> <p>Reads 0: The capture channel 5 interrupt is not pending.</p> <p>Reads 1: The capture channel 5 interrupt is pending.</p> <p>Writes 0: No effect.</p> <p>Writes 1: Clear the status of the capture channel 5 interrupt.</p>
9	R/W1C	0x0	<p>CFIS4</p> <p>Status of the capture channel 4 falling lock interrupt</p> <p>When the capture channel 4 captures falling edge, if the fall lock interrupt (<a href="#">CFIE4</a>) is enabled, this bit is set to 1 by hardware.</p> <p>Writing 1 to clear this bit.</p> <p>Reads 0: The capture channel 4 interrupt is not pending.</p> <p>Reads 1: The capture channel 4 interrupt is pending.</p> <p>Writes 0: No effect.</p> <p>Writes 1: Clear the status of the capture channel 4 interrupt.</p>
8	R/W1C	0x0	<p>CRIS4</p> <p>Status of the capture channel 4 rising lock interrupt.</p> <p>When the capture channel 4 captures rising edge, if the rise lock interrupt (<a href="#">CRIES4</a>) is enabled, this bit is set to 1 by hardware.</p> <p>Writing 1 to clear this bit.</p> <p>Reads 0: The capture channel 4 interrupt is not pending.</p> <p>Reads 1: The capture channel 4 interrupt is pending.</p> <p>Writes 0: No effect.</p> <p>Writes 1: Clear the status of the capture channel 4 interrupt status.</p>
7	R/W1C	0x0	<p>CFIS3</p> <p>Status of the capture channel 3 falling lock interrupt.</p> <p>When the capture channel 3 captures falling edge, if the fall lock interrupt (<a href="#">CFIE3</a>) is enabled, this bit is set to 1 by hardware.</p> <p>Writing 1 to clear this bit.</p> <p>Reads 0: The capture channel 3 interrupt is not pending.</p> <p>Reads 1: The capture channel 3 interrupt is pending.</p> <p>Writes 0: no effect.</p> <p>Writes 1: Clear the status of the capture channel 3 interrupt.</p>

Offset:0x0014			Register Name: CISR
Bit	Read/Write	Default/Hex	Description
6	R/W1C	0x0	<p>CRIS3</p> <p>Status of the capture channel 3 rising lock interrupt</p> <p>When the capture channel 3 captures rising edge, if the rise lock interrupt (<a href="#">CRIE3</a>) is enabled, this bit is set to 1 by hardware.</p> <p>Writing 1 to clear this bit.</p> <p>Reads 0: The capture channel 3 interrupt is not pending.</p> <p>Reads 1: The capture channel 3 interrupt is pending.</p> <p>Writes 0: no effect.</p> <p>Writes 1: Clear the status of the capture channel 3 interrupt.</p>
5	R/W1C	0x0	<p>CFIS2</p> <p>Status of the capture channel 2 falling lock interrupt</p> <p>When the capture channel 2 captures falling edge, if the fall lock interrupt (<a href="#">CFIE2</a>) is enabled, this bit is set to 1 by hardware.</p> <p>Writing 1 to clear this bit.</p> <p>Reads 0: The capture channel 2 interrupt is not pending.</p> <p>Reads 1: The capture channel 2 interrupt is pending.</p> <p>Writes 0: no effect.</p> <p>Writes 1: Clear the status of the capture channel 2 interrupt.</p>
4	R/W1C	0x0	<p>CRIS2</p> <p>Status of the capture channel 2 rising lock interrupt.</p> <p>When the capture channel 2 captures rising edge, if the rise lock interrupt (<a href="#">CRIE2</a>) is enabled, this bit is set to 1 by hardware.</p> <p>Writing 1 to clear this bit.</p> <p>Reads 0: The capture channel 2 interrupt is not pending.</p> <p>Reads 1: The capture channel 2 interrupt is pending.</p> <p>Writes 0: no effect.</p> <p>Writes 1: Clear the status of the capture channel 2 interrupt.</p>
3	R/W1C	0x0	<p>CFIS1</p> <p>Status of the capture channel 1 falling lock interrupt</p> <p>When the capture channel 1 captures falling edge, if the fall lock interrupt (<a href="#">CFIE1</a>) is enabled, this bit is set to 1 by hardware.</p> <p>Writing 1 to clear this bit.</p> <p>Reads 0: The capture channel 1 interrupt is not pending.</p> <p>Reads 1: The capture channel 1 interrupt is pending.</p> <p>Writes 0: no effect.</p> <p>Writes 1: Clear the status of the capture channel 1 interrupt.</p>

Offset:0x0014			Register Name: CISR
Bit	Read/Write	Default/Hex	Description
2	R/W1C	0x0	<p>CRIS1</p> <p>Status of the capture channel 1 rising lock interrupt.</p> <p>When the capture channel 1 captures rising edge, if the rise lock interrupt (<a href="#">CRIE1</a>) is enabled, this bit is set to 1 by hardware.</p> <p>Writing 1 to clear this bit.</p> <p>Reads 0: The capture channel 1 interrupt is not pending.</p> <p>Reads 1: The capture channel 1 interrupt is pending.</p> <p>Writes 0: no effect.</p> <p>Writes 1: Clear the status of the capture channel 1 interrupt.</p>
1	R/W1C	0x0	<p>CFISO</p> <p>Status of the capture channel 0 falling lock interrupt</p> <p>When the capture channel 0 captures falling edge, if the fall lock interrupt (<a href="#">CFIE0</a>) is enabled, this bit is set to 1 by hardware.</p> <p>Writing 1 to clear this bit.</p> <p>Reads 0: The capture channel 0 interrupt is not pending.</p> <p>Reads 1: The capture channel 0 interrupt is pending.</p> <p>Writes 0: no effect.</p> <p>Writes 1: Clear the status of the capture channel 0 interrupt.</p>
0	R/W1C	0x0	<p>CRISO</p> <p>Status of the capture channel 0 rising lock interrupt</p> <p>When the capture channel 0 captures rising edge, if the rise lock interrupt (<a href="#">CRIE0</a>) is enabled, this bit is set 1 by hardware.</p> <p>Writing 1 to clear this bit.</p> <p>Reads 0: The capture channel 0 interrupt is not pending.</p> <p>Reads 1: The capture channel 0 interrupt is pending.</p> <p>Writes 0: no effect.</p> <p>Writes 1: Clear the status of the capture channel 0 interrupt.</p>

#### 8.9.6.5 0x0020 PWM01 Clock Configuration Register (Default Value: 0x0000\_0000)

Offset:0x0020			Register Name: PCCR01
Bit	Read/Write	Default/Hex	Description
31:9	/	/	/

Offset:0x0020			Register Name: PCCR01
Bit	Read/Write	Default/Hex	Description
8:7	R/W	0x0	PWM01_CLK_SRC Select PWM01 Clock Source 00: HOSC 01: APBO Others: Reserved
6:4	/	/	/
3:0	R/W	0x0	PWM01_CLK_DIV_M PWM01 Clock Divide M 0000: /1 0001: /2 0010: /4 0011: /8 0100: /16 0101: /32 0110: /64 0111: /128 1000: /256 Others: Reserved

#### 8.9.6.6 0x0024 PWM23 Clock Configuration Register (Default Value: 0x0000\_0000)

Offset:0x0024			Register Name: PCCR23
Bit	Read/Write	Default/Hex	Description
31:9	/	/	/
8:7	R/W	0x0	PWM23_CLK_SRC_SEL Select PWM23 Clock Source 00: HOSC 01: APBO Others: Reserved
6:4	/	/	/

Offset:0x0024			Register Name: PCCR23
Bit	Read/Write	Default/Hex	Description
3:0	R/W	0x0	PWM23_CLK_DIV_M PWM23 Clock Divide M 0000: /1 0001: /2 0010: /4 0011: /8 0100: /16 0101: /32 0110: /64 0111: /128 1000: /256 Others: Reserved

#### 8.9.6.7 0x0028 PWM45 Clock Configuration Register (Default Value: 0x0000\_0000)

Offset:0x0028			Register Name: PCCR45
Bit	Read/Write	Default/Hex	Description
31:9	/	/	/
8:7	R/W	0x0	PWM45_CLK_SRC_SEL Select PWM45 Clock Source 00: HOSC 01: APB0 Others: Reserved
6:4	/	/	/

Offset:0x0028			Register Name: PCCR45
Bit	Read/Write	Default/Hex	Description
3:0	R/W	0x0	PWM45_CLK_DIV_M PWM45 Clock Divide M 0000: /1 0001: /2 0010: /4 0011: /8 0100: /16 0101: /32 0110: /64 0111: /128 1000: /256 Others: Reserved

#### 8.9.6.8 0x002C PWM67 Clock Configuration Register (Default Value: 0x0000\_0000)

Offset:0x002C			Register Name: PCCR67
Bit	Read/Write	Default/Hex	Description
31:9	/	/	/
8:7	R/W	0x0	PWM67_CLK_SRC_SEL Select PWM67 Clock Source 00: HOSC 01: APBO Others: Reserved
6:4	/	/	/

Offset:0x002C			Register Name: PCCR67
Bit	Read/Write	Default/Hex	Description
3:0	R/W	0x0	PWM67_CLK_DIV_M PWM67 Clock Divide M 0000: /1 0001: /2 0010: /4 0011: /8 0100: /16 0101: /32 0110: /64 0111: /128 1000: /256 Others: Reserved

#### 8.9.6.9 0x0040 PWM Clock Gating Register (Default Value: 0x0000\_0000)

Offset: 0x0040			Register Name: PCGR
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23	R/W	0x0	PWM7_CLK_BYPASS Bypass clock source (after pre-scale) to PWM7 output 0: not bypass 1: bypass
22	R/W	0x0	PWM6_CLK_BYPASS Bypass clock source (after pre-scale) to PWM6 output 0: not bypass 1: bypass
21	R/W	0x0	PWM5_CLK_BYPASS Bypass clock source (after pre-scale) to PWM5 output 0: not bypass 1: bypass
20	R/W	0x0	PWM4_CLK_BYPASS Bypass clock source (after pre-scale) to PWM4 output 0: not bypass 1: bypass

Offset: 0x0040			Register Name: PCGR
Bit	Read/Write	Default/Hex	Description
19	R/W	0x0	PWM3_CLK_BYPASS Bypass clock source (after pre-scale) to PWM3 output 0: not bypass 1: bypass
18	R/W	0x0	PWM2_CLK_BYPASS Bypass clock source (after pre-scale) to PWM2 output 0: not bypass 1: bypass
17	R/W	0x0	PWM1_CLK_BYPASS Bypass clock source (after pre-scale) to PWM1 output 0: not bypass 1: bypass
16	R/W	0x0	PWM0_CLK_BYPASS Bypass clock source (after pre-scale) to PWM0 output 0: not bypass 1: bypass
15:8	/	/	/
7	R/W	0x0	PWM7_CLK_GATING Gating clock for PWM7 0: Mask 1: Pass
6	R/W	0x0	PWM6_CLK_GATING Gating clock for PWM6 0: Mask 1: Pass
5	R/W	0x0	PWM5_CLK_GATING Gating clock for PWM5 0: Mask 1: Pass
4	R/W	0x0	PWM4_CLK_GATING Gating clock for PWM4 0: Mask 1: Pass

Offset: 0x0040			Register Name: PCGR
Bit	Read/Write	Default/Hex	Description
3	R/W	0x0	PWM3_CLK_GATING Gating clock for PWM3 0: Mask 1: Pass
2	R/W	0x0	PWM2_CLK_GATING Gating clock for PWM2 0: Mask 1: Pass
1	R/W	0x0	PWM1_CLK_GATING Gating clock for PWM1 0: Mask 1: Pass
0	R/W	0x0	PWM0_CLK_GATING Gating clock for PWM0 0: Mask 1: Pass

#### 8.9.6.10 0x0060 PWM01 Dead Zone Control Register (Default Value: 0x0000\_0000)

Offset:0x0060			Register Name: PDZCR01
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:8	R/W	0x0	PWM01_DZ_INTV PWM01 Dead Zone Interval Value
7:1	/	/	/
0	R/W	0x0	PWM01_DZ_EN PWM01 Dead Zone Enable 0: Dead Zone disable 1: Dead Zone enable

#### 8.9.6.11 0x0064 PWM23 Dead Zone Control Register (Default Value: 0x0000\_0000)

Offset:0x0064			Register Name: PDZCR23
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:8	R/W	0x0	PWM23_DZ_INTV PWM23 Dead Zone Interval Value
7:1	/	/	/
0	R/W	0x0	PWM23_DZ_EN PWM23 Dead Zone Enable 0: Dead Zone disable 1: Dead Zone enable

#### 8.9.6.12 0x0068 PWM45 Dead Zone Control Register (Default Value: 0x0000\_0000)

Offset:0x0068			Register Name: PDZCR45
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:8	R/W	0x0	PWM45_DZ_INTV PWM45 Dead Zone Interval Value
7:1	/	/	/
0	R/W	0x0	PWM45_DZ_EN PWM45 Dead Zone Enable 0: Dead Zone disable 1: Dead Zone enable

#### 8.9.6.13 0x006C PWM67 Dead Zone Control Register (Default Value: 0x0000\_0000)

Offset:0x006C			Register Name: PDZCR67
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:8	R/W	0x0	PWM67_DZ_INTV PWM67 Dead Zone Interval Value
7:1	/	/	/

Offset:0x006C			Register Name: PDZCR67
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	PWM67_DZ_EN PWM67 Dead Zone Enable 0: Dead Zone disable 1: Dead Zone enable

#### 8.9.6.14 0x0080 PWM Enable Register (Default Value: 0x0000\_0000)

Offset:0x0080			Register Name: PER
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R/W	0x0	PWM7_EN When PWM is enabled, the 16-bit up-counter starts working and PWM channel7 is permitted to output PWM waveform. 0: PWM disable 1: PWM enable
6	R/W	0x0	PWM6_EN When PWM is enabled, the 16-bit up-counter starts working and PWM channel6 is permitted to output PWM waveform. 0: PWM disable 1: PWM enable
5	R/W	0x0	PWM5_EN When PWM is enabled, the 16-bit up-counter starts working and PWM channel5 is permitted to output PWM waveform. 0: PWM disable 1: PWM enable
4	R/W	0x0	PWM4_EN When PWM is enabled, the 16-bit up-counter starts working and PWM channel4 is permitted to output PWM waveform. 0: PWM disable 1: PWM enable
3	R/W	0x0	PWM3_EN When PWM is enabled, the 16-bit up-counter starts working and PWM channel3 is permitted to output PWM waveform. 0: PWM disable 1: PWM enable

Offset:0x0080			Register Name: PER
Bit	Read/Write	Default/Hex	Description
2	R/W	0x0	<p>PWM2_EN</p> <p>When PWM is enabled, the 16-bit up-counter starts working and PWM channel2 is permitted to output PWM waveform.</p> <p>0: PWM disable 1: PWM enable</p>
1	R/W	0x0	<p>PWM1_EN</p> <p>When PWM is enabled, the 16-bit up-counter starts working and PWM channel1 is permitted to output PWM waveform.</p> <p>0: PWM disable 1: PWM enable</p>
0	R/W	0x0	<p>PWM0_EN</p> <p>When PWM is enabled, the 16-bit up-counter starts working and PWM channel0 is permitted to output PWM waveform.</p> <p>0: PWM disable 1: PWM enable</p>

#### 8.9.6.15 0x0090 PWM Group0 Register (Default Value: 0x0000\_0000)

Offset: 0x0090			Register Name: PGR0
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
17	R/WAC	0x0	<p>PWMG0_START</p> <p>The PWM channels selected in <b>PWMG0_CS</b> start to output PWM waveform at the same time.</p>
16	R/W	0x0	<p>PWMG0_EN</p> <p>PWM Group0 Enable.</p>
15:0	R/W	0x0	<p>PWMG0_CS</p> <p>If bit[i] is set, the PWM i is selected as one channel of PWM Group0.</p>

#### 8.9.6.16 0x0094 PWM Group1 Register (Default Value: 0x0000\_0000)

Offset: 0x0094			Register Name: PGR1
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/

Offset: 0x0094			Register Name: PGR1
Bit	Read/Write	Default/Hex	Description
17	R/WAC	0x0	PWMG1_START The PWM channels selected in <b>PWMG1_CS</b> start to output PWM waveform at the same time.
16	R/W	0x0	PWMG1_EN PWM Group1 Enable.
15:0	R/W	0x0	PWMG1_CS If bit[i] is set, the PWM i is selected as one channel of PWM Group1.

#### 8.9.6.17 0x0098 PWM Group2 Register (Default Value: 0x0000\_0000)

Offset: 0x0098			Register Name: PGR2
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
17	R/WAC	0x0	PWMG2_START The PWM channels selected in <b>PWMG2_CS</b> start to output PWM waveform at the same time.
16	R/W	0x0	PWMG2_EN PWM Group2 Enable.
15:0	R/W	0x0	PWMG2_CS If bit[i] is set, the PWM i is selected as one channel of PWM Group2.

#### 8.9.6.18 0x009C PWM Group3 Register (Default Value: 0x0000\_0000)

Offset: 0x009C			Register Name: PGR3
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
17	R/WAC	0x0	PWMG3_START The PWM channels selected in <b>PWMG3_CS</b> start to output PWM waveform at the same time.
16	R/W	0x0	PWMG3_EN PWM Group3 Enable.
15:0	R/W	0x0	PWMG3_CS If bit[i] is set, the PWM i is selected as one channel of PWM Group3.

**8.9.6.19 0x00C0 Capture Enable Register (Default Value: 0x0000\_0000)**

Offset: 0x00C0			Register Name: CER
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R/W	0x0	CAP7_EN When enabling the capture function, the 16-bit up-counter starts working, and the capture channel7 is permitted to capture external falling edge or rising edge. 0: Capture disable 1: Capture enable
6	R/W	0x0	CAP6_EN When enabling the capture function, the 16-bit up-counter starts working, and the capture channel6 is permitted to capture external falling edge or rising edge. 0: Capture disable 1: Capture enable
5	R/W	0x0	CAP5_EN When enabling the capture function, the 16-bit up-counter starts working, and the capture channel5 is permitted to capture external falling edge or rising edge. 0: Capture disable 1: Capture enable
4	R/W	0x0	CAP4_EN When enabling the capture function, the 16-bit up-counter starts working, and the capture channel4 is permitted to capture external falling edge or rising edge. 0: Capture disable 1: Capture enable
3	R/W	0x0	CAP3_EN When enabling the capture function, the 16-bit up-counter starts working, and the capture channel3 is permitted to capture external falling edge or rising edge. 0: Capture disable 1: Capture enable

Offset: 0x00C0			Register Name: CER
Bit	Read/Write	Default/Hex	Description
2	R/W	0x0	<p>CAP2_EN</p> <p>When enabling the capture function, the 16-bit up-counter starts working, and the capture channel2 is permitted to capture external falling edge or rising edge.</p> <p>0: Capture disable 1: Capture enable</p>
1	R/W	0x0	<p>CAP1_EN</p> <p>When enabling the capture function, the 16-bit up-counter starts working, and the capture channel1 is permitted to capture external falling edge or rising edge.</p> <p>0: Capture disable 1: Capture enable</p>
0	R/W	0x0	<p>CAPO_EN</p> <p>When enabling the capture function, the 16-bit up-counter starts working, and the capture channel is permitted to capture external falling edge or rising edge.</p> <p>0: Capture disable 1: Capture enable</p>

#### 8.9.6.20 0x0100 + N\*0x20 PWM Control Register (Default Value: 0x0000\_0000)

Offset:0x0100+0x0+N*0x20 (N=0~7)			Register Name: PCR
Bit	Read/Write	Default/Hex	Description
31:16	R/W	0x0	<p>PWM_PUL_NUM</p> <p>In pulse mode, the PWM outputs pulse for PWM_CYCLE_NUM+1 times and then stops.</p>
15:12	/	/	/
11	R	0x0	<p>PWM_PERIOD_RDY</p> <p>PWM Period Register Ready</p> <p>0: PWM period register is ready to write 1: PWM period register is busy</p>

Offset:0x0100+0x0+N*0x20 (N=0~7)			Register Name: PCR
Bit	Read/Write	Default/Hex	Description
10	R/WAC	0x0	<p>PWM_PUL_START PWM Pulse Output Start 0: No effect 1: Output pulse for PWM_CYCLE_NUM+1. After finishing configuration for the output pulse, set this bit once , then PWM would output waveform. After the waveform is finished, the bit will be cleared automatically.</p>
9	R/W	0x0	<p>PWM_MODE PWM Output Mode Select 0: Cycle mode 1: Pulse mode</p>
8	R/W	0x0	<p>PWM_ACT_STA PWM Active State 0: Low Level 1: High Level</p>
7:0	R/W	0x0	<p>PWM_PRESCAL_K PWM pre-scale K, actual pre-scale is (K+1). K = 0, actual pre-scale: 1 K = 1, actual pre-scale: 2 K = 2, actual pre-scale: 3 K = 3, actual pre-scale: 4 ... K = 255, actual pre-scale: 256</p>

#### 8.9.6.21 0x0104 + N\*0x20 PWM Period Register (Default Value: 0x0000\_0000)

Offset:0x0100+0x04+N*0x20 (N=0~7)			Register Name: PPR
Bit	Read/Write	Default/Hex	Description
31:16	R/W	0x0	<p>PWM_ENTIRE_CYCLE Number of the entire cycles in the PWM clock. 0: 1 cycle 1: 2 cycles ... N: N+1 cycles If the register needs to be modified dynamically, the PCLK should be faster than the PWM CLK.</p>

Offset:0x0100+0x04+N*0x20 (N=0~7)			Register Name: PPR
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0x0	<p>PWM_ACT_CYCLE Number of the active cycles in the PWM clock. 0: 0 cycle 1: 1 cycle ... N: N cycles</p>

#### 8.9.6.22 0x0108 + N\*0x20 PWM Counter Register (Default Value: 0x0000\_0000)

Offset:0x0100+0x08+N*0x20 (N=0~7)			Register Name: PCNTR
Bit	Read/Write	Default/Hex	Description
31:16	R/W	0x0	<p>PWM_COUNTER_START PWM counter value is set for phase control.</p>
15:0	R	0x0	<p>PWM_COUNTER_STATUS On PWM output or capture input, reading this register could get the current value of the PWM 16-bit up-counter.</p>

#### 8.9.6.23 0x010C + N\*0x20 PWM Pulse Counter Register (Default Value: 0x0000\_0000)

Offset: 0x0100+0x0C+N*0x20 (N=0~7)			Register Name: PPCNTR
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R	0x0	<p>PWM_PUL_COUNTER_STATUS On PWM output, reading this register could get the current value of the PWM pulse counter.</p>

#### 8.9.6.24 0x0110 + N\*0x20 PWM Capture Control Register (Default Value: 0x0000\_0000)

Offset:0x0100+0x10+N*0x20 (N=0~7)			Register Name: CCR
Bit	Read/Write	Default/Hex	Description
31:5	/	/	/

Offset:0x0100+0x10+N*0x20 (N=0~7)			Register Name: CCR
Bit	Read/Write	Default/Hex	Description
4	R/W1C	0x0	CRLF When the capture channel captures a rising edge, the current value of the 16-bit up-counter is latched to CRLR, and then this bit is set 1 by hardware. Write 1 to clear this bit.
3	R/W1C	0x0	CFLF When the capture channel captures a falling edge, the current value of the 16-bit up-counter is latched to CFLR, and then this bit is set 1 by hardware. Write 1 to clear this bit.
2	R/W	0x0	CRTE Rising edge capture trigger enable
1	R/W	0x0	CFTE Falling edge capture trigger enable
0	R/W	0x0	CAPINV Inverse the signal input from capture channel before 16-bit counter of capture channel. 0: not inverse 1: inverse

#### 8.9.6.25 0x0114 + N\*0x20 PWM Capture Rise Lock Register (Default Value: 0x0000\_0000)

Offset:0x0100+0x14+N*0x20 (N=0~7)			Register Name: CRLR
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R	0x0	CRLR When the capture channel captures a rising edge, the current value of the 16-bit up-counter is latched to the register.

#### 8.9.6.26 0x0118 + N\*0x20 PWM Capture Fall Lock Register (Default Value: 0x0000\_0000)

Offset:0x0100+0x18+N*0x20 (N=0~7)			Register Name: CFLR
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/

Offset:0x0100+0x18+N*0x20 (N=0~7)			Register Name: CFLR
Bit	Read/Write	Default/Hex	Description
15:0	R	0x0	CFLR When the capture channel captures a falling edge, the current value of the 16-bit up-counter is latched to the register.

## 8.10 LEDC

### 8.10.1 Overview

The LEDC is used to control the external LED lamp.

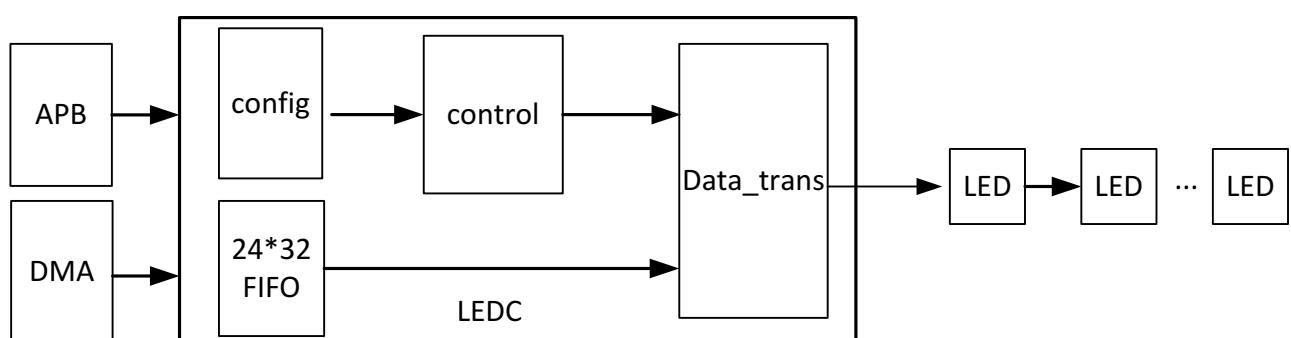
The LEDC has the following features:

- Configurable LED output high-/low-level width
- Configurable LED reset time
- LEDC data supports DMA configuration mode and CPU configuration mode
- Maximum 1024 LEDs serial connect
- LED data transfer rate up to 800 kbit/s
- Configurable RGB display mode
- The default level of non-data output is configurable

### 8.10.2 Block Diagram

The following figure shows a block diagram of the LEDC.

**Figure 8-85 LEDC Block Diagram**



LEDC contains the following sub-blocks:

**Table 8-30 LEDC Sub-blocks**

Sub-block	Description
config	register configuration
control	LEDC timing control and status control
FIFO	24-bit width x 32 depth
Data_trans	Convert input data to the 0 and 1 characters of LED

### 8.10.3 Functional Description

#### 8.10.3.1 External Signals

The following table describes the external signals of the LEDC.

**Table 8-31 LEDC External Signals**

Signal	Description	Type
LEDC-DO	Intelligent Control LED Signal Output	O

#### 8.10.3.2 Clock Sources

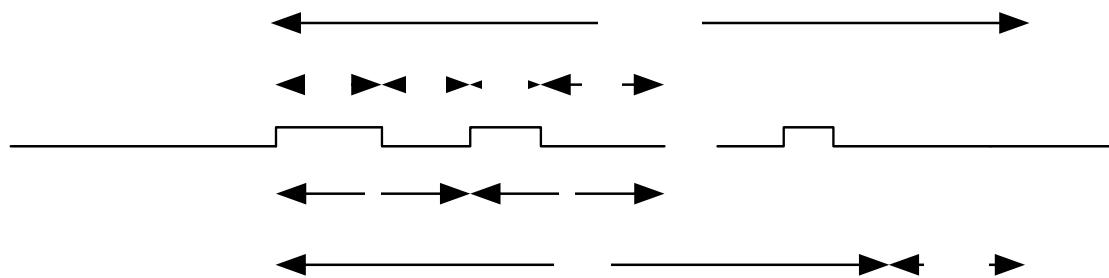
The following table describes the clock sources of the LEDC.

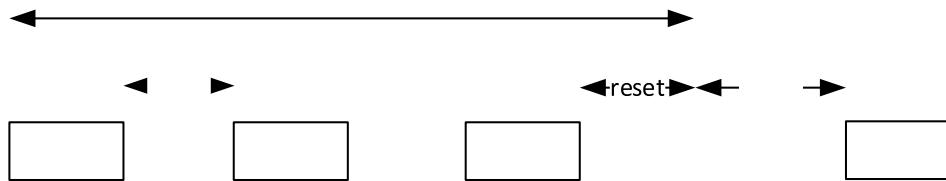
**Table 8-32 LEDC Clock Sources**

Clock Sources	Description
HOSC	24 MHz
PLL_PERI(1X)	Peripheral Clock. The default value is 600 MHz

#### 8.10.3.3 LEDC Timing

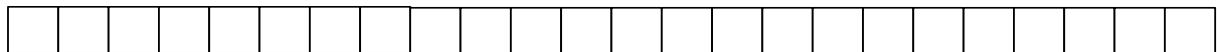
**Figure 8-86 LEDC Package Output Timing Diagram**



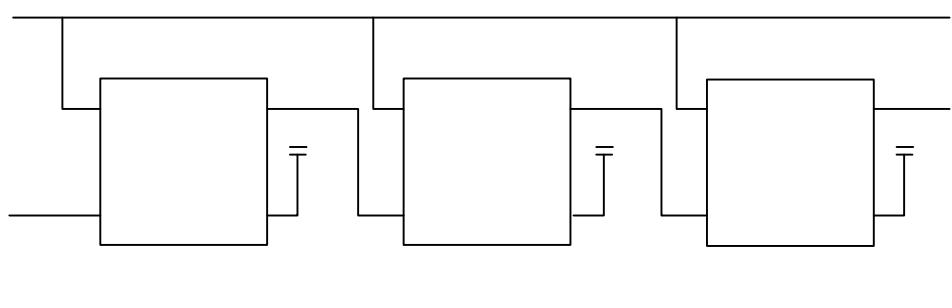
**Figure 8-87 LEDC 1-frame Output Timing Diagram**

#### 8.10.3.4 LEDC Input Data Structure

The RGB mode of LEDC data is configurable. By default, the data is sent in GRB order, and the higher bit is transmitted first.

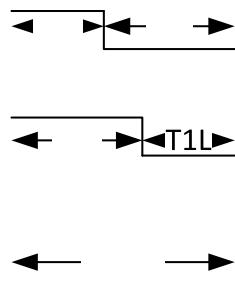
**Figure 8-88 LEDC Input Data Structure**

#### 8.10.3.5 LEDC Typical Circuit

**Figure 8-89 LEDC Typical Circuit**

### 8.10.3.6 LEDC Data Input Code

Figure 8-90 LEDC Data Input Code



### 8.10.3.7 LEDC Data Transfer Time

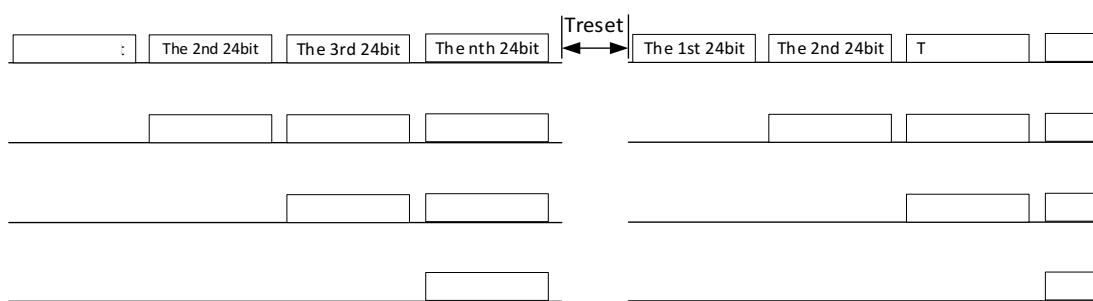
The time parameter of the typical LED specification shows as follows.

Table 8-33 Time Parameters of Typical LED Specification

<b>T0H</b>	0 code, high-level time	220 ns to 380 ns
<b>T0L</b>	0 code, low-level time	580 ns to 1.6 us
<b>T1H</b>	1 code, high-level time	580 ns to 1.6 us
<b>T1L</b>	1 code, low-level time	220 ns to 420 ns
<b>RESET</b>	Frame unit, low-level time	> 280 us

### 8.10.3.8 LEDC Data Transfer Mode

Figure 8-91 LEDC Data Transfer Mode



### 8.10.3.9 LEDC Parameter

1. PAD rate > 800 kbit/s
2. LED number supported:

$T_{0\text{-code}}$ : 800 ns to 1980 ns,  $T_{1\text{-code}}$ : 800 ns to 2020 ns

When the LED refresh rate is 30 frame/s, LED number supported is  $(1 \text{ s}/30\text{-}280 \text{ us})/((800 \text{ ns to } 2020 \text{ ns})^{\star}24)$   
=1024 ↓to 681.

When the LED refresh rate is 60 frame/s, LED number supported is  $(1 \text{ s}/60\text{-}280 \text{ us})/((800 \text{ ns to } 2020 \text{ ns})^{\star}24)$   
=853 ↓to 338.

### 8.10.3.10 LEDC Data Transfer

The LEDC supports DMA data transfer mode or CPU data transfer mode. The DMA data transfer mode is set by LEDC\_DMA\_EN

#### Data transfer in DMA mode

When the valid space of internal FIFO is greater than the setting FIFO free space threshold, the LEDC sends DMA\_REQ to require DMA to transfer data from DRAM to LEDC. The maximum data transfer size in DMA mode is 16 words. (The internal FIFO level is 32.)

#### Data transfer in CPU mode

When the valid space of internal FIFO is greater than the setting FIFO free space threshold, the LEDC sends LEDC\_CPUREQ\_INT to require CPU to transfer data to LEDC. The transfer data size in CPU mode is controlled by software. The internal FIFO destination address is 0x06700014. The data width is 32-bit. (The lower 24-bit is valid.)

### 8.10.3.11 LEDC Interrupt

Module Name	Description
FIFO_OVERFLOW_INT	<p>FIFO overflow interrupt.</p> <p>The data written by external is more than the maximum storage space of LED FIFO, the LEDC will be in data loss state. At this time, software needs to deal with the abnormal situation. The processing mode is as follows.</p> <p>The software can query LED_FIFO_DATA_REG to determine which data has been stored in the internal FIFO of LEDC. The LEDC performs soft_reset operation to refresh all data.</p>
WAITDATA_TIMEOUT_INT	<p>Wait for data timeout interrupt</p> <p>When internal FIFO of LEDC cannot get data because of some abnormal situation, the timeout interrupt is set after led_wait_data_time, now the LEDC is in WAIT_DATA state, and the LEDC outputs a level state configured by LED_POLARITY; in the course of wait_data, if the new data arrives, the LEDC will continue to send data, at this time the software needs to notice whether the waiting time of the LEDC exceeds the operation time of reset. If the waiting time of the LEDC exceeds the operation time of reset (this is equivalent to reset operation sent by LEDC), the LED may enter in refresh state, the data has not been sent.</p>
FIFO_CPUREQ_INT	<p>FIFO request CPU data interrupt</p> <p>When FIFO data is less than a threshold, the interrupt will be reported to the CPU.</p>
LEDC_TRANS_FINISH_INT	<p>Data transfer complete interrupt</p> <p>The value indicates that the data configured as total_data_length has been transferred completely.</p>

LEDC interrupt usage scenario:

#### CPU mode

The software can enable GLOBAL\_INT\_EN, FIFO\_CPUREQ\_INT\_EN, WAITDATA\_TIMEOUT\_INT\_EN, FIFO\_OVERFLOW\_INT\_EN, LEDC\_TRANS\_FINISH\_INT\_EN, and cooperate with LEDC\_FIFO\_TRIG\_LEVEL to use. When FIFO\_CPUREQ\_INT is set to 1, the software can configure data of LEDC\_FIFO\_TRIG\_LEVEL to LEDC.

#### DMA mode

The software can enable GLOBAL\_INT\_EN, WAITDATA\_TIMEOUT\_INT\_EN, FIFO\_OVERFLOW\_INT\_EN, LEDC\_TRANS\_FINISH\_INT\_EN, and cooperate with LEDC\_FIFO\_TRIG\_LEVEL to use. When DMA receives LEDC\_DMA\_REQ, DMA can transfer data of LEDC\_FIFO\_TRIG\_LEVEL to LEDC.

## 8.10.4 Programming Guidelines

### 8.10.4.1 LEDC Normal Configuration Process

**Step 1** Configure LEDC\_CLK and bus pclk.

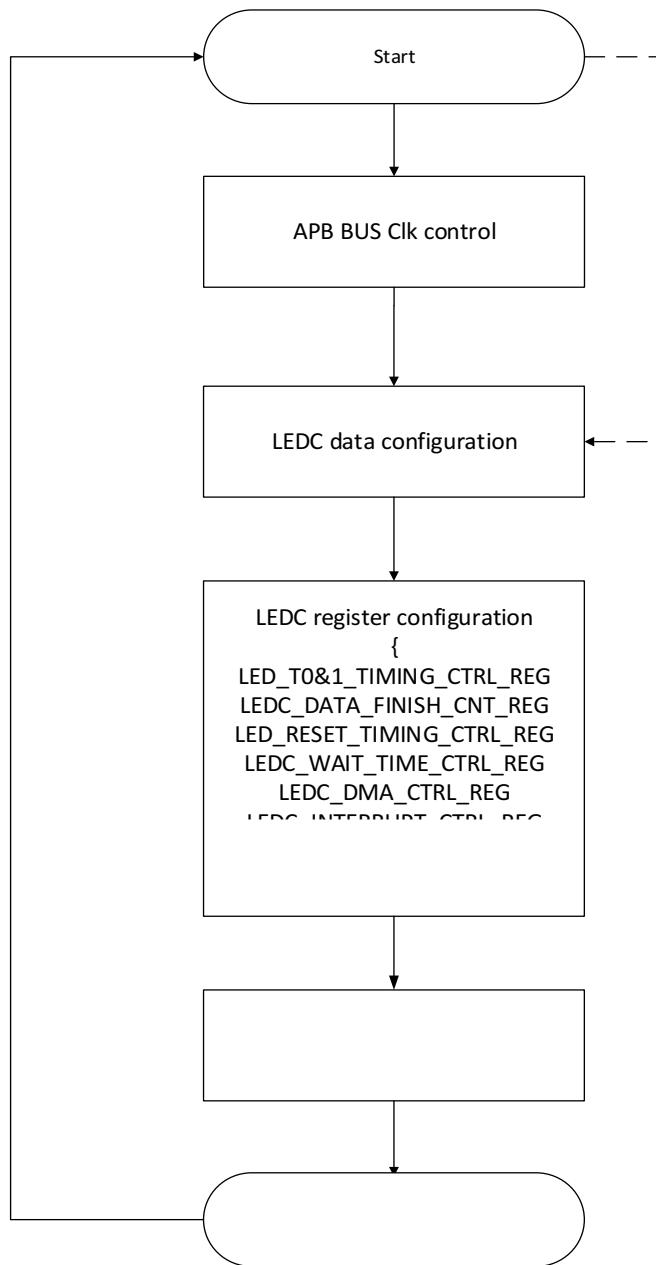
**Step 2** Configure the written LEDC data.

**Step 3** Configure [LED T01 TIMING CTRL REG](#), [LEDC DATA FINISH CNT REG](#), [LED RESET TIMING CTRL REG](#), [LEDC WAIT TIME0 CTRL REG](#), [LEDC DMA CTRL REG](#), [LEDC INTERRUPT CTRL REG](#). Configure 0-code, 1-code, reset time, LEDC waiting time, and the number of external connected LEDC and the threshold of DMA transfer data.

**Step 4** Configure [LEDC CTRL REG](#) to enable LEDC\_EN, the LEDC will start to output data.

**Step 5** When the LEDC interrupt is pulled up, it indicates the configured data has transferred complete, at this time LED\_EN will be set to 0, and the read/write point of LEDC FIFO is cleared to 0.

**Step 6** Repeat step1, 2, 3, 4 to re-execute a new round of configuration, enable LEDC\_EN, the LEDC will start new data transfer.

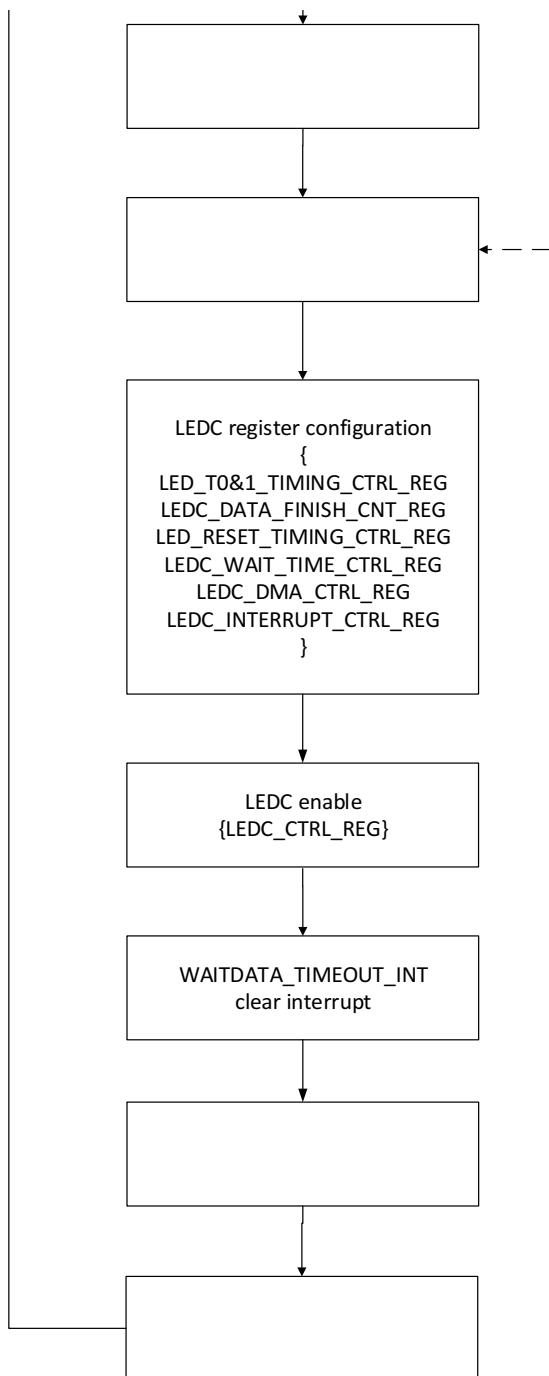
**Figure 8-92 LEDC Normal Configuration Process**

#### 8.10.4.2 LEDC Abnormal Scene Processing Flow

##### **WAITDATA\_TIMEOUT Abnormal Status**

**Step 1** When WAITDATA\_TIMEOUT\_INT appears, it indicates the internal FIFO data request of LEDC cannot obtain a response, at this time if the default output level is low, then the external LED may think there was a reset operation and cause LED data to be flushed incorrectly.

- Step 2** The LEDC needs to be performed soft\_reset operation (LED\_C\_SOFT\_RESET=1); after soft\_reset, the LEDC\_EN will be pulled-down automatically, all internal status register and control state machine will return to the idle state, the LEDC FIFO read & write point is cleared to 0, the LEDC interrupt is cleared.
- Step 3** Setting reset\_led\_en to 1 indicates LEDC can actively send a reset operation to ensure the external LED lamp in the right state.
- Step 4** The software reads the status of reset\_led\_en, when the status value is 1, it indicates LEDC does not perform the transmission of LED reset operation; when the status value is 0, the LEDC completes the transmission of LED reset operation.
- Step 5** When LEDC reset operation finishes, the LEDC data and register configuration need to be re-operated to start re-transmission data operation.

**Figure 8-93 LEDC Timeout Abnormal Processing Flow**

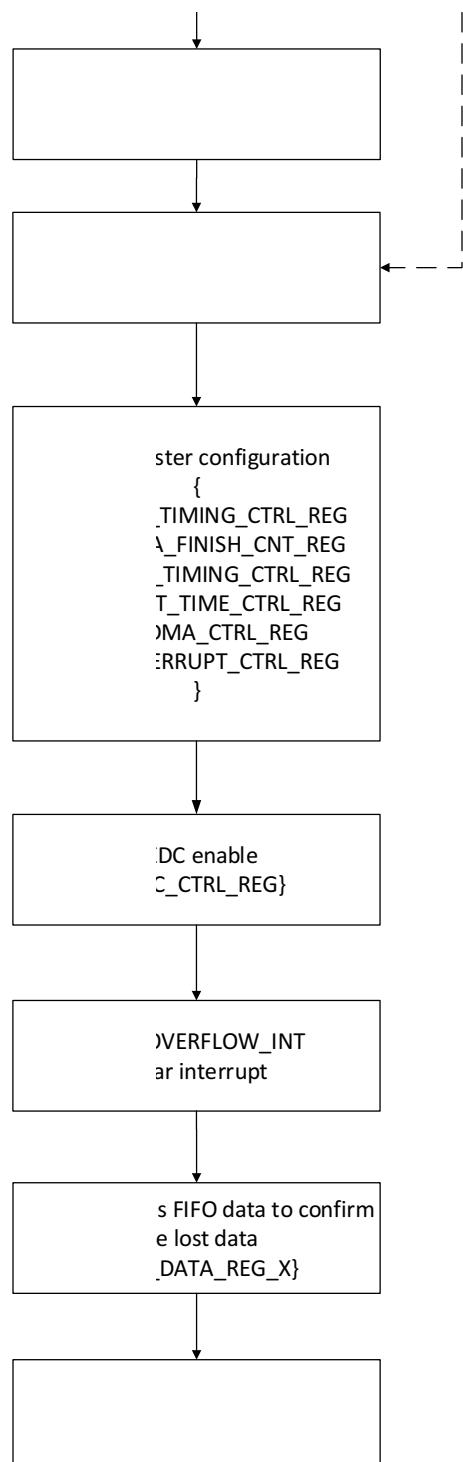
#### FIFO Overflow Abnormal Status

- Step 1** When FIFO\_OVERFLOW\_INT appears, it indicates the data configured by software exceeds the LEDC FIFO space, at this time the redundant data will be lost.
- Step 2** The software needs to read data in [LEDC\\_FIFO\\_DATA\\_X](#) to confirm the lost data.

**Step 3** The software re-configures the lost data to the LEDC.

**Step 4** If the software uses the soft\_reset operation, the operation is the same with the timeout abnormal processing flow.

**Figure 8-94 FIFO Overflow Abnormal Processing Flow**



### 8.10.5 Register List

Module Name	Base Address
LEDC	0x02008000

Register Name	Offset	Description
LEDC_CTRL_REG	0x0000	LEDC Control Register
LED_T01_TIMING_CTRL_REG	0x0004	LEDC T0 & T1 Timing Control Register
LEDC_DATA_FINISH_CNT_REG	0x0008	LEDC Data Finish Counter Register
LED_RESET_TIMING_CTRL_REG	0x000C	LEDC Reset Timing Control Register
LEDC_WAIT_TIME0_CTRL_REG	0x0010	LEDC Wait Time0 Control Register
LEDC_DATA_REG	0x0014	LEDC Data Register
LEDC_DMA_CTRL_REG	0x0018	LEDC DMA Control Register
LEDC_INT_CTRL_REG	0x001C	LEDC Interrupt Control Register
LEDC_INT_STS_REG	0x0020	LEDC Interrupt Status Register
LEDC_WAIT_TIME1_CTRL_REG	0x0028	LEDC Wait Time1 Control Register
LEDC_FIFO_DATA_REG	0x0030+0x04*N	LEDC FIFO Data Register

### 8.10.6 Register Description

#### 8.10.6.1 0x0000 LEDC Control Register (Default Value: 0x0000\_003C)

Offset: 0x0000			Register Name: LEDC_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:29	/	/	/
28:16	R/W	0x0	<p>TOTAL_DATA_LENGTH Total length of transfer data (range: 0 to 8K, unit: 32-bit, only low 24-bit is valid)</p> <p>The field is recommended to be set to an integer multiple of (LED_NUM+1).</p> <p>If TOTAL_DATA_LENGTH is greater than (LED_NUM+1), but non-integer multiple, the last frame of data will transfer data less than (LED_NUM+1).</p>
15:11	/	/	/
10	R/W	0x0	<p>RESET_LED_EN Write operation:</p>

Offset: 0x0000			Register Name: LEDC_CTRL_REG																									
Bit	Read/Write	Default/Hex	Description																									
			<p>The software writes 1 to the bit, the CPU triggers LEDC to transfer a reset to LED.</p> <p><b>Only when LEDC is in IDLE status, the reset can be performed. After the reset finished, the control state machine returns to the IDLE status. To return LEDC to the IDLE status, it also needs to be used with SOFT_RESET.</b></p> <p><b>When the software sets the bit, the software can read the bit to check if the reset is complete.</b></p> <p>Read operation:</p> <p>0: LEDC completes the transmission of the LED reset operation 1: LEDC does not complete the transmission of the LED reset operation</p>																									
9	/	/	/																									
8:6	R/W	0x0	<p>LED_RGB_MODE</p> <p>000 GRB (bypass)</p> <p>001 GBR</p> <p>010 RGB</p> <p>011 RBG</p> <p>100 BGR</p> <p>101 BRG</p> <p>By default, the software configures data to LEDC according to GRB (MSB) mode, the LEDC internal combines data to output to the external LED.</p> <p>Other modes configure as follows.</p> <table border="1"> <thead> <tr> <th>Software Input Mode</th> <th>Configuration</th> <th>LED Output Mode</th> </tr> </thead> <tbody> <tr> <td rowspan="6">GRB</td> <td>000</td> <td>GRB</td> </tr> <tr> <td>001</td> <td>GBR</td> </tr> <tr> <td>010</td> <td>RGB</td> </tr> <tr> <td>011</td> <td>RBG</td> </tr> <tr> <td>100</td> <td>BGR</td> </tr> <tr> <td>101</td> <td>BRG</td> </tr> <tr> <td rowspan="4">GBR</td> <td>000</td> <td>GBR</td> </tr> <tr> <td>001</td> <td>GRB</td> </tr> <tr> <td>010</td> <td>BGR</td> </tr> <tr> <td>011</td> <td>BRG</td> </tr> </tbody> </table>	Software Input Mode	Configuration	LED Output Mode	GRB	000	GRB	001	GBR	010	RGB	011	RBG	100	BGR	101	BRG	GBR	000	GBR	001	GRB	010	BGR	011	BRG
Software Input Mode	Configuration	LED Output Mode																										
GRB	000	GRB																										
	001	GBR																										
	010	RGB																										
	011	RBG																										
	100	BGR																										
	101	BRG																										
GBR	000	GBR																										
	001	GRB																										
	010	BGR																										
	011	BRG																										

Offset: 0x0000			Register Name: LEDC_CTRL_REG																																																												
Bit	Read/Write	Default/Hex	Description																																																												
			<table border="1"> <tr><td></td><td>100</td><td>RGB</td></tr> <tr><td></td><td>101</td><td>RBG</td></tr> <tr><td rowspan="7">RGB</td><td>000</td><td>RGB</td></tr> <tr><td>001</td><td>RBG</td></tr> <tr><td>010</td><td>GRB</td></tr> <tr><td>011</td><td>GBR</td></tr> <tr><td>100</td><td>BRG</td></tr> <tr><td>101</td><td>BGR</td></tr> <tr><td>000</td><td>RBG</td></tr> <tr><td rowspan="6">RBG</td><td>001</td><td>RGB</td></tr> <tr><td>010</td><td>BRG</td></tr> <tr><td>011</td><td>BGR</td></tr> <tr><td>100</td><td>GRB</td></tr> <tr><td>101</td><td>GBR</td></tr> <tr><td>000</td><td>BGR</td></tr> <tr><td rowspan="6">BGR</td><td>001</td><td>BRG</td></tr> <tr><td>010</td><td>GBR</td></tr> <tr><td>011</td><td>GRB</td></tr> <tr><td>100</td><td>RBG</td></tr> <tr><td>101</td><td>RGB</td></tr> <tr><td>000</td><td>BRG</td></tr> <tr><td rowspan="6">BRG</td><td>001</td><td>BGR</td></tr> <tr><td>010</td><td>RBG</td></tr> <tr><td>011</td><td>RGB</td></tr> <tr><td>100</td><td>GBR</td></tr> <tr><td>101</td><td>GRB</td></tr> <tr><td>000</td><td>GRB</td></tr> </table>		100	RGB		101	RBG	RGB	000	RGB	001	RBG	010	GRB	011	GBR	100	BRG	101	BGR	000	RBG	RBG	001	RGB	010	BRG	011	BGR	100	GRB	101	GBR	000	BGR	BGR	001	BRG	010	GBR	011	GRB	100	RBG	101	RGB	000	BRG	BRG	001	BGR	010	RBG	011	RGB	100	GBR	101	GRB	000	GRB
	100	RGB																																																													
	101	RBG																																																													
RGB	000	RGB																																																													
	001	RBG																																																													
	010	GRB																																																													
	011	GBR																																																													
	100	BRG																																																													
	101	BGR																																																													
	000	RBG																																																													
RBG	001	RGB																																																													
	010	BRG																																																													
	011	BGR																																																													
	100	GRB																																																													
	101	GBR																																																													
	000	BGR																																																													
BGR	001	BRG																																																													
	010	GBR																																																													
	011	GRB																																																													
	100	RBG																																																													
	101	RGB																																																													
	000	BRG																																																													
BRG	001	BGR																																																													
	010	RBG																																																													
	011	RGB																																																													
	100	GBR																																																													
	101	GRB																																																													
	000	GRB																																																													
5	R/W	0x1	<p>LED_MSB_TOP</p> <p>Adjust sequence of the combined GRB data</p> <p>0: LSB 1: MSB</p>																																																												
4	R/W	0x1	<p>LED_MSB_G</p> <p>MSB control for Green data</p> <p>0: LSB</p>																																																												

Offset: 0x0000			Register Name: LEDC_CTRL_REG
Bit	Read/Write	Default/Hex	Description
			1: MSB
3	R/W	0x1	<p>LED_MSB_R MSB control for Red data 0: LSB 1: MSB</p>
2	R/W	0x1	<p>LED_MSB_B MSB control for Blue data 0: LSB 1: MSB</p>
1	R/W1C	0x0	<p>LEDC_SOFT_RESET LEDC soft reset Write 1 to clear it automatically. The ranges of LEDC soft reset include the following points: all internal status registers, the control state machine returns to in idle status, the LEDC FIFO read &amp; write point is cleared to 0, the LEDC interrupt is cleared; and the affected registers are followed.</p> <ul style="list-style-type: none"> <li>1.LEDC_CTRL_REG (LEDC_EN is cleared to 0);</li> <li>2. PLL_T0&amp;1_TIMING_CTRL_REG remains unchanged;</li> <li>3. LEDC_DATA_FINISH_CNT_REG (LEDC_DATA_FINISH_CNT is cleared to 0);</li> <li>4.LED_RESET_TIMING_CTRL_REG remains unchanged;</li> <li>5. LEDC_WAIT_TIME_CTRL_REG remains unchanged;</li> <li>6. LEDC_DMA_CTRL_REG remains unchanged;</li> <li>7. LEDC_INTERRUPT_CTRL_REG remains unchanged;</li> <li>8.LEDC_INT_STS _REG is cleared to 0;</li> <li>9. LEDC_CLK_GATING_REG remains unchanged;</li> <li>10.LEDC_FIFO_DATA_REG remains unchanged;</li> </ul>
0	R/W	0x0	<p>LEDC_EN LEDC Enable 0: Disable 1: Enable That the bit is enabled indicates LEDC can be started when LEDC data finished transmission or LEDC_EN is cleared to 0 by hardware in LEDC_SOFT_RESET situation.</p>

**8.10.6.2 0x0004 LEDC T0 & T1 Timing Control Register (Default Value: 0x0286\_01D3)**

Offset: 0x0004			Register Name: LED_T01_TIMING_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:27	/	/	/
26:21	R/W	0x14	T1H_TIME LED T1H time Unit: cycle (24 MHz), T1H_TIME = 42 ns*(N+1) The default value is 882 ns, the range is 80 ns–2560 ns. N: 1–3F. When is 0, T1H_TIME = 3F
20:16	R/W	0x6	T1L_TIME LED T1L time Unit: cycle (24 MHz), T1L_TIME = 42 ns*(N+1) The default value is 294 ns, the range is 80 ns–1280 ns. N: 1–1F. When is 0, T1L_TIME = 1F
15:11	/	/	/
10:6	R/W	0x7	TOH_TIME LED TOh time Unit: cycle (24 MHz), TOH_TIME = 42 ns*(N+1) The default value is 336 ns, the range is 80 ns–1280 ns. N: 1–1F. When is 0, TOH_TIME = 1F
5:0	R/W	0x13	TOL_TIME LED T0l time Unit: cycle (24 MHz), TOL_TIME = 42 ns*(N+1) The default value is 840 ns, the range is 80 ns–2560 ns. N: 1–3F. When is 0, TOL_TIME = 3F

**8.10.6.3 0x0008 LEDC Data Finish Counter Register (Default Value: 0x1D4C\_0000)**

Offset: 0x0008			Register Name: LEDC_DATA_FINISH_CNT_REG
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/

Offset: 0x0008			Register Name: LEDC_DATA_FINISH_CNT_REG
Bit	Read/Write	Default/Hex	Description
29:16	R/W	0x1D4C	<p>LED_WAIT_DATA_TIME The value is the time that internal FIFO in LEDC is waiting for data. When the time is exceeded, the LEDC will send the wait_data_timeout_int interrupt. (This is an abnormal situation, software needs to reset LEDC.) The value is about 300 us by default. The adjust range is 80 ns–655 us. led_wait_data_time=42ns*(N+1). N: 1–1FFF. When the field is 0, LEDC_WAIT_DATA_TIME=1FFF</p>
15:13	/	/	/
12:0	R	0x0	<p>LED_DATA_FINISH_CNT The value is the total LED data that have been sent. (Range: 0–8k)</p>

#### 8.10.6.4 0x000C LEDC Reset Timing Control Register (Default Value: 0x1D4C\_0000)

Offset: 0x000C			Register Name: LED_RESET_TIMING_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:29	/	/	/
28:16	R/W	0x1D4C	<p>TR_TIME Reset time control of LED lamp Unit: cycle (24 MHz), tr_time=42 ns*(N+1) The default value is 300 us. The adjust range is 80 ns–327 us. N: 1–1FFF</p>
15:10	/	/	/
9:0	R/W	0x0	<p>LED_NUM The value is the number of external LED lamp. Maximum up to 1024. The default value 0 indicates that 1 LED lamp is external connected. The range is from 0 to 1023.</p>

**8.10.6.5 0x0010 LEDC Wait Time 0 Control Register (Default Value: 0x0000\_00FF)**

Offset: 0x0010			Register Name: LEDC_WAIT_TIME0_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:9	/	/	/
8	R/W	0x0	<p>WAIT_TIM0_EN WAIT_TIME0 enable When it is 1, the controller automatically inserts waiting time between LED package data. 0: Disable 1: Enable</p>
7:0	R/W	0xFF	<p>TOTAL_WAIT_TIME0 Waiting time between 2 LED data. The LEDC output is low level. The adjust range is 80 ns–10 us. wait_time0=42 ns*(N+1) Unit: cycle(24 MHz) N: 1–FF</p>

**8.10.6.6 0x0014 LEDC Data Register (Default Value: 0x0000\_0000)**

Offset: 0x0014			Register Name: LEDC_DATA_REG
Bit	Read/Write	Default/Hex	Description
31:0	W	0x0	LEDC DATA LED display data (the lower 24-bit is valid)

**8.10.6.7 0x0018 LEDC DMA Control Register (Default Value: 0x0000\_002F)**

Offset: 0x0018			Register Name: LEDC_DMA_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:6	/	/	/
5	R/W	0x1	<p>LEDC_DMA_EN LEDC DMA request enable 0: Disable request of DMA transfer data 1: Enable request of DMA transfer data</p>

Offset: 0x0018			Register Name: LEDC_DMA_CTRL_REG
Bit	Read/Write	Default/Hex	Description
4:0	R/W	0x0F	<p>LEDC_FIFO_TRIG_LEVEL The remaining space of internal FIFO in LEDC The internal FIFO in LEDC is 24*32.</p> <p>When the remaining space of internal FIFO in LEDC is more than or equal to LEDFIFO_TRIG_LEVEL, the DMA or the CPU request will generate. The default value is 15.</p> <p>The adjusted value is from 1 to 31. The recommended configuration is 7 or 15. When the configuration value is 0, LEDFIFO_TRIG_LEVEL=F.</p>

#### 8.10.6.8 0x001C LEDC Interrupt Control Register (Default Value: 0x0000\_0000)

Offset: 0x001C			Register Name: LEDC_INTERRUPT_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:6	/	/	/
5	R/W	0x0	<p>GLOBAL_INT_EN Global interrupt enable 0: Disable 1: Enable</p>
4	R/W	0x0	<p>FIFO_OVERFLOW_INT_EN FIFO overflow interrupt enable When the data written by the software is more than the internal FIFO level of LEDC, the LEDC is in the data loss state. 0: Disable 1: Enable</p>
3	R/W	0x0	<p>WAITDATA_TIMEOUT_INT_EN The internal FIFO in LEDC cannot get data because of some abnormal situation, after the time of led_wait_data_time, the interrupt will be enabled. 0: Disable 1: Enable</p>
2	/	/	/
1	R/W	0x0	<p>FIFO_CPUREQ_INT_EN FIFO request CPU data interrupt enable 0: Disable 1: Enable</p>

Offset: 0x001C			Register Name: LEDC_INTERRUPT_CTRL_REG
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	<p>LED_TRANS_FINISH_INT_EN Data transmission complete interrupt enable 0: Disable 1: Enable</p>

#### 8.10.6.9 0x0020 LEDC Interrupt Status Register (Default Value: 0x0002\_0000)

Offset: 0x0020			Register Name: LEDC_INT_STS_REG
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
17	R	0x1	FIFO_EMPTY FIFO empty status flag
16	R	0x0	FIFO_FULL FIFO full status flag
15:10	R	0x0	FIFO_WLW FIFO internal valid data depth It indicates the space FIFO has been occupied.
9:5	/	/	/
4	R/W1C	0x0	<p>FIFO_OVERFLOW_INT FIFO overflow interrupt The data written by external is more than the maximum storage space of LED FIFO, the LEDC will be in the data loss state. At this time, the software needs to deal with the abnormal situation. The processing mode is as follows.</p> <ul style="list-style-type: none"> <li>(1) The software can query LED_FIFO_DATA_REG to determine which data has been stored in the internal FIFO of LEDC.</li> <li>(2) The LEDC performs soft_reset operation to refresh all data.</li> </ul> <p>0: FIFO not overflow 1: FIFO overflow</p>

Offset: 0x0020			Register Name: LEDC_INT_STS_REG
Bit	Read/Write	Default/Hex	Description
3	R/W1C	0x0	<p>WAITDATA_TIMEOUT_INT When internal FIFO of LEDC cannot get data because of some abnormal situation, after led_wait_data_time, the timeout interrupt is set, the LEDC is in WAIT_DATA state, the LEDC outputs a level state configured by LED_POLARITY; in the course of wait_data, if the new data arrives, the LEDC will continue to send data, at this time software needs to notice whether the waiting time of LEDC exceeds the operation time of reset. If the waiting time of LEDC exceeds the operation time of reset (this is equivalent to reset operation sent by LEDC), the LED may enter in refresh state, the data has not been sent.</p> <p>0: LEDC not timeout 1: LEDC timeout</p>
2	/	/	/
1	R/W1C	0x0	<p>FIFO_CPUREQ_INT FIFO request CPU data interrupt When FIFO data is less than the threshold, the interrupt will be reported to the CPU.</p> <p>0: FIFO does not request that CPU transfers data 1: FIFO requests that CPU transfers data</p>
0	R/W1C	0x0	<p>LED_TRANS_FINISH_INT Data transfer complete interrupt The value indicates that the data configured as total_data_length is transferred completely.</p> <p>0: Data is not transferred completely 1: Data is transferred completely</p>

#### 8.10.6.10 0x0028 LEDC Wait Time 1 Control Register (Default Value: 0x01FF\_FFFF)

Offset: 0x0028			Register Name: LEDC_WAIT_TIME1_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<p>WAIT_TIM1_EN 0: Disable 1: Enable WAIT_TIME1 enable When the bit is 1, the controller automatically inserts the waiting time between the LED frame data.</p>

Offset: 0x0028			Register Name: LEDC_WAIT_TIME1_CTRL_REG
Bit	Read/Write	Default/Hex	Description
30:0	R/W	0x01FFFFFF	<p>TOTAL_WAIT_TIME1                      Waiting time between 2 frame data.                      The LEDC output is low level.</p> <p>The adjust range is 80 ns– 85 s. wait_time1=42 ns*(N+1)                      Unit: cycle (24 MHz)                      N: 0x80–0x7FFFFFFF                      If the value is 0, TOTAL_WAIT_TIME1=0x7FFFFFFF</p>

#### 8.10.6.11 0x0030+N\*0x04 LEDC FIFO Data Register X (Default Value: 0x0000\_0000)

Offset: 0x0030+N*0x04 (N=0–31)			Register Name: LEDC_FIFO_DATA_X
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	<p>LEDC_FIFO_DATA_X                      Internal FIFO data of LEDC                      The lower 24-bit is valid.</p>

## 8.11 EMAC

### 8.11.1 Overview

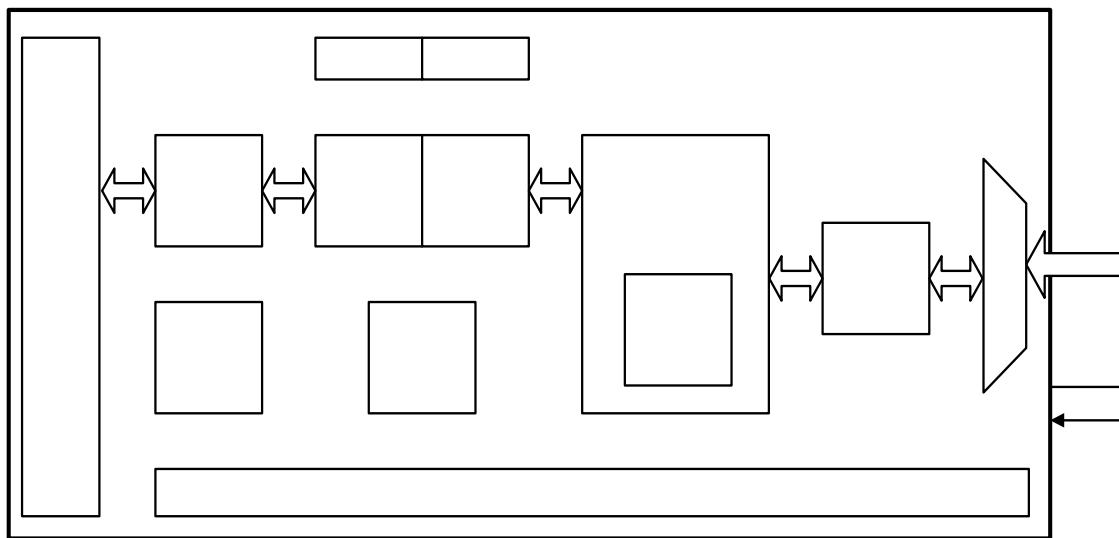
The Ethernet Medium Access Controller (EMAC) enables a host to transmit and receive data over Ethernet in compliance with the IEEE 802.3-2002 standard. It supports 10/100/1000 Mbit/s external PHY with RMII/RGMII interface in full-duplex and half-duplex modes. The internal DMA is designed for packet-oriented data transfers based on a linked list of descriptors.

The EMAC has the following features:

- One 10/100/1000 Mbit/s Ethernet port with reduced gigabit media independent interface (RGMII) and reduced media independent interface (RMII) interfaces, for connecting the external EPHY
- Compliant with IEEE 802.3-2002 standard
- Provides the management data input/output (MDIO) interface for PHY device configuration and management with configurable clock frequencies
- Supports both full-duplex and half-duplex operations
- Programmable frame length to support Standard or Jumbo Ethernet frames with sizes up to 16 KB
- Supports a variety of flexible address filtering modes
- Separate 32-bit status returned for transmission and reception packets
- Optimization for packet-oriented DMA transfers with frame delimiters
  - Supports linked-list descriptor list structure
  - Descriptor architecture, allowing large blocks of data transfer with minimum CPU intervention; each descriptor can transfer up to 4 KB of data
  - Comprehensive status reporting for normal operation and transfers with errors
- 4 KB TXFIFO for transmission packets and 16 KB RXFIFO for reception packets
- Programmable interrupt options for different operational conditions

### 8.11.2 Block Diagram

The following figure shows the block diagram of EMAC.

**Figure 8-95 EMAC Block Diagram**

### 8.11.3 Functional Description

#### 8.11.3.1 External Signals

The following table describes the pin mapping of EMAC.

**Table 8-34 EMAC Pin Mapping**

Pin Name	RGMII	RMII
RGMII-RXD3	RXD3	/
RGMII-RXD2	RXD2	/
RGMII-RXD1/RMII-RXD1	RXD1	RXD1
RGMII-RXD0/RMII-RXD0	RXD0	RXD0
RGMII-RXCK	RXCK	/
RGMII-RXCTRL/RMII-CRS-DV	RXCTRL	CRS-DV
RGMII-TXD3	TXD3	/
RGMII-TXD2	TXD2	/
RGMII-TXD1/RMII-TXD1	TXD1	TXD1
RGMII-TXD0/RMII-TXD0	TXD0	TXD0
RGMII-TXCK/RMII-TXCK	TXCK	TXCK
RGMII-TXCTRL/RMII-TXEN	TXCTRL	TXEN
RGMII-CLKIN/ RMII-RXER	CLKIN	RXER
MDC	MDC	MDC
MDIO	MDIO	MDIO

Pin Name	RGMII	RMII
EPHY-25M	EPHY-25M	EPHY-25M

The following table describes the pin list of RGMII.

**Table 8-35 EMAC RGMII Pin List**

Pin Name	Description	Type
RGMII-TXD[3:0]	EMAC RGMII transmit data	O
RGMII-TXCTRL	EMAC RGMII transmit control	O
RGMII-TXCK	EMAC RGMII transmit clock	O
RGMII-RXD[3:0]	EMAC RGMII receive data	I
RGMII-RXCTRL	EMAC RGMII receive control	I
RGMII-RXCK	EMAC RGMII receive clock	I
RGMII-CKIN	EMAC RGMII 125M reference clock input	I
MDC	EMAC management data clock	O
MDIO	EMAC management data input output	I/O
EPHY-25M	25 MHz output for EMAC PHY	O

The following table describes the pin list of RMII.

**Table 8-36 EMAC RMII Pin List**

Pin Name	Description	Type
RMII-TXD[1:0]	EMAC RMII transmit data	O
RMII-TXEN	EMAC RMII transmit enable	O
RMII-TXCK	EMAC RMII transmit clock	I
RMII-RXD[1:0]	EMAC RMII receive data	I
RMII-CRS-DV	EMAC RMII receive data valid	I
RMII-RXER	EMAC RMII receive error	I
MDC	EMAC management data clock	O
MDIO	EMAC management data input output	I/O
EPHY-25M	25 MHz output for EMAC PHY	O

### 8.11.3.2 Clock Characteristics

The following table describes the clock of EMAC.

**Table 8-37 EMAC Clock Characteristics**

Clock Name	Description	Type
RGMII0-TXCK/RMII0-TXCK	In RGMII mode, output 2.5 MHz/25 MHz/125 MHz. In RMII mode, input 5 MHz/50 MHz.	O/I
RGMII0-RXCK	In RGMII mode, input 2.5 MHz/25 MHz/125 MHz. In RMII mode, no input.	I
RGMII0-CLKIN	In RGMII mode, input 125M Reference Clock. In RMII mode, no clock.	I

### 8.11.3.3 Typical Application

**Figure 8-96 EMAC Typical Application**

### 8.11.3.4 EMAC RX/TX Descriptor

The internal DMA of EMAC transfers data between host memory and internal RX/TX FIFO by a linked list of descriptors. Each descriptor consists of four words and contains some necessary information to transfer TX and RX frames. The following figure shows the descriptor list structure. The address of each descriptor must be 32-bit aligned.

**Figure 8-97 EMAC RX/TX Descriptor List**

### 8.11.3.5 TX Descriptor

#### 1st Word of TX Descriptor

Bits	Description
31	TX_DESC_CTL When set, the current descriptor can be used by DMA. This bit is cleared by DMA when the whole frame is transmitted or all data in the buffer of the current descriptor are transmitted.
30:17	Reserved
16	TX_HEADER_ERR When set, the checksum of the header for the transmitted frame is wrong.
15	Reserved
14	TX_LENGTH_ERR When set, the length of the transmitted frame is wrong.
13	Reserved
12	TX_PAYLOAD_ERR When set, the checksum of the payload for the transmitted frame is wrong.
11	Reserved
10	TX_CRS_ERR When set, the carrier is lost during transmission.
9	TX_COL_ERR_0 When set, the frame is aborted because of a collision after the contention period.
8	TX_COL_ERR_1 When set, the frame is aborted because of too many collisions.
7	Reserved

<b>Bits</b>	<b>Description</b>
6:3	TX_COL_CNT The number of collisions before transmission.
2	TX_DEFER_ERR When set, the frame is aborted because of too much deferral.
1	TX_UNDERFLOW_ERR When set, the frame is aborted because of the TX FIFO underflow error.
0	TX_DEFER When set in Half-Duplex mode, the EMAC defers the frame transmission.

**2nd Word of TX Descriptor**

<b>Bits</b>	<b>Description</b>
31	TX_INT_CTL When it is set and the current frame has been transmitted, the TX_INT in Interrupt Status Register will be set.
30	LAST_DESC When it is set, the current descriptor is the last one of the current frame.
29	FIR_DESC When it is set, the current descriptor is the first one of the current frame.
28:27	CHECKSUM_CTL These bits control to insert checksum in the transmit frame.
26	CRC_CTL When it is set, the CRC field is not transmitted.
25:11	Reserved
10:0	BUF_SIZE The size of the buffer specified by the current descriptor.

**3rd Word of TX Descriptor**

<b>Bits</b>	<b>Description</b>
31:0	BUF_ADDR The address of the buffer specified by the current descriptor.

**4th Word of TX Descriptor**

<b>Bits</b>	<b>Description</b>
31:0	NEXT_DESC_ADDR The address of the next descriptor. It must be 32-bit aligned.

**8.11.3.6 RX Descriptor****1st Word of RX Descriptor**

<b>Bits</b>	<b>Description</b>
31	RX_DESC_CTL When it is set, the current descriptor can be used by DMA. This bit is cleared by DMA when the complete frame is received or the buffer of the current descriptor is full.
30	RX_DAF_FAIL When it is set, the current frame does not pass the DA filter.
29:16	RX_FRM_LEN When LAST_DESC is not set and no error bit is set, this field is the length of received data for the current frame. When LAST_DESC is set, RX_OVERFLOW_ERR and RX_NO_ENOUGH_BUF_ERR are not set, this field is the length of the received frame.
15	Reserved
14	RX_NO_ENOUGH_BUF_ERR When it is set, the current frame is clipped because of no enough buffer.
13	RX_SAF_FAIL When it is set, the current frame does not pass the SA filter.
12	Reserved
11	RX_OVERFLOW_ERR When it is set, a buffer overflow error occurred and the current frame is wrong.
10	Reserved
9	FIR_DESC When it is set, the current descriptor is the first descriptor of the current frame.
8	LAST_DESC When it is set, the current descriptor is the last descriptor of the current frame.
7	RX_HEADER_ERR When it is set, the checksum of the frame header is wrong.

<b>Bits</b>	<b>Description</b>
6	RX_COL_ERR When it is set, there is a late collision during the reception in half-duplex mode.
5	Reserved
4	RX_LENGTH_ERR When it is set, the length of the current frame is wrong.
3	RX_PHY_ERR When it is set, the receive error signal from PHY is asserted during the reception.
2	Reserved
1	RX_CRC_ERR When it is set, the CRC field of the received frame is wrong.
0	RX_PAYLOAD_ERR When it is set, the checksum or length of the payload for the received frame is wrong.

**2nd Word of RX Descriptor**

<b>Bits</b>	<b>Description</b>
31	RX_INT_CTL When it is set and a frame has been received, the RX_INT will not be set.
30:11	Reserved
10:0	BUF_SIZE The size of the buffer is specified by the current descriptor.

**3rd Word of RX Descriptor**

<b>Bits</b>	<b>Description</b>
31:0	BUF_ADDR The address of the buffer specified by the current descriptor.

**4th Word of RX Descriptor**

<b>Bits</b>	<b>Description</b>
31:0	NEXT_DESC_ADDR The address of the next descriptor. This field must be 32-bit aligned.

## 8.11.4 Programming Guidelines

### 8.11.4.1 EMAC System Configuration

Perform the following steps:

- Step 1** Write 0 to [EMAC\\_BGR\\_REG](#)[bit16] to assert the module reset.
- Step 2** Write 1 to [EMAC\\_BGR\\_REG](#)[bit16] to deassert the module reset.
- Step 3** Write 1 to [EMAC\\_BGR\\_REG](#)[bit0] to enable the bus clock of the module.
- Step 4** Configure the pin interfaces of EMAC by setting GPIO module.
- Step 5** Configure [EMAC\\_EPHY\\_CLK\\_REG0](#) to set the transmission clock source of RGMII/RMII.

For RGMII RXCLK/CLK125M:

In RGMII mode, in addition to the configuration of the transmission clock source, it is generally necessary to adjust the timing by configuring the transmission clock delay, reception clock delay, transmission clock reverse, reception clock reverse.

- Write 0 to the bit[13] and write 1 to the bit[2] to select the RGMII interface.
- If selecting RXCLK as the clock source of RGMII, write 2 to the bit[1:0]; if selecting CLK125M as the clock source of RGMII, write 1 to the bit[1:0].
- Write 0 to the bit[3], write 0 to the bit[4], write 31 to the bit[9:5], and write 7 to the bit[12:10] to transmit the reception sequence adjustment.

For RMII TXCLK:

- Write 1 to the bit[13] and write 0 to the bit[2] to select the RMII interface.
- Write 0 to the bit[0] to select TXCLK as the clock source of RMII.

The configuration value of [EMAC\\_EPHY\\_CLK\\_REG0](#) can refer to the following table.

Table 8-38 EMAC\_EPHY\_CLK\_REG0 Configuration Value

EMAC_EPHY_CLK_REG0	PHY_SEL	RMII_EN	ETXDC	ERXDC	ERXIE	ETXIE	RMII/RGMII	ETCS
	Bit15	Bit13	Bit[12:10]	Bit[9:5]	Bit4	Bit3	Bit2	Bit[1:0]
RGMII	0	0	7	31	0	0	1	1/2
RMII	0	1	0	0	0	0	0	0

### 8.11.4.2 EMAC Initialization

- Step 1** Write 1 to [EMAC\\_BASIC\\_CTL1](#)[bit0] to perform the software reset.

- Step 2** Write 1 to EMAC\_BASIC\_CTL1[bit1] to set the DMA priority of TX/RX.
- Step 3** Configure EMAC\_TX\_CTL1 and EMAC\_RX\_CTL1 to set the configuration of DMA TX and DMA RX.
- Step 4** Configure EMAC\_INT\_EN to set the corresponding interrupts and shield the needless interrupts.
- Step 5** Configure EMAC\_TX\_DMA\_LIST and EMAC\_RX\_DMA\_LIST to set the first address of the TX descriptor and the RX descriptor, respectively.
- Step 6** Configure EMAC\_TX\_CTL0 and EMAC\_RX\_CTL0 to set the TX and RX parameters. Configure EMAC\_BASIC\_CTL0 to set the speed, duplex mode, loopback configuration. (If enabled the auto-negotiation, the configuration is performed as a result of the negotiation)
- Step 7** Configure EMAC\_RX\_FRM\_FLT to set the RX frame filter.
- Step 8** Configure EMAC\_TX\_FLOW\_CTL and EMAC\_RX\_CTL0 to set the control mechanism of TX and RX.
- Step 9** Clear all interrupt flags.
- Step 10** Write 1 to EMAC\_TX\_CTL0[bit31] and write 1 to EMAC\_RX\_CTL0[bit31] to enable the TX and RX functions.

### 8.11.5 Register List

Module Name	Base Address
EMAC	0x04500000

Register Name	Offset	Description
EMAC_BASIC_CTL0	0x0000	EMAC Basic Control Register0
EMAC_BASIC_CTL1	0x0004	EMAC Basic Control Register1
EMAC_INT_STA	0x0008	EMAC Interrupt Status Register
EMAC_INT_EN	0x000C	EMAC Interrupt Enable Register
EMAC_TX_CTL0	0x0010	EMAC Transmit Control Register0
EMAC_TX_CTL1	0x0014	EMAC Transmit Control Register1
EMAC_TX_FLOW_CTL	0x001C	EMAC Transmit Flow Control Register
EMAC_TX_DMA_DESC_LIST	0x0020	EMAC Transmit Descriptor List Address Register
EMAC_RX_CTL0	0x0024	EMAC Receive Control Register0
EMAC_RX_CTL1	0x0028	EMAC Receive Control Register1
EMAC_RX_DMA_DESC_LIST	0x0034	EMAC Receive Descriptor List Address Register
EMAC_RX_FRM_FLT	0x0038	EMAC Receive Frame Filter Register
EMAC_RX_HASH0	0x0040	EMAC Hash Table Register0

Register Name	Offset	Description
EMAC_RX_HASH1	0x0044	EMAC Hash Table Register1
EMAC_MII_CMD	0x0048	EMAC Management Interface Command Register
EMAC_MII_DATA	0x004C	EMAC Management Interface Data Register
EMAC_ADDR_HIGH0	0x0050	EMAC MAC Address High Register0
EMAC_ADDR_LOW0	0x0054	EMAC MAC Address Low Register0
EMAC_ADDR_HIGN	0x0050+0x08*N (N=1-7)	EMAC MAC Address High Register N (N=1-7)
EMAC_ADDR_LOWN	0x0054+0x08*N (N=1-7)	EMAC MAC Address Low Register N (N=1-7)
EMAC_TX_DMA_STA	0x00B0	EMAC Transmit DMA Status Register
EMAC_TX_CUR_DESC	0x00B4	EMAC Current Transmit Descriptor Register
EMAC_TX_CUR_BUF	0x00B8	EMAC Current Transmit Buffer Address Register
EMAC_RX_DMA_STA	0x00C0	EMAC Receive DMA Status Register
EMAC_RX_CUR_DESC	0x00C4	EMAC Current Receive Descriptor Register
EMAC_RX_CUR_BUF	0x00C8	EMAC Current Receive Buffer Address Register
EMAC_RGMII_STA	0x00D0	EMAC RGMII Status Register

## 8.11.6 Register Description

### 8.11.6.1 0x0000 EMAC Basic Control Register0 (Default Value: 0x0000\_0000)

Offset: 0x0000			Register Name: EMAC_BASIC_CTL0
Bit	Read/Write	Default/Hex	Description
31:4	/	/	/
3:2	R/W	0x0	SPEED EMAC Working Speed 00: 1000 Mbit/s 01: Reserved 10: 10 Mbit/s 11: 100 Mbit/s
1	R/W	0x0	LOOPBACK EMAC Loopback Mode For Test 0: Disable 1: Enable

Offset: 0x0000			Register Name: EMAC_BASIC_CTL0
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	DUPLEX EMAC Transfer Mode 0: Half-duplex 1: Full-duplex

#### 8.11.6.2 0x0004 EMAC Basic Control Register1 (Default Value: 0x0800\_0000)

Offset: 0x0004			Register Name: EMAC_BASIC_CTL1
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29:24	R/W	0x8	BURST_LEN The burst length of RX and TX DMA transfer.
23:2	/	/	/
1	R/W	0x0	RX_TX_PRI RX TX DMA Priority 0: Same priority 1: RX priority is over TX
0	R/W	0x0	SOFT_RST Soft Reset all Registers and Logic 0: No valid 1: Reset All clock inputs must be valid before soft reset. This bit is cleared internally when the reset operation is completed fully. Before writing any register, this bit should read a 0.

#### 8.11.6.3 0x0008 EMAC Interrupt Status Register (Default Value: 0x4000\_0000)

Offset: 0x0008			Register Name: EMAC_INT_STA
Bit	Read/Write	Default/Hex	Description
31:17	R	0x2000	Reserved
16	R/W1C	0x0	RGMII_LINK_STA_P RMII Link Status Changed Interrupt Pending 0: No Pending 1: Pending Write '1' to clear it.

Offset: 0x0008			Register Name: EMAC_INT_STA
Bit	Read/Write	Default/Hex	Description
15:14	/	/	/
13	R/W1C	0x0	<p>RX_EARLY_P RX DMA Filled First Data Buffer of the Receive Frame Interrupt Pending 0: No Pending 1: Pending Write '1' to clear it.</p>
12	R/W1C	0x0	<p>RX_OVERFLOW_P RX FIFO Overflow Error Interrupt Pending 0: No Pending 1: Pending Write '1' to clear it.</p>
11	R/W1C	0x0	<p>RX_TIMEOUT_P RX Timeout Interrupt Pending 0: No Pending 1: Pending Write '1' to clear it. When this bit is asserted, the length of the received frame is greater than 2048 bytes (10240 when JUMBO_FRM_EN is set)</p>
10	R/W1C	0x0	<p>RX_DMA_STOPPED_P When this bit asserted, the RX DMA FSM is stopped.</p>
9	R/W1C	0x0	<p>RX_BUF_UA_P RX Buffer UA Interrupt Pending 0: No Pending 1: Pending Write '1' to clear it. When this bit is asserted, the RX DMA cannot acquire the next RX descriptor and RX DMA FSM is suspended. The ownership of the next RX descriptor should be changed to RX DMA. The RX DMA FSM will resume when the RX_DMA_START is written or the next receive frame is coming.</p>
8	R/W1C	0x0	<p>RX_P Frame RX Completed Interrupt Pending 0: No Pending 1: Pending Write '1' to clear it. When this bit is asserted, a frame reception is completed. The RX DMA FSM remains running.</p>
7:6	/	/	/

Offset: 0x0008			Register Name: EMAC_INT_STA
Bit	Read/Write	Default/Hex	Description
5	R/W1C	0x0	<p>TX_EARLY_P</p> <p>Total interrupt pending which the frame is transmitted to FIFO</p> <p>0: No Pending 1: Pending Write '1' to clear it.</p>
4	R/W1C	0x0	<p>TX_UNDERFLOW_P</p> <p>TX FIFO Underflow Interrupt Pending</p> <p>0: No Pending 1: Pending Write '1' to clear it.</p>
3	R/W1C	0x0	<p>TX_TIMEOUT_P</p> <p>Transmitter Timeout Interrupt Pending</p> <p>0: No Pending 1: Pending Write '1' to clear it.</p>
2	R/W1C	0x0	<p>TX_BUF_UA_P</p> <p>TX Buffer UA Interrupt Pending</p> <p>0: No Pending 1: Pending When this asserted, the TX DMA can not acquire the next TX descriptor and the TX DMA FSM is suspended. The ownership of the next TX descriptor should be changed to TX DMA. The TX DMA FSM will resume when writing to TX_DMA_START bit.</p>
1	R/W1C	0x0	<p>TX_DMA_STOPPED_P</p> <p>Transmission DMA Stopped Interrupt Pending</p> <p>0: No Pending 1: Pending Write '1' to clear it.</p>
0	R/W1C	0x0	<p>TX_P</p> <p>Frame Transmission Interrupt Pending</p> <p>0: No Pending 1: Pending Write '1' to clear it.</p>

**8.11.6.4 0x000C EMAC Interrupt Enable Register (Default Value: 0x0000\_0000)**

Offset: 0x000C			Register Name: EMAC_INT_EN
Bit	Read/Write	Default/Hex	Description
31:14	/	/	/
13	R/W	0x0	<b>RX_EARLY_INT_EN</b> Early Receive Interrupt 0: Disable 1: Enable
12	R/W	0x0	<b>RX_OVERFLOW_INT_EN</b> Receive Overflow Interrupt 0: Disable 1: Enable
11	R/W	0x0	<b>RX_TIMEOUT_INT_EN</b> Receive Timeout Interrupt 0: Disable 1: Enable
10	R/W	0x0	<b>RX_DMA_STOPPED_INT_EN</b> Receive DMA FSM Stopped Interrupt 0: Disable 1: Enable
9	R/W	0x0	<b>RX_BUF_UA_INT_EN</b> Receive Buffer Unavailable Interrupt 0: Disable 1: Enable
8	R/W	0x0	<b>RX_INT_EN</b> Receive Interrupt 0: Disable 1: Enable
7:6	/	/	/
5	R/W	0x0	<b>TX_EARLY_INT_EN</b> Early Transmit Interrupt 0: Disable 1: Enable
4	R/W	0x0	<b>TX_UNDERFLOW_INT_EN</b> Transmit Underflow Interrupt 0: Disable 1: Enable

Offset: 0x000C			Register Name: EMAC_INT_EN
Bit	Read/Write	Default/Hex	Description
3	R/W	0x0	TX_TIMEOUT_INT_EN Transmit Timeout Interrupt 0: Disable 1: Enable
2	R/W	0x0	TX_BUF_UA_INT_EN Transmit Buffer Available Interrupt 0: Disable 1: Enable
1	R/W	0x0	TX_DMA_STOPPED_INT_EN Transmit DMA FSM Stopped Interrupt 0: Disable 1: Enable
0	R/W	0x0	TX_INT_EN Transmit Interrupt 0: Disable 1: Enable

#### 8.11.6.5 0x0010 EMAC Transmit Control Register0 (Default Value: 0x0000\_0000)

Offset: 0x0010			Register Name: EMAC_TX_CTL0
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	TX_EN Enable Transmitter 0: Disable 1: Enable When disabled, the transmission will continue until the current transmission finishes.
30	R/W	0x0	TX_FRM_LEN_CTL Frame Transmit Length Control 0: Up to 2,048 bytes (JUMBO_FRM_EN==0) Up to 10,240 bytes (JUMBO_FRM_EN==1) 1: Up to 16,384 bytes Any bytes after that is cut off.
29:0	/	/	/

**8.11.6.6 0x0014 EMAC Transmit Control Register1 (Default Value: 0x0000\_0000)**

Offset: 0x0014			Register Name: EMAC_TX_CTL1
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<p><b>TX_DMA_START</b> Transmit DMA FSM Start 0: No valid 1: Start It is cleared internally and always read a 0.</p>
30	R/W	0x0	<p><b>TX_DMA_EN</b> 0: Stop TX DMA after the completion of current frame transmission 1: Start and run TX DMA</p>
29:11	/	/	/
10:8	R/W	0x0	<p><b>TX_TH</b> Threshold value of TX DMA FIFO When TX_MD is 0, the transmission starts when the frame size in TX DMA FIFO is greater than the threshold. In addition, the full frames with a length less than the threshold are transferred automatically. 000: 64 001: 128 010: 192 011: 256 Others: Reserved</p>
7:2	/	/	/
1	R/W	0x0	<p><b>TX_MD</b> Transmission Mode 0: TX starts after the TX DMA FIFO bytes is greater than the TX_TH 1: TX starts after the TX DMA FIFO is located a full frame</p>
0	R/WAC	0x0	<p><b>FLUSH_TX_FIFO</b> Flush the data in the TX FIFO 0: Enable 1: Disable</p>

#### 8.11.6.7 0x001C EMAC Transmit Flow Control Register (Default Value: 0x0000\_0000)

Offset: 0x001C			Register Name: EMAC_TX_FLOW_CTL
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<p><b>TX_FLOW_CTL_STA</b></p> <p>This bit indicates a pause frame transmission is in progress. When the configuration of flow control is ready, set this bit to transmit a pause frame in full-duplex mode or activate the backpressure function. After the transmission is completed, this bit will be cleared automatically. Before writing TX_FLOW_CTRL register, this bit must be read as 0.</p>
30:22	/	/	/
21:20	R/W	0x0	<p><b>TX_PAUSE_FRM_SLOT</b></p> <p>The threshold of the pause timer at which the input flow control signal is checked for automatic re-transmission of the pause frame. The threshold values should be always less than PAUSE_TIME.</p>
19:4	R/W	0x0	<p><b>PAUSE_TIME</b></p> <p>The pause time field in the transmitted control frame.</p>
3:2	/	/	/
1	R/W	0x0	<p><b>ZQP_FRM_EN</b></p> <p>0: Disable 1: Enable</p> <p>When set, enable the functionality to generate the Zero-Quanta Pause control frame.</p>
0	R/W	0x0	<p><b>TX_FLOW_CTL_EN</b></p> <p>TX Flow Control Enable</p> <p>0: Disable 1: Enable</p> <p>When set, enable flow control operation to transmit pause frames in full-duplex mode, or enable the back-pressure operation in half-duplex mode.</p>

#### 8.11.6.8 0x0020 EMAC Transmit DMA Descriptor List Address Register (Default Value: 0x0000\_0000)

Offset: 0x0020			Register Name: EMAC_TX_DMA_LIST
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<p><b>TX_DESC_LIST</b></p> <p>The base address of the transmission descriptor list</p> <p>It must be 32-bit aligned.</p>

### 8.11.6.9 0x0024 EMAC Receive Control Register0 (Default Value: 0x0000\_0000)

Offset: 0x0024			Register Name: EMAC_RX_CTL0
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<p>RX_EN Enable Receiver 0: Disable receiver after current reception 1: Enable</p>
30	R/W	0x0	<p>RX_FRM_LEN_CTL Frame Receive Length Control 0: Up to 2048 bytes (JUMBO_FRM_EN==0) Up to 10240 bytes (JUMBO_FRM_EN==1) 1: Up to 16384 bytes Any bytes after that is cut off.</p>
29	R/W	0x0	<p>JUMBO_FRM_EN Jumbo Frame Enable 0: Disable 1: Enable Jumbo frames of 9018 bytes without reporting a giant</p>
28	R/W	0x0	<p>STRIP_FCS When set, strip the Pad/FCS field on received frames only when the length of field value is less than or equal to 1500 bytes.</p>
27	R/W	0x0	<p>CHECK_CRC Check CRC Enable 0: Disable 1: Calculate CRC and check the IPv4 Header Checksum</p>
26:18	/	/	/
17	R/W	0x0	<p>RX_PAUSE_FRM_MD 0: Only detect multicast pause frame specified in the 802.3x standard. 1: In addition to detect multicast pause frame specified in the 802.3x standard, also detect unicast pause frame with the address specified in MAC Address 0 High Register and MAC address 0 Low Register.</p>
16	R/W	0x0	<p>RX_FLOW_CTL_EN When set, enable the functionality that decodes the received pause frame and disable its transmitter for a specified time by pause frame.</p>
15:0	/	/	/

### 8.11.6.10 0x0028 EMAC Receive Control Register1 (Default Value: 0x0000\_0000)

Offset: 0x0028			Register Name: EMAC_RX_CTL1
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<p>RX_DMA_START</p> <p>When set, the RX DMA will work. It is cleared internally and always read a 0.</p>
30	R/W	0x0	<p>RX_DMA_EN</p> <p>Receive DMA Enable</p> <p>0: Stop RX DMA after finishing the received current frame</p> <p>1: Start and run RX DMA</p>
29:25	/	/	/
24	R/W	0x0	<p>RX_FIFO_FLOW_CTL</p> <p>Receive FIFO Flow Control Enable</p> <p>0: Disable</p> <p>1: Enable, base on RX_FLOW_CTL_TH_DEACT and RX_FLOW_CTL_TH_ACT</p>
23:22	R/W	0x0	<p>RX_FLOW_CTL_TH_DEACT</p> <p>Threshold for Deactivating Flow Control</p> <p>00: Full minus 1 KB</p> <p>01: Full minus 2 KB</p> <p>10: Full minus 3 KB</p> <p>11: Full minus 4 KB</p> <p>Valid in both half-duplex mode and full-duplex mode.</p>
21:20	R/W	0x0	<p>RX_FLOW_CTL_TH_ACT</p> <p>Threshold for Activating Flow Control</p> <p>00: Full minus 1 KB</p> <p>01: Full minus 2 KB</p> <p>10: Full minus 3 KB</p> <p>11: Full minus 4 KB</p> <p>Valid in both half-duplex mode and full-duplex mode.</p>
19:6	/	/	/

Offset: 0x0028			Register Name: EMAC_RX_CTL1
Bit	Read/Write	Default/Hex	Description
5:4	R/W	0x0	<p>RX_TH Threshold for RX DMA FIFO Start 00: 64 01: 32 10: 96 11: 128</p> <p>Only valid when RX_MD == 0, the full frames with a length less than the threshold are transferred automatically.</p>
3	R/W	0x0	<p>RX_ERR_FRM 0: RX DMA drops frames with error 1: RX DMA forwards frames with error</p>
2	R/W	0x0	<p>RX_RUNT_FRM When the bit is set to 1, it indicates forward undersized frames with no error and length less than 64 bytes.</p>
1	R/W	0x0	<p>RX_MD Receive Mode 0: RX starts to read after the RX DMA FIFO byte is greater than RX_TH 1: RX starts to read after the RX DMA FIFO is located a full frame</p>
0	R/W	0x0	<p>FLUSH_RX_FRM Flush Receive Frames 0: Enable when the receive descriptors/buffers are unavailable 1: Disable</p>

#### 8.11.6.11 0x0034 EMAC Receive DMA Descriptor List Address Register (Default Value: 0x0000\_0000)

Offset: 0x0034			Register Name: EMAC_RX_DMA_LIST
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<p>RX_DESC_LIST The base address of the received descriptor list It must be 32-bit aligned.</p>

**8.11.6.12 0x0038 EMAC Receive Frame Filter Register (Default Value: 0x0000\_0000)**

Offset: 0x0038			Register Name: EMAC_RX_FRM_FLT
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	DIS_ADDR_FILTER Disable Address Filter 0: Enable 1: Disable
30:18	/	/	/
17	R/W	0x0	DIS_BROADCAST Disable Receive Broadcast Frames 0: Receive 1: Drop
16	R/W	0x0	RX_ALL_MULTICAST Receive All Multicast Frames Filter 0: Filter according to HASH_MULTICAST 1: Receive all
15:14	/	/	/
13:12	R/W	0x0	CTL_FRM_FILTER Receive Control Frames Filter 00: Drop all control frames 01: Drop all control frames 10: Receive all control frames 11: Receive all control frames when passing the address filter
11:10	/	/	/
9	R/W	0x0	HASH_MULTICAST Filter Multicast Frames Set 0: By comparing the DA field in DA MAC address registers 1: According to the hash table
8	R/W	0x0	HASH_UNICAST Filter Unicast Frames Set 0: By comparing the DA field in DA MAC address registers 1: According to the hash table
7	/	/	/

Offset: 0x0038			Register Name: EMAC_RX_FRM_FLT
Bit	Read/Write	Default/Hex	Description
6	R/W	0x0	<p>SA_FILTER_EN Receive SA Filter Enable 0: Receive frames and update the result of SA filter 1: Update the result of the SA filter. In addition, if the SA field of the received frame does not match the values in SA MAC address registers, drop this frame.</p>
5	R/W	0x0	<p>SA_INV_FILTER Receive SA Invert Filter Set 0: Pass frames whose SA field matches SA MAC address registers 1: Pass frames whose SA field does not match SA MAC address registers</p>
4	R/W	0x0	<p>DA_INV_FILTER 0: Normal filtering of frames is performed 1: Filter both unicast and multicast frames by comparing DA field in inverse filtering mode</p>
3:2	/	/	/
1	R/W	0x0	<p>FLT_MD 0: If the HASH_MULTICAST or HASH_UNICAST is set, the frame is passed only when it matches the Hash filter 1: Receive the frame when it passes the address register filter or the hash filter (set by HASH_MULTICAST or HASH_UNICAST)</p>
0	R/W	0x0	<p>RX_ALL Receive All Frame 0: Receive the frames that pass the SA/DA address filter 1: Receive all frames and update the result of address filter (pass or fail) in the receive status word</p>

#### 8.11.6.13 0x0040 EMAC Receive Hash Table Register0 (Default Value: 0x0000\_0000)

Offset: 0x0040			Register Name: EMAC_RX_HASH0
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<p>HASH_TAB0 The upper 32 bits of Hash table for the received frame filter.</p>

**8.11.6.14 0x0044 EMAC Receive Hash Table Register1 (Default Value: 0x0000\_0000)**

Offset: 0x0044			Register Name: EMAC_RX_HASH1
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	HASH_TAB1 The lower 32 bits of Hash table for the received frame filter.

**8.11.6.15 0x0048 EMAC MII Command Register (Default Value: 0x0000\_0000)**

Offset: 0x0048			Register Name: EMAC_MII_CMD
Bit	Read/Write	Default/Hex	Description
31:23	/	/	/
22:20	R/W	0x0	MDC_DIV_RATIO_M MDC Clock Divider Ratio The MDC Clock is divided from the AHB clock. 000: 16 001: 32 010: 64 011: 128 Others: Reserved
19:17	/	/	/
16:12	R/W	0x0	PHY_ADDR PHY Address
11:9	/	/	/
8:4	R/W	0x0	PHY_REG_ADDR PHY Register Address
3:2	/	/	/
1	R/W	0x0	MII_WR MII Write and Read 0: Read 1: Write
0	R/WAC	0x0	MII_BUSY MII Status 0: Writing 0 is no valid, and reading 0 indicates the read/write operation is finished 1: Writing 1 starts the read/write operation, and reading 1 indicates busy.

**8.11.6.16 0x004C EMAC MII Data Register (Default Value: 0x0000\_0000)**

Offset: 0x004C			Register Name: EMAC_MII_DATA
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0	MII_DATA Write to or read from the register in the selected PHY.

**8.11.6.17 0x0050 EMAC MAC Address High Register0 (Default Value: 0x0000\_FFFF)**

Offset: 0x0050			Register Name: EMAC_ADDR_HIGH0
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xFFFF	MAC_ADDR_HIGH0 The upper 16 bits of the 1st MAC address.

**8.11.6.18 0x0054 EMAC MAC Address Low Register0 (Default Value: 0xFFFF\_FFFF)**

Offset: 0x0054			Register Name: EMAC_ADDR_LOW0
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0xFFFFFFFF	MAC_ADDR_LOW0 The lower 32 bits of 1st MAC address.

**8.11.6.19 0x0050+0x08\*N EMAC MAC Address High Register N (Default Value: 0x0000\_FFFF)**

Offset: 0x0050+0x08*N (N=1~7)			Register Name: EMAC_ADDR_HIGHN
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	MAC_ADDR_CTL MAC Address Valid 0: Not valid 1: Valid
30	R/W	0x0	MAC_ADDR_TYPE MAC Address Type 0: Used to compare with the destination address of the received frame 1: Used to compare with the source address of the received frame

Offset: 0x0050+0x08*N (N=1~7)			Register Name: EMAC_ADDR_HIGHN
Bit	Read/Write	Default/Hex	Description
29:24	R/W	0x0	<p>MAC_ADDR_BYTE_CTL MAC Address Byte Control Mask</p> <p>The lower bit of mask controls the lower byte of the MAC address. When the bit of mask is 1, do not compare the corresponding byte.</p>
23:16	/	/	/
15:0	R/W	0xFFFF	<p>MAC_ADDR_HIGH The upper 16 bits of the MAC address.</p>

#### 8.11.6.20 0x0054+0x08\*N EMAC MAC Address Low Register N (Default Value: 0x0000\_0000)

Offset: 0x0054+0x08*N (N=1~7)			Register Name: EMAC_ADDR_LOWN
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<p>MAC_ADDR_LOWN The lower 32 bits of MAC address N (N: 1~7).</p>

#### 8.11.6.21 0x00B0 EMAC Transmit DMA Status Register (Default Value: 0x0000\_0000)

Offset: 0x00B0			Register Name: EMAC_TX_DMA_STA
Bit	Read/Write	Default/Hex	Description
31:3	/	/	/
2:0	R	0x0	<p>TX_DMA_STA The State of Transmit DMA FSM</p> <p>000: STOP, when reset or disable TX DMA</p> <p>001: RUN_FETCH_DESC, fetching TX DMA descriptor</p> <p>010: RUN_WAIT_STA, waiting for the status of TX frame</p> <p>011: RUN_TRANS_DATA, passing the frame from host memory to TX DMA FIFO</p> <p>100: Reserved</p> <p>101: Reserved</p> <p>111: RUN_CLOSE_DESC, closing TX descriptor</p> <p>110: SUSPEND, TX descriptor is unavailable or TX DMA FIFO underflow</p>

**8.11.6.22 0x00B4 EMAC Transmit DMA Current Descriptor Register (Default Value: 0x0000\_0000)**

Offset: 0x00B4			Register Name: EMAC_TX_DMA_CUR_DESC
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	TX_DMA_CUR_DESC The address of current transmit descriptor.

**8.11.6.23 0x00B8 EMAC Transmit DMA Current Buffer Address Register (Default Value: 0x0000\_0000)**

Offset: 0x00B8			Register Name: EMAC_TX_DMA_CUR_BUF
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	TX_DMA_CUR_BUF The address of current transmit DMA buffer.

**8.11.6.24 0x00C0 EMAC Receive DMA Status Register (Default Value: 0x0000\_0000)**

Offset: 0x00C0			Register Name: EMAC_RX_DMA_STA
Bit	Read/Write	Default/Hex	Description
31:3	/	/	/
2:0	R	0x0	RX_DMA_STA The State of RX DMA FSM 000: STOP, when reset or disable RX DMA 001: RUN_FETCH_DESC, fetching RX DMA descriptor 010: Reserved 011: RUN_WAIT_FRM, waiting for the frame 100: SUSPEND, RX descriptor is unavailable 101: RUN_CLOSE_DESC, closing RX descriptor 110: Reserved 111: RUN_TRANS_DATA, passing the frame from host memory to RX DMA FIFO

**8.11.6.25 0x00C4 EMAC Receive DMA Current Descriptor Register (Default Value: 0x0000\_0000)**

Offset: 0x00C4			Register Name: EMAC_RX_DMA_CUR_DESC
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	RX_DMA_CUR_DESC The address of current receive descriptor

**8.11.6.26 0x00C8 EMAC Receive DMA Current Buffer Address Register (Default Value: 0x0000\_0000)**

Offset: 0x00C8			Register Name: EMAC_RX_DMA_CUR_BUF
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	RX_DMA_CUR_BUF The address of current receive DMA buffer

**8.11.6.27 0x00D0 EMAC RGMII Status Register (Default Value: 0x0000\_0000)**

Offset: 0x00D0			Register Name: EMAC_RGMII_STA
Bit	Read/Write	Default/Hex	Description
31:4	/	/	/
3	R	0x0	RGMII_LINK The link status of the RGMII interface 0: Down 1: Up
2:1	R	0x0	RGMII_LINK_SPD The link speed of the RGMII interface 00: 2.5 MHz 01: 25 MHz 10: 125 MHz 11: Reserved
0	R	0x0	RGMII_LINK_MD The link mode of the RGMII interface 0: Half-Duplex 1: Full-Duplex

## 8.12 CIR Receiver

### 8.12.1 Overview

The Consumer Infrared (CIR) receiver captures pulse from the IR Receiver module and uses the Run-Length Code (RLC) to encode the pulse.

The CIR receiver has the following features:

- Supports CIR remote control receiver
- Supports NEC IR protocol
- 64x8 bits RX FIFO for data buffer
- Programmable RX FIFO thresholds
- Supports interrupt
- Sample clock up to 1 MHz

### 8.12.2 Block Diagram

**Figure 8-98 CIR Receiver Block Diagram**

The CIR receiver samples the input signal on the programmable frequency and records these samples into RX FIFO when one CIR signal is found on the air. The CIR receiver uses Run-Length Code (RLC) to encode pulse width. The encoded data is buffered in 64 levels and 8-bit width RX FIFO; the MSB bit is used to record the polarity of the receiving CIR signal, the rest 7 bits are used for the length of RLC. The maximum length of the RLC is 128. If the duration of one level (high or low level) is more than 128, another byte is used.

### 8.12.3 Functional Description

#### 8.12.3.1 External Signals

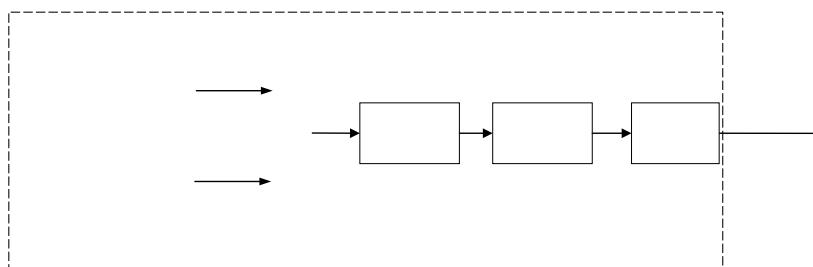
The following table describes the external signals of CIR Receiver.

**Table 8-39 CIR Receiver External Signals**

Signal	Description	Type
IR-RX	Consumer Infrared Receiver	I

#### 8.12.3.2 Clock Sources

**Figure 8-99 CIR Receiver Clock**



#### 8.12.3.3 Typical Application

**Figure 8-100 CIR Receiver Application Diagram**

#### 8.12.3.4 NEC Protocol Format

**Figure 8-101 NEC Protocol**

The CIR receiver module is a timer with a capture function.

When CIR\_RX signals satisfy the Active Threshold (ATHR), the CIR receiver can start to capture. In the process, the signal is ignored if the pulse width of the signal is less than NTHR. When CIR\_RX signals satisfy ITHR (Idle Threshold), the capture process is stopped and the Receiver Packet End interrupt is generated, then the Receiver Packet End Flag is asserted.

In a capture process, every effective pulse is buffered to FIFO in bytes according to the form of the Run-Length Code. The MSB bit of a byte is the polarity of pulse, and the rest 7 bits is pulse width by taking Sample Clock as a basic unit. This is the code form of the RLC-Byte. When the level changes or the pulse width counter overflows, the RLC-Byte is buffered to FIFO. The CIR\_RX module receives the infrared signals transmitted by the infrared remote control, the software decodes the signals.

#### 8.12.3.5 Operating Mode

##### Sample Clock

**Figure 8-102 Logical '0' and Logical '1' of NEC Protocol**

For NEC protocol, a logical "1" takes 2.25 ms (560 us+1680 us) to transmit, while a logical "0" is only half of that, being 1.12 ms (560 us+560 us).

For example, if the sample clock is 31.25 kHz, a sample cycle is 32 us, then 18 sample cycles are 560 us. So the RLC of 560 us low level is 0x12 (b'00010010), the RLC of 560 us high level is 0x92 (b'10010010). Then a logical "1" takes code 0x12 (b'00010010) and code 0xb5 (b'10110101) to transmit, a logical "0" takes code 0x12 and code 0x92 to transmit.

### Active Threshold (ATHR)

When the CIR receiver is in Idle state, if the electrical level of the CIR\_RX signal changes (positive jump or negative jump), and the duration reaches this threshold, then the CIR receiver takes the starting of the signal as a lead code, and the CIR receiver turns into an active state and starts to capture CIR\_RX signals.

**Figure 8-103 ATHR Definition**

### Idle Threshold (ITHR)

If the electrical level of CIR\_RX signals has no change, and the duration reaches this threshold, then the CIR receiver enters into Idle state and ends this capture.

**Figure 8-104 ITHR Definition**

### Noise Threshold (NTHR)

In the capture process, the pulse is ignored if the pulse width is less than the Noise Threshold.

**Figure 8-105 NTHR Definition**

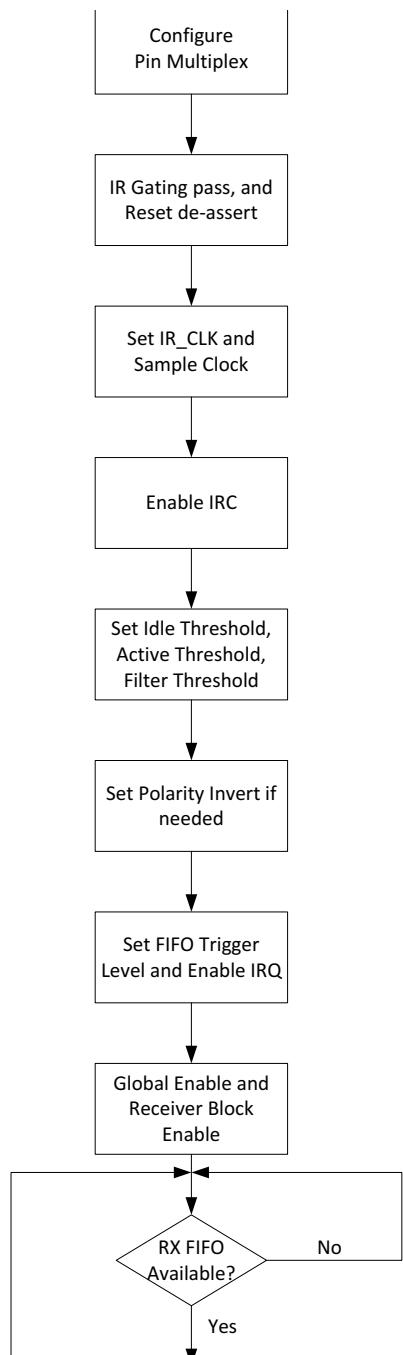
### Active Pulse Accept Mode (APAM)

The APAM is used to fit the type of lead code. If a pulse does not fit the type of lead code, it is not regarded as a lead code even if the pulse width reaches ATHR.

**Figure 8-106 APAM Definition**

#### 8.12.4 Programming Guidelines

Figure 8-107 CIR Receiver Process



### 8.12.5 Register List

Module Name	Base Address
CIR_RX	0x07040000

Register Name	Offset	Description
CIR_CTL	0x0000	CIR Control Register
CIR_RXPCFG	0x0010	CIR Receiver Pulse Configure Register
CIR_RXFIFO	0x0020	CIR Receiver FIFO Register
CIR_RXINT	0x002C	CIR Receiver Interrupt Control Register
CIR_RXSTA	0x0030	CIR Receiver Status Register
CIR_RXCFG	0x0034	CIR Receiver Configure Register

### 8.12.6 Register Description

#### 8.12.6.1 0x0000 CIR Receiver Control Register (Default Value: 0x0000\_0000)

Offset: 0x0000			Register Name: CIR_CTL
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:6	R/W	0x0	APAM Active Pulse Accept Mode 00, 01: Both positive and negative pulses are valid as a leading code 10: Only negative pulse is valid as a leading code 11: Only positive pulse is valid as a leading code
5:4	R/W	0x0	CIR ENABLE 00~10: Reserved 11: CIR mode enable
3:2	/	/	/
1	R/W	0x0	RXEN Receiver Block Enable 0: Disable 1: Enable

Offset: 0x0000			Register Name: CIR_CTL
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	<p>GEN Global Enable</p> <p>A disable on this bit overrides any other block or channel enables and flushes all FIFOs.</p> <p>0: Disable 1: Enable</p>

#### 8.12.6.2 0x0010 CIR Receiver Pulse Configure Register (Default Value: 0x0000\_0004)

Offset: 0x0010			Register Name: CIR_RXPCFG
Bit	Read/Write	Default/Hex	Description
31:3	/	/	/
2	R/W	0x1	<p>RPPI Receiver Pulse Polarity Invert</p> <p>0: Do not invert receiver signal 1: Invert receiver signal</p>
1:0	/	/	/

#### 8.12.6.3 0x0020 CIR Receiver FIFO Register (Default Value: 0x0000\_0000)

Offset: 0x0020			Register Name: CIR_RXFIFO
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R	0x0	RBF Receiver Byte FIFO

#### 8.12.6.4 0x002C CIR Receiver Interrupt Control Register (Default Value: 0x0000\_0000)

Offset: 0x002C			Register Name: CIR_RXINT
Bit	Read/Write	Default/Hex	Description
31:14	/	/	/

Offset: 0x002C			Register Name: CIR_RXINT
Bit	Read/Write	Default/Hex	Description
13:8	R/W	0x0	<p>RAL</p> <p>RX FIFO available received byte level for interrupt and DMA request</p> <p>TRIGGER_LEVEL = RAL + 1</p>
5	R/W	0x0	<p>DRQ_EN</p> <p>RX FIFO DMA Enable</p> <p>0: Disable</p> <p>1: Enable</p> <p>When it is set to '1', the Receiver FIFO DRQ is asserted if reaching RAL. The DRQ is de-asserted when the condition fails.</p>
4	R/W	0x0	<p>RAI_EN</p> <p>RX FIFO Available Interrupt Enable</p> <p>0: Disable</p> <p>1: Enable</p> <p>When it is set to '1', the Receiver FIFO IRQ is asserted if reaching RAL. The IRQ is de-asserted when the condition fails.</p>
3:2	/	/	/
1	R/W	0x0	<p>RPEI_EN</p> <p>Receiver Packet End Interrupt Enable</p> <p>0: Disable</p> <p>1: Enable</p>
0	R/W	0x0	<p>ROI_EN</p> <p>Receiver FIFO Overrun Interrupt Enable</p> <p>0: Disable</p> <p>1: Enable</p>

#### 8.12.6.5 0x0030 CIR Receiver Status Register (Default Value: 0x0000\_0000)

Offset: 0x0030			Register Name: CIR_RXSTA
Bit	Read/Write	Default/Hex	Description
31:15	/	/	/

Offset: 0x0030			Register Name: CIR_RXSTA
Bit	Read/Write	Default/Hex	Description
14:8	R	0x0	<p>RAC</p> <p>RX FIFO Available Counter</p> <p>0: No available data in RX FIFO</p> <p>1: 1-byte available data in RX FIFO</p> <p>2: 2-bytes available data in RX FIFO</p> <p>...</p> <p>64: 64-bytes available data in RX FIFO</p>
7	R	0x0	<p>STAT</p> <p>Status of CIR</p> <p>0: Idle</p> <p>1: Busy</p>
6:5	/	/	/
4	R/W1C	0x0	<p>RA</p> <p>RX FIFO Available</p> <p>0: RX FIFO not available according to its level</p> <p>1: RX FIFO available according to its level</p> <p>Writing 1 clears this bit.</p>
3:2	/	/	/
1	R/W1C	0x0	<p>RPE</p> <p>Receiver Packet End Flag</p> <p>0: STO was not detected. In CIR mode, one CIR symbol is receiving or not detected.</p> <p>1: STO field or packet abort symbol (7'b0000,000 and 8'b0000,0000 for MIR and FIR) is detected. In CIR mode, one CIR symbol is received.</p> <p>Writing 1 clears this bit.</p>
0	R/W1C	0x0	<p>ROI</p> <p>Receiver FIFO Overrun</p> <p>0: Receiver FIFO not overrun</p> <p>1: Receiver FIFO overrun</p> <p>Writing 1 clears this bit.</p>

#### 8.12.6.6 0x0034 CIR Receiver Configure Register (Default Value: 0x0000\_1828)

Offset: 0x0034			Register Name: CIR_RXCFG
Bit	Read/Write	Default/Hex	Description
31:25	/	/	/
24	R/W	0x0	SCS2 Bit2 of Sample Clock Select for CIR This bit is defined by SCS bits below.
23	R/W	0x0	ATHC Active Threshold Control for CIR 0: ATHR in a unit of (Sample Clock) 1: ATHR in a unit of (128*Sample Clocks)
22:16	R/W	0x0	ATHR Active Threshold for CIR These bits control the duration of CIR from the idle to the active state. The duration can be calculated by ((ATHR + 1)*(ATHC? Sample Clock: 128*Sample Clock)).
15:8	R/W	0x18	ITHR Idle Threshold for CIR The Receiver uses it to decide whether the CIR command is received. If there is no CIR signal on the air, the receiver is staying in IDLE status. One active pulse will bring the receiver from IDLE status to Receiving status. After the CIR receiver ends, the inputting signal will keep the specified level (high or low level) for a long time. The receiver can use this idle signal duration to decide that it has received the CIR command. The corresponding flag is asserted. If the corresponding interrupt is enabled, the interrupt line is asserted to the CPU. When the duration of the signal keeps one status (high or low level) for the specified duration ((ITHR + 1)*128 sample_clk), this means that the previous CIR command is finished.

Offset: 0x0034			Register Name: CIR_RXCFG																																				
Bit	Read/Write	Default/Hex	Description																																				
7:2	R/W	0xA	<p>NTHR Noise Threshold for CIR When the duration of the signal pulse (high or low level) is less than NTHR, the pulse is taken as noise and should be discarded by hardware.</p> <p>0: All samples are recorded into RX FIFO 1: If the signal is only one sample duration, it is taken as noise and discarded. 2: If the signal is less than (<math>\leq</math>) two sample duration, it is taken as noise and discarded. ... 61: If the signal is less than (<math>\leq</math>) sixty-one sample duration, it is taken as noise and discarded.</p>																																				
1:0	R/W	0x0	<p>SCS Sample Clock Select for CIR</p> <table border="1"> <thead> <tr> <th>SCS2</th> <th>SCS[1]</th> <th>SCS[0]</th> <th>Sample Clock</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>CIR_CLK/64</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>CIR_CLK/128</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>CIR_CLK/256</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>CIR_CLK/512</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>CIR_CLK</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Reserved</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>Reserved</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Reserved</td> </tr> </tbody> </table>	SCS2	SCS[1]	SCS[0]	Sample Clock	0	0	0	CIR_CLK/64	0	0	1	CIR_CLK/128	0	1	0	CIR_CLK/256	0	1	1	CIR_CLK/512	1	0	0	CIR_CLK	1	0	1	Reserved	1	1	0	Reserved	1	1	1	Reserved
SCS2	SCS[1]	SCS[0]	Sample Clock																																				
0	0	0	CIR_CLK/64																																				
0	0	1	CIR_CLK/128																																				
0	1	0	CIR_CLK/256																																				
0	1	1	CIR_CLK/512																																				
1	0	0	CIR_CLK																																				
1	0	1	Reserved																																				
1	1	0	Reserved																																				
1	1	1	Reserved																																				

## 8.13 CIR Transmitter

### 8.13.1 Overview

The CIR transmitter (CIR\_TX) can transfer arbitrary waves which can be modulated with configurable carrier waves such as 38 kHz.

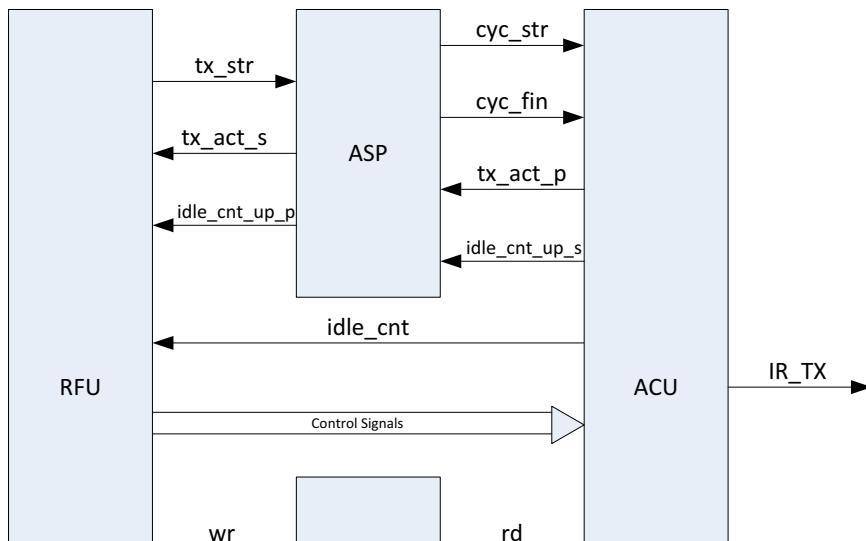
The CIR\_TX has the following features:

- Supports CIR remote control transmitter
- 128 bytes FIFO for data buffer
- Configurable carrier frequency
- Supports Interrupt and DMA
- Supports handshake mode and waiting mode of DMA

### 8.13.2 Block Diagram

The following figure shows a block diagram of the CIR\_TX.

**Figure 8-108 CIR\_TX Block Diagram**



### 8.13.3 Functional Description

#### 8.13.3.1 External Signals

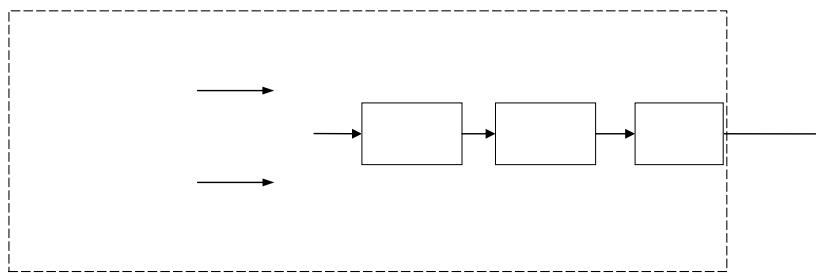
The following table describes the external signals of CIR\_TX.

**Table 8-40 CIR\_TX External Signals**

Signal	Description	Type
CIR-TX	Consumer infrared transmitter	O

#### 8.13.3.2 Clock Sources

**Figure 8-109 CIR\_TX Clock Description**



#### 8.13.3.3 Function Implementation

The CIR\_TX is used to generate a waveform of arbitrary length, arbitrary shape, and no high-speed requirement, and it can change the data into the high-/low-level sequence of a certain length. Every transmitting data is in bytes, the Bit[7] of a byte means whether the level of a transmitting wave is high or low, the Bit[6:0] is the length of this wave. If the current transmitting frequency-division is 1, 0x88 is a high level of 8 cycles, 0x08 is a low level of 8 cycles. If the current transmitting frequency-division is 4, 0x88 is a high level of 32 cycles, 0x08 is a low level of 32 cycles.

The CIR\_TX has two transmission modes: non-cycle transmission, and cycle transmission.

The non-cycle transmission is to transmit all the data in TX\_FIFO until the FIFO is empty.

The cycle transmission is to transmit all the data in TX\_FIFO, after the transmission completion, wait for a certain time to recover the data in TX\_FIFO and then send it until a stop signal is detected. The data recovery in FIFO is implemented by clearing the read pointer.

#### 8.13.3.4 Timing Diagram

The CIR remote control contains many protocols designed by different manufacturers. Here to NEC protocol as an example, the CIR-TX module uses a variable pulse-width modulation technique to encompass the various formats of infrared encoding for remote-control applications. A message is started by a 9 ms AGC burst, which is used to set the gain of the earlier CIR receivers. This AGC burst is then followed by a 4.5 ms space, which is then followed by the address and command.

Bit definition: the logical “1” takes 2.25 ms to transmit, while a logical “0” is only 1.12 ms.

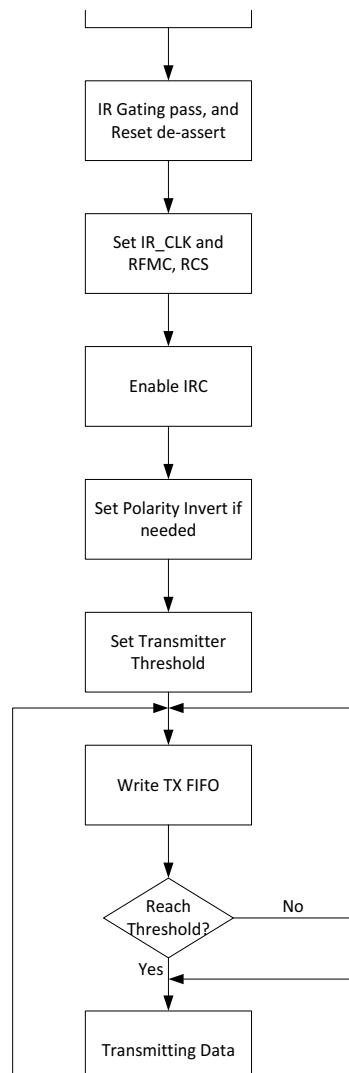
**Figure 8-110 Definitions of Logical “1” and Logical “0”**

Timing for a message:

**Figure 8-111 CIR Message Timing Diagram**

#### 8.13.4 Programming Guidelines

Figure 8-112 CIR Transmitter Process



### 8.13.5 Register List

Module Name	Base Address
CIR_TX	0x02003000

Register Name	Offset	Description
CIR_TGLR	0x0000	CIR Transmit Global Register
CIR_TMCR	0x0004	CIR Transmit Modulation Control Register
CIR_TCR	0x0008	CIR Transmit Control Register
CIR_IDC_H	0x000C	CIR Transmit Idle Duration Threshold High Bit Register
CIR_IDC_L	0x0010	CIR Transmit Idle Duration Threshold Low Bit Register
CIR_TICR_H	0x0014	CIR Transmit Idle Counter High Bit Register
CIR_TICR_L	0x0018	CIR Transmit Idle Counter Low Bit Register
CIR_TEL	0x0020	CIR TX FIFO Empty Level Register
CIR_TXINT	0x0024	CIR Transmit Interrupt Control Register
CIR_TAC	0x0028	CIR Transmit FIFO Available Counter Register
CIR_TXSTA	0x002C	CIR Transmit Status Register
CIR_TXT	0x0030	CIR Transmit Threshold Register
CIR_DMA	0x0034	CIR DMA Control Register
CIR_TXFIFO	0x0080	CIR Transmit FIFO Data Register

### 8.13.6 Register Description

#### 8.13.6.1 0x0000 CIR Transmitter Global Register (Default Value: 0x0000\_0000)

Offset: 0x0000			Register Name: CIR_TGLR
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R/W	0x0	IMS Internal Modulation Select 0: The transmitting signal is not modulated 1: The transmitting signal is modulated internally

Offset: 0x0000			Register Name: CIR_TGLR
Bit	Read/Write	Default/Hex	Description
6:5	R/W	0x0	<p>DRMC</p> <p>Duty ratio of modulated carrier is high level/low level.</p> <p>00: Low level is equal to high level</p> <p>01: Low level is the double of high level</p> <p>10: Low level is the triple of high level</p> <p>11: Reserved</p>
4:3	/	/	/
2	R/W	0x0	<p>TPPI</p> <p>Transmit Pulse Polarity Invert</p> <p>0: Not invert transmit pulse</p> <p>1: Invert transmit pulse</p>
1	R/W	0x0	<p>TR</p> <p>Transmit Reset</p> <p>When this bit is set, the transmitting is reset. The FIFO will be flushed, the TIC filed and the CSS field will be cleared during Transmit Reset. This field will automatically be cleared when the Transmit Reset is finished, and the CIR transmitter will state Idle.</p>
0	R/W	0x0	<p>TXEN</p> <p>Transmit Block Enable</p> <p>0: Disable the CIR Transmitter</p> <p>1: Enable the CIR Transmitter</p>

#### 8.13.6.2 0x0004 CIR Transmitter Modulation Control Register (Default Value: 0x0000\_009E)

Offset: 0x0004			Register Name: CIR_TMCR
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R/W	0x9E	<p>RFMC</p> <p>Reference Frequency of modulated carrier.</p> <p>Reference Frequency of modulated carrier based on a division of a fixed functional clock (FCLK). The range of the modulated carrier is usually 30 kHz to 60 kHz. Most consumer electronics is 38 kHz.</p> <p>The default modulated carrier is 38 kHz when FCLK is 12 MHz.</p> <p>RFMC= FCLK/((N+1)*(DRMC+2)).</p>

#### 8.13.6.3 0x0008 CIR Transmitter Control Register (Default Value: 0x0000\_0000)

Offset: 0x0008			Register Name: CIR_TCR
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R/W	0x0	CSS Cyclical Pulse Start/Stop Control 0: Stop when cleared to '0'. From start to stop, all data in FIFO must be transmitted. 1: Start. Start to transmit when it is set to '1'.
6:4	/	/	/
3:1	R/W	0x0	RCS Reference Clock Select for CIR Transmit The data in TX_FIFO is used to describe the pulse in Run-Length Code. The basic unit of pulse width is Reference Clock. 000: CIR Transmit reference clock is ir_clk 001: CIR Transmit reference clock is ir_clk/2 010: CIR Transmit reference clock is ir_clk/4 011: CIR Transmit reference clock is ir_clk/8 100: CIR Transmit reference clock is ir_clk/64 101: CIR Transmit reference clock is ir_clk/128 110: CIR Transmit reference clock is ir_clk/256 111: CIR Transmit reference clock is ir_clk/512
0	R/W	0x0	TTS Type of the transmission signal 0: The transmitting wave is a single non-cyclical pulse. 1: The transmitting wave is a cyclical short-pulse.

#### 8.13.6.4 0x000C CIR Transmitter Idle Duration Counter High Bit Register (Default Value: 0x0000\_0000)

Offset: 0x000C			Register Name: CIR_IDC_H
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/

Offset: 0x000C			Register Name: CIR_IDC_H
Bit	Read/Write	Default/Hex	Description
3:0	R/W	0x0	<p>IDC_H</p> <p>Idle Duration Counter Threshold (High 4 bits)</p> <p>Idle Duration = 128*IDC*Ts (IDC = 0~4095)</p> <p>It is used in cyclical transmission mode. When all the data in FIFO is transmitted, the signals can be transmitted after a specific time.</p>

#### 8.13.6.5 0x0010 CIR Transmitter Idle Duration Counter Low Bit Register (Default Value: 0x0000\_0000)

Offset: 0x0010			Register Name: CIR_IDC_L
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
3:0	R/W	0x0	<p>IDC_L</p> <p>Idle Duration Counter Threshold (Low 8 bits)</p> <p>Idle Duration = 128*IDC*Ts (IDC = 0~4095)</p> <p>It is used in cyclical transmission mode. When all the data in FIFO is transmitted, the signals can be transmitted after a specific time.</p>

#### 8.13.6.6 0x0014 CIR Transmitter Idle Counter High Bit Register (Default Value: 0x0000\_0000)

Offset: 0x0014			Register Name: CIR_TICR_H
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R	0x0	<p>TIC_H</p> <p>Transmit Idle Counter_H (High 8 bits)</p> <p>It is used to count the idle duration of CIR transmitter by software.</p> <p>Count in 128*Ts (Sample Duration, 1/Fs) when the transmitter is idle, and it should be reset when the transmitter is active.</p> <p>When this counter reaches the maximum value (0xFFFF), it will stop automatically, and should not be cleared to zero.</p>

#### 8.13.6.7 0x0018 CIR Transmitter Idle Counter Low Bit Register (Default Value: 0x0000\_0000)

Offset: 0x0018			Register Name: CIR_TICR_L
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R	0x0	<p>TIC_L Transmit Idle Counter_L (Low 8 bits) It is used to count the idle duration of CIR transmitter by software. Count in 128*Ts (Sample Duration, 1/Fs) when the transmitter is idle, and it should be reset when the transmitter is active. When this counter reaches the maximum value (0xFFFF), it will stop automatically, and should not be cleared to zero.</p>

#### 8.13.6.8 0x0020 CIR Transmitter FIFO Empty Register (Default Value: 0x0000\_0000)

Offset: 0x0020			Register Name: CIR_TEL
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R/W	0x0	<p>TEL TX FIFO empty Level for DRQ and IRQ. TRIGGER_LEVEL = TEL + 1</p>

#### 8.13.6.9 0x0024 CIR Transmitter Interrupt Control Register (Default Value: 0x0000\_0000)

Offset: 0x0024			Register Name: CIR_TXINT
Bit	Read/Write	Default/Hex	Description
31:3	/	/	/
2	R/W	0x0	<p>DRQ_EN TX FIFO DMA Enable 0: Disable 1: Enable When it is set to '1', the TX FIFO DRQ is asserted if the number of the transmitting data in the FIFO is less than the RAL. The DRQ is de-asserted when the condition fails.</p>

Offset: 0x0024			Register Name: CIR_TXINT
Bit	Read/Write	Default/Hex	Description
1	R/W	0x0	<p>TAI_EN TX FIFO Available Interrupt Enable 0: Disable 1: Enable</p>
0	R/W	0x0	<p>TPEI_EN Transmit Packet End Interrupt Enable for Cyclical Pulse 0: Disable 1: Enable</p> <p>TUI_EN Transmitter FIFO Underrun Interrupt Enable for Non-cyclical Pulse 0: Disable 1: Enable</p>

#### 8.13.6.10 0x0028 CIR Transmitter FIFO Available Counter Register (Default Value: 0x0000\_0080)

Offset: 0x0028			Register Name: CIR_TAC
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R	0x80	<p>TAC TX FIFO Available Space Counter 0x00: No available space in TX FIFO 0x01: 1 byte available space in TX FIFO 0x02: 2 bytes available space in TX FIFO ... 0x80: 128 bytes available space in TX FIFO</p>

#### 8.13.6.11 0x002C CIR Transmitter Status Register (Default Value: 0x0000\_0002)

Offset: 0x002C			Register Name: CIR_TXSTA
Bit	Read/Write	Default/Hex	Description
31:4	/	/	/

Offset: 0x002C			Register Name: CIR_TXSTA
Bit	Read/Write	Default/Hex	Description
3	R	0x0	<p>STCT Status of CIR Transmitter 0: Idle 1: Active</p> <p>This bit will automatically set when the controller begins to transmit the data in the FIFO. The “1” will last when the data in the FIFO. It will automatically be cleared to “0” when all data in the FIFO is transmitted.</p> <p>The bit is for debugging.</p> <p>The output Level of Idle state is determined by the level of the last data output.</p>
2	R	0x0	<p>DRQ DMA Request Flag</p> <p>When set to ‘1’, the TX FIFO DRQ is asserted if the number of the transmission data in the FIFO is less than the RAL. The DRQ is de-asserted when the condition fails.</p> <p>This bit is for debugging.</p>
1	R/W	0x1	<p>TAI TX FIFO Available Interrupt Flag 0: TX FIFO not available by its level 1: TX FIFO available by its level</p> <p>Writing 1 clears this bit.</p>
0	R/W	0x0	<p>TPE Transmitter Packet End Flag for Cyclical Pulse 0: Transmissions of address, control and data fields not completed 1: Transmissions of address, control and data fields completed</p> <p>TUR Transmitter FIFO Underrun Flag for Non-cyclical Pulse 0: No transmitter FIFO underrun 1: Transmitter FIFO underrun</p> <p>Writing 1 clears this bit.</p>

#### 8.13.6.12 0x0030 CIR Transmitter Threshold Register (Default Value: 0x0000\_0000)

Offset: 0x0030			Register Name: CIR_TXT
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R/W	0x0	NCTT Non-cyclical Pulse Transmit Threshold The controller will trigger transmitting the data in the FIFO when the data byte number has reached the Transmit Threshold set in this field.

#### 8.13.6.13 0x0034 CIR Transmitter DMA Control Register (Default Value: 0x0000\_00A5)

Offset: 0x0034			Register Name: CIR_DMA_CTL
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R/W	0xA5	DMA Handshake Configuration 0xA5: DMA waiting cycle mode 0xEA: DMA handshake mode

#### 8.13.6.14 0x0080 CIR Transmitter FIFO Data Register (Default Value: 0x0000\_0000)

Offset: 0x0080			Register Name: CIR_TXFIFO
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	W	0x0	TBF Transmit Byte FIFO When the transmission is triggered, the data in the FIFO will be transmitted until the data number is transmitted completely.

# Contents

9 Security System.....	1063
9.1 Crypto Engine.....	1063
9.1.1 Overview .....	1063
9.1.2 Block Diagram .....	1064
9.1.3 Functional Description .....	1065
9.1.4 Programming Guidelines.....	1084
9.1.5 Register List.....	1087
9.1.6 Register Description .....	1088
9.2 Security ID .....	1092

# Figures

Figure 9-1 CE Block Diagram .....	1064
Figure 9-2 DES Encryption and Decryption .....	1065
Figure 9-3 3DES Encryption and Decryption of a 3-key Operation and a 2-key Operation .....	1066
Figure 9-4 ECB Mode Encryption and Decryption.....	1067
Figure 9-5 CBC Mode Encryption and Decryption .....	1068
Figure 9-6 CTR Mode Encryption and Decryption.....	1069
Figure 9-7 CFB Mode Encryption and Decryption.....	1070
Figure 9-8 OFB Mode Encryption and Decryption .....	1071
Figure 9-9 CTS Mode Encryption and Decryption .....	1072
Figure 9-10 Structure of Task Descriptor Chaining .....	1073
Figure 9-11 Word Address of Message.....	1080
Figure 9-12 Byte Order.....	1080
Figure 9-13 Bit Order .....	1080
Figure 9-14 The Storage Method of 32-bit IV .....	1081
Figure 9-15 The Storage Method of 64-bit IV .....	1081
Figure 9-16 Task Request Process.....	1084

# 9 Security System

## 9.1 Crypto Engine

### 9.1.1 Overview

The Crypto Engine (CE) module is one encryption/decryption algorithm accelerator. It supports kinds of symmetric, asymmetric, Hash, and RNG algorithms.

The symmetric algorithm supports data encryption and decryption by following the data encryption standard (DES), 3DES, or advanced encryption standard (AES) algorithms. It can encrypt or decrypt a large amount of data effectively.

The Rivest-Shamir-Adleman (RSA) asymmetric algorithm is used for data encryption/decryption and digital signature verification. It is a public key encryption/decryption algorithm implemented through the modular exponentiation operation.

The Hash algorithm supports data integrity authentication and digital signature. The hash supports MD5, SHA1, SHA224, SHA256, SHA384, SHA512, HMAC-SHA1, and HMAC-SHA256 algorithms.

The RNG algorithm can generate true-random numbers and pseudo-random numbers.

The H133 has two separate CE controllers: one CE is for secure world, the other CE is for non-secure world. The software interface of the controller is simple, only setting interrupt control, task description address, and load tag. The algorithm control information is written in memory by task descriptor, then the CE automatically reads it when executing a request. It supports parallel execution of 4 channels and has an internal DMA controller to transfer data between CE and memory.

The CE has the following features:

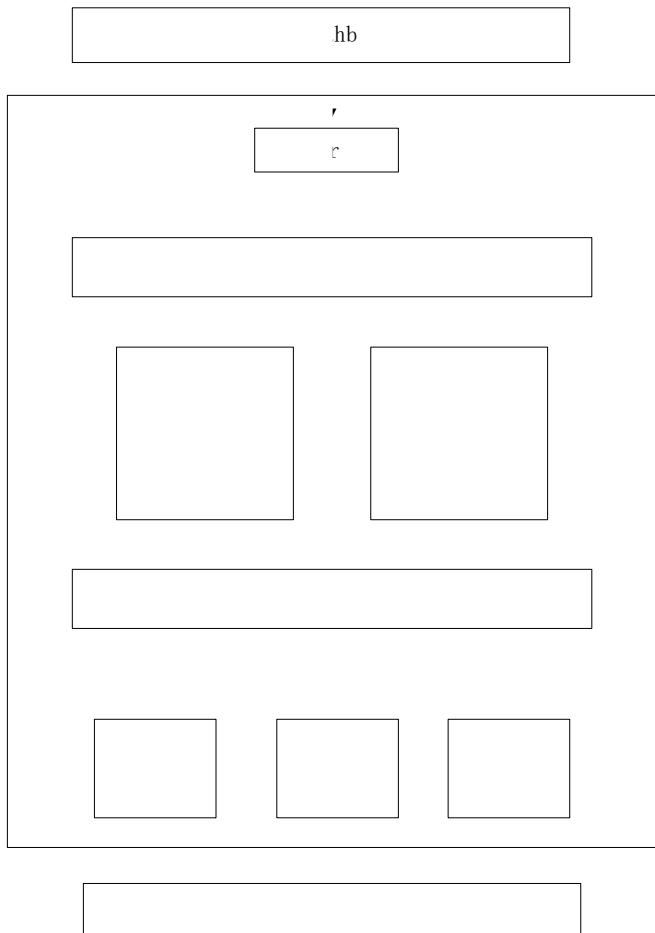
- Symmetrical algorithm: AES, DES, 3DES
- Hash algorithm: MD5, SHA1, SHA224, SHA256, SHA384, SHA512, HMAC-SHA1, HMAC-SHA256
- Asymmetrical algorithm: RSA512/1024/2048-bit
- 160-bit hardware PRNG with 175-bit seed
- 256-bit hardware TRNG
- Electronic codebook (ECB), cipher block chaining (CBC), counter (CTR), cipher text stealing (CTS), 128-output feedback (OFB), 1-/8-/64-/128-cipher feedback (CFB) modes for AES algorithm
- ECB, CBC, CTR modes for DES/3DES algorithm
- 128-, 192-, 256-bit key size for AES algorithms
- 16-, 32-, 64-, 128-bit wide size for AES CTR mode

- 16-, 32-, 64-bit wide size for DES/3DES CTR mode
- One or more blocks mode for MD5/SHA1/SHA224/SHA256/SHA384/SHA512
- Internal DMA controller for data transfer with memory
- Supports secure and non-secure interfaces respectively

### 9.1.2 Block Diagram

The following figure shows the block diagram of Crypto Engine.

**Figure 9-1 CE Block Diagram**

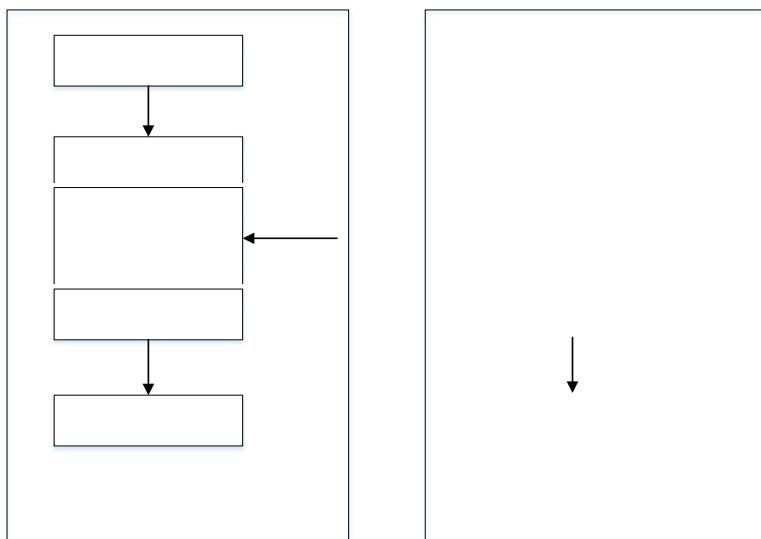


### 9.1.3 Functional Description

#### 9.1.3.1 DES Algorithm

Figure 9-2 shows the DES encryption and decryption operation.

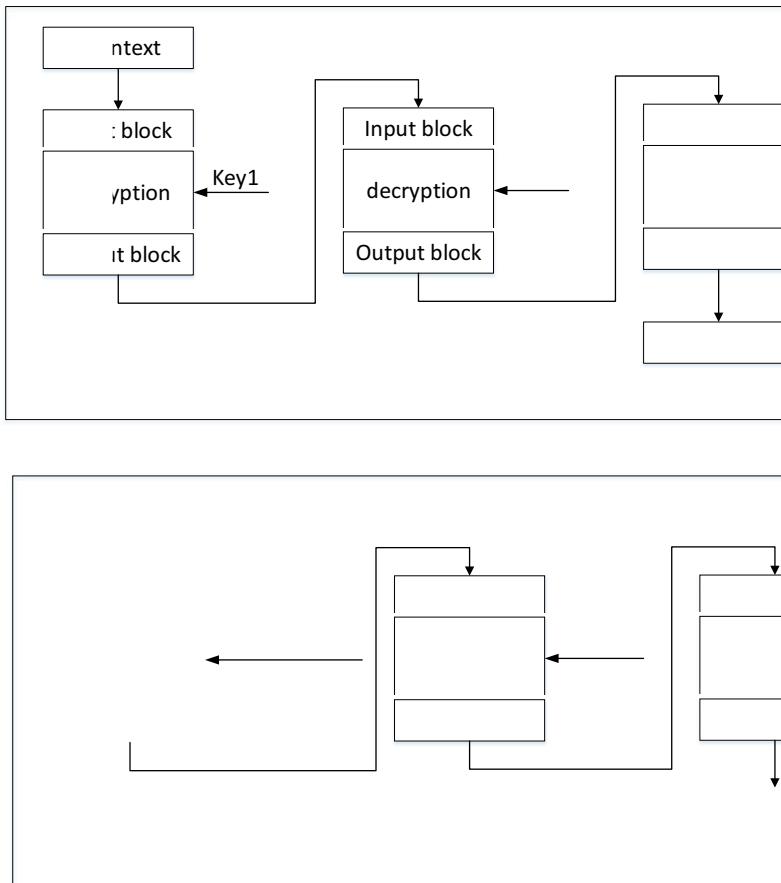
**Figure 9-2 DES Encryption and Decryption**



#### 9.1.3.2 3DES Algorithm

The 3DES algorithm supports both 3-key and 2-key operations. A 2-key operation can be regarded as a simplified 3-key operation. To be specific, key 3 is represented by key 1 in a 2-key operation. Figure 10-3 shows the 3DES encryption and decryption operation of a 3-key operation and a 2-key operation.

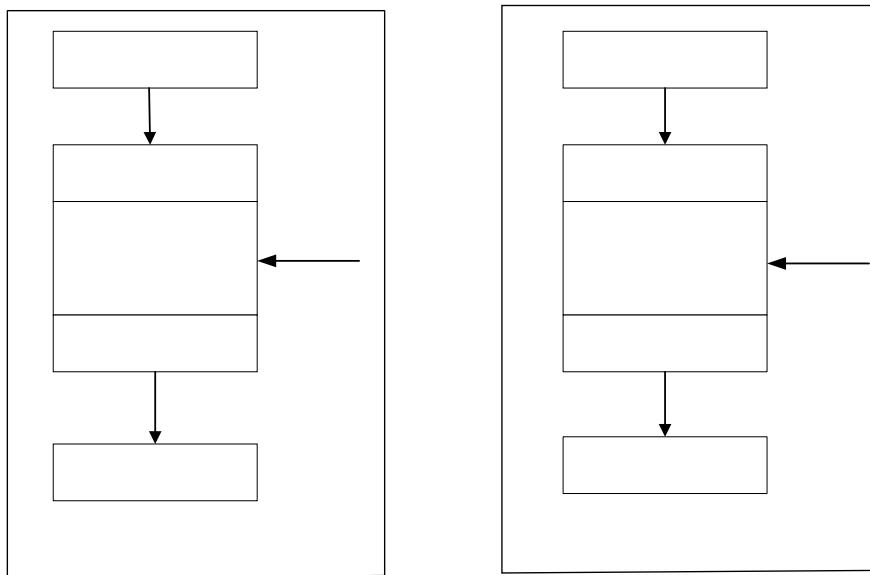
Figure 9-3 3DES Encryption and Decryption of a 3-key Operation and a 2-key Operation



#### 9.1.3.3 ECB Mode

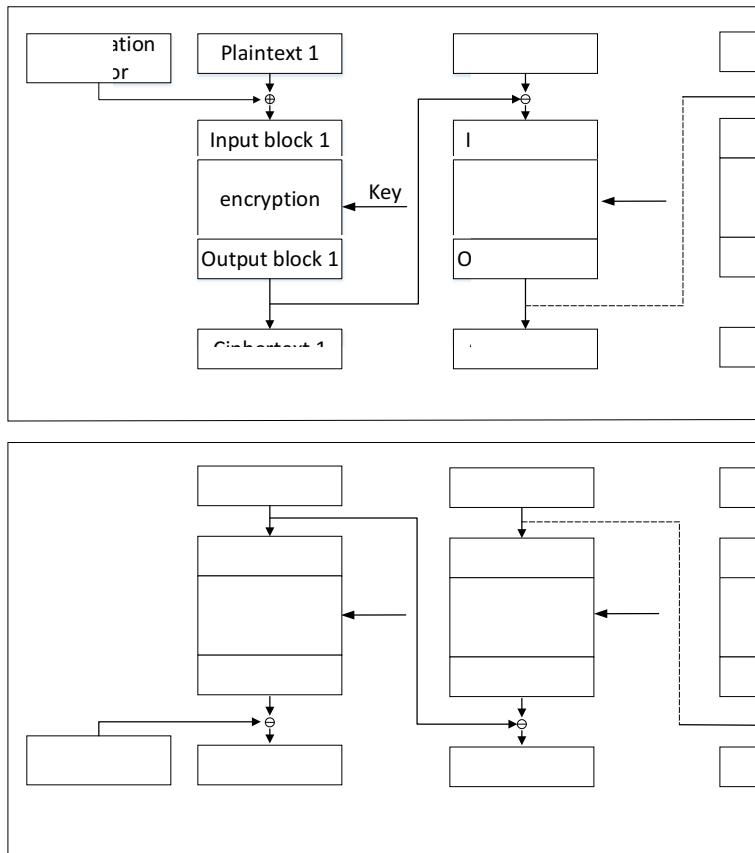
The ECB mode is a confidentiality mode that features, for a given key, the assignment of a fixed ciphertext block to each plaintext block, analogous to the assignment of code words in a codebook.

In ECB mode, encryption and decryption algorithms are directly applied to the block data. The operation of each block is independent, so the plaintext encryption and ciphertext decryption can be performed concurrently.

**Figure 9-4 ECB Mode Encryption and Decryption**

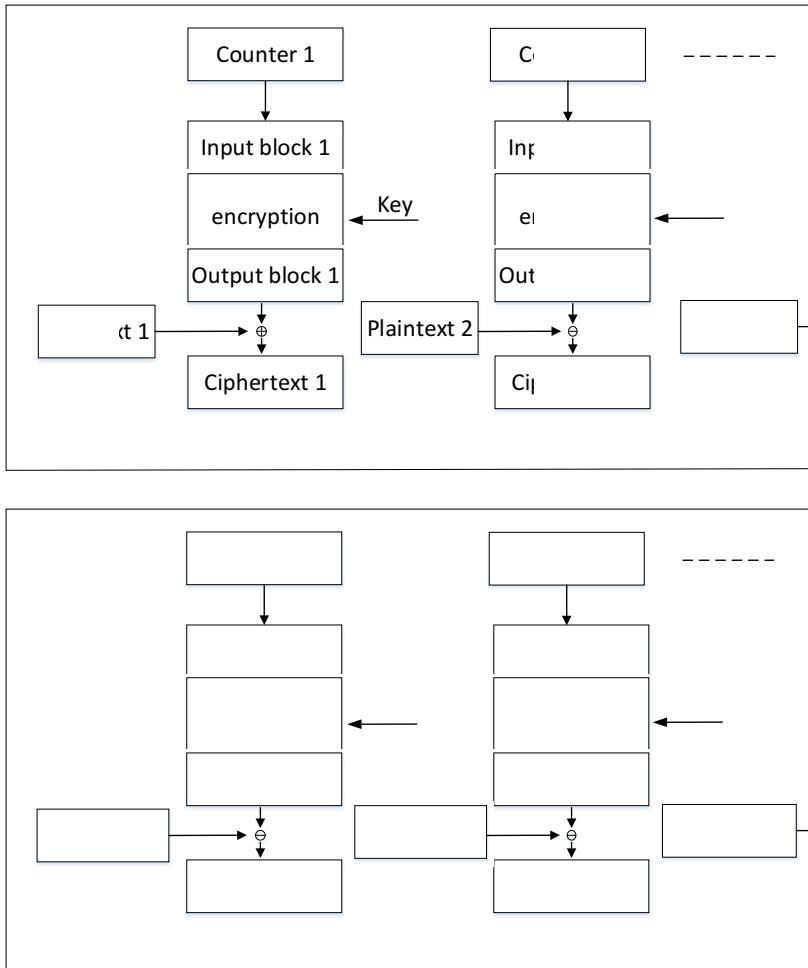
#### 9.1.3.4 CBC Mode

The CBC mode is a confidentiality mode whose encryption process features the combining of the plaintext blocks with the previous ciphertext blocks. The CBC mode requires an initialization vector (IV) to combine with the first plaintext block. The encryption process of each plaintext block is related to the block processing result of the previous ciphertext blocks, so encryption operations cannot be concurrently performed in CBC mode. The decryption operation is independent of output plain text of the previous block, so decryption operations can be performed concurrently.

**Figure 9-5 CBC Mode Encryption and Decryption**

#### 9.1.3.5 CTR Mode

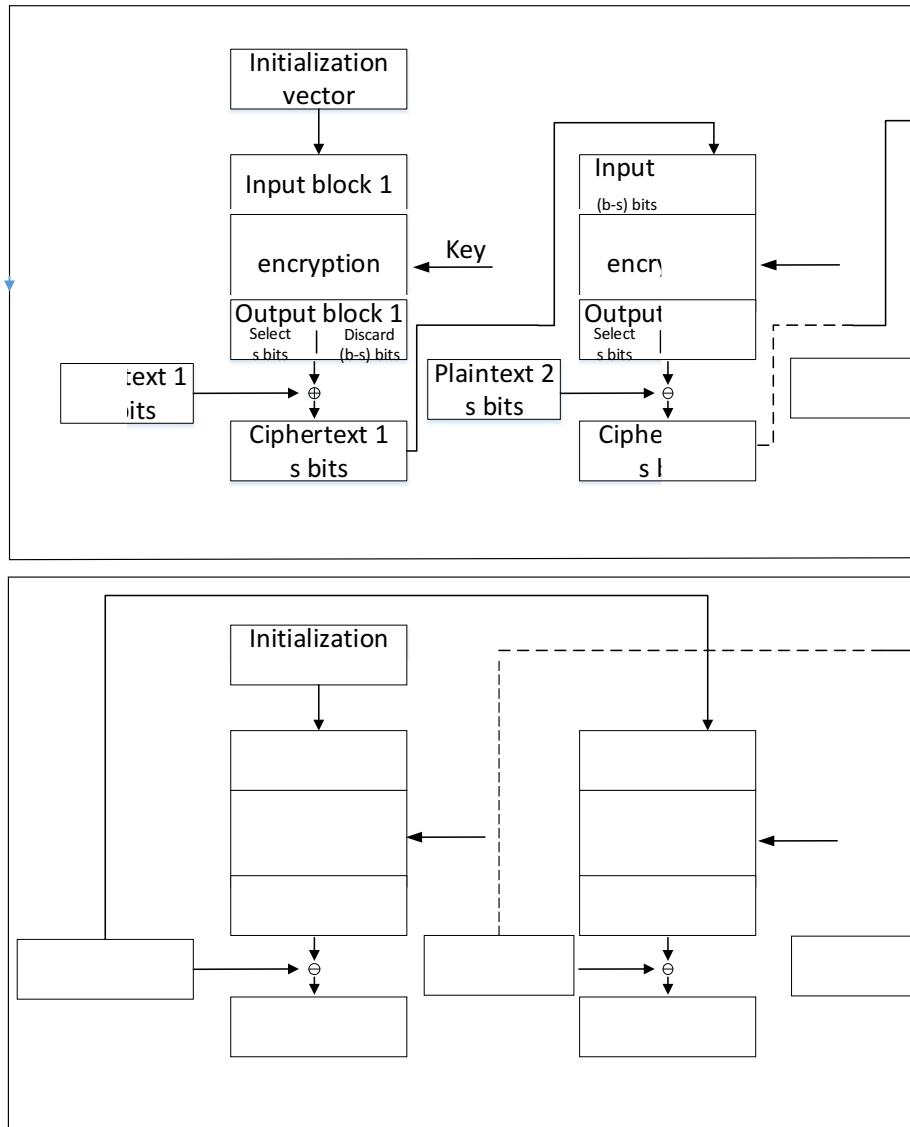
The CTR mode is a confidentiality mode that features the application of the forward cipher to a set of input blocks, called counters, to produce a sequence of output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa. All of the counters must be distinct.

**Figure 9-6 CTR Mode Encryption and Decryption**

#### 9.1.3.6 CFB Mode

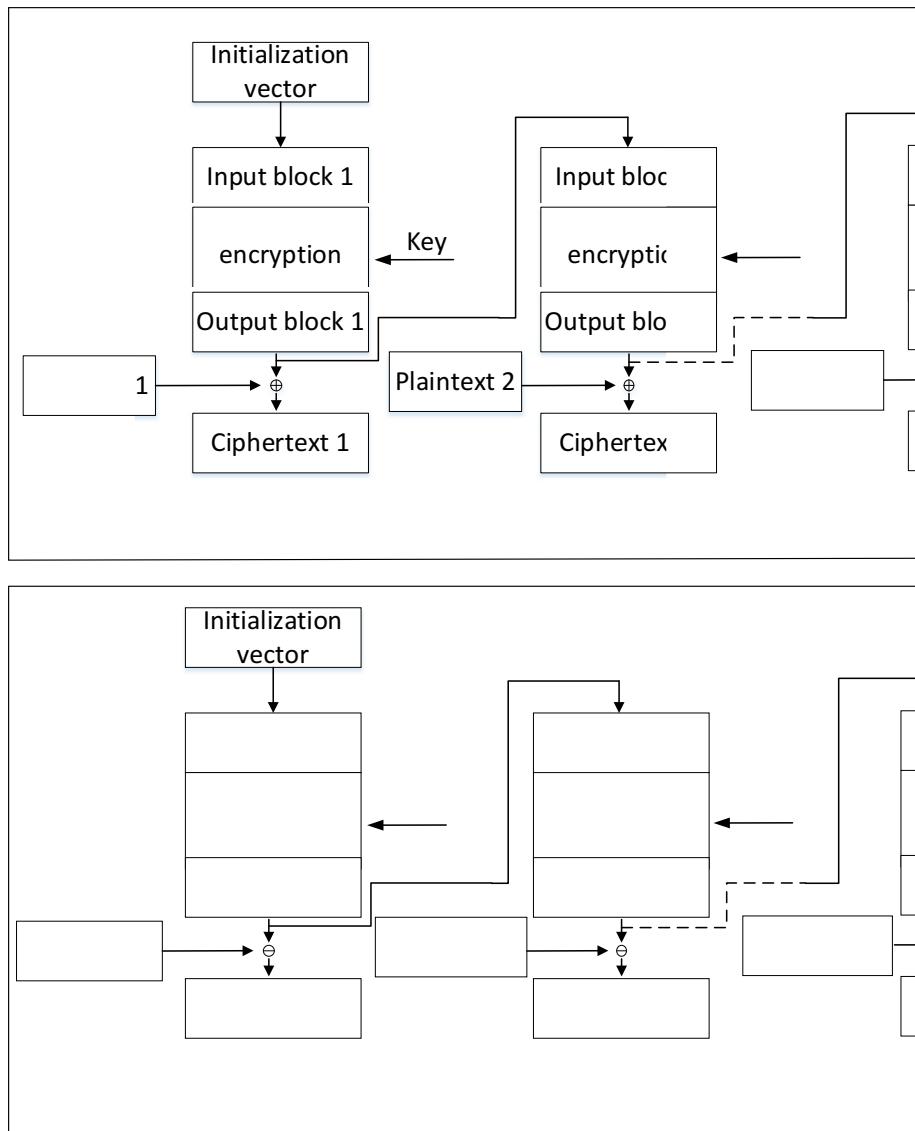
The CFB mode is a confidentiality mode that features the feedback of successive ciphertext segments into the input blocks of the forward cipher to generate output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa. The CFB mode requires an IV as the initial input block, and the forward cipher operation is applied to the IV to produce the first output block. The first ciphertext segment is produced by exclusive-ORing the first plaintext segment with the  $s$  most significant bits of the first output block. The value of  $s$  is 1 bit, 8 bits, 64 bits, or 128 bits.

The following figure shows the  $s$ -bit CFB mode of the AES algorithms.

**Figure 9-7 CFB Mode Encryption and Decryption**

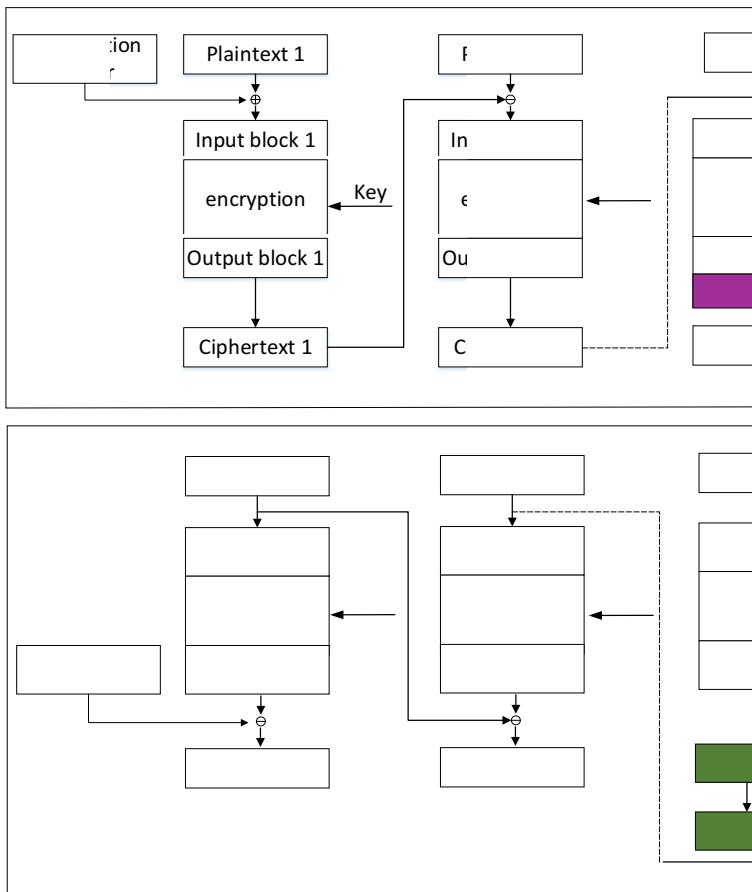
#### 9.1.3.7 OFB Mode

The OFB mode is a confidentiality mode that features the iteration of the forward cipher on an IV to generate a sequence of output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa. If a same key is used, different IVs must be used to ensure operation security.

**Figure 9-8 OFB Mode Encryption and Decryption**

#### 9.1.3.8 CTS Mode

The CTS mode is a confidentiality mode that accepts any plaintext input whose bit length is greater than or equal to the block size but not necessarily a multiple of the block size. Below are the diagrams for CTS encryption and decryption.

**Figure 9-9 CTS Mode Encryption and Decryption**

### 9.1.3.9 HASH Algorithm

The hash algorithms support MD5, SHA1, SHA224, SHA256, SHA384, SHA512, HMAC-SHA1, and HMAC-SHA256. All algorithms are iterative, one-way hash functions that can process a message to produce a condensed representation called a *message digest*. When a message is received, the *message digest* can be used to verify whether the data has changed, that is, to verify its integrity.

The hash algorithm of H133 supports block-aligned total length of the input data (padded by software), that is, a multiple of 64 bytes. The message length after padding by software is used as the configured data length for the hash algorithm.

### 9.1.3.10 RSA Algorithm

The RSA is a public key encryption/decryption algorithm implemented through the modular exponentiation operation.

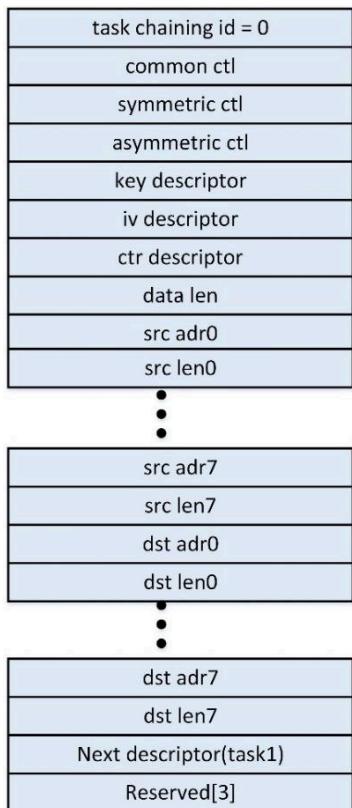
The ciphertext is obtained as follows:  $C = M^E \bmod N$ . The plaintext is obtained as follows:  $M = C^D \bmod N$ .

M indicates the plaintext, C indicates the ciphertext, (N, E) indicates the public key, and (N, D) indicates the private key.

#### 9.1.3.11 Task Descriptor

The software makes request through task descriptor, including algorithm type, algorithm mode, key address, source/destination address and size, and so on. The structure of the task descriptor is as follows.

**Figure 9-10 Structure of Task Descriptor Chaining**



The bit definitions of the task descriptor are as follows.

#### Task ID

Bit	Read/Write	Default/Hex	Description
31:4	/	/	/

Bit	Read/Write	Default/Hex	Description
3:0	R/W	0x0	<p>CHN</p> <p>Task channel ID</p> <p>Indicates which channel the task is running on.</p> <p>It supports 0 to 3.</p>

### Common Control

Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<p>Interrupt enable (IE) for the current task</p> <p>0: disable interrupt</p> <p>1: enable interrupt</p> <p>Represents whether an interrupt signal is generated when the task chain ends at the end of this task.</p> <p>When the last task in a task chain ends, the operation of the task chain will end normally; if a task fails in the middle, the task chain will be aborted abnormally. And it is determined whether to generate an interrupt signal according to the IE configuration of the current task when the current task ends or aborts. Therefore, if you want to use interrupts, it is recommended that not only the IE of the last task of each task chain is configured to 1 to generate the end interrupt of the task chain, but also the IEs of other tasks in this task chain are also configured to 1. The purpose is to generate an interrupt signal once an abnormal error occurs in these tasks and the interrupt is aborted.</p>
30:17	/	/	/
16	R/W	0x0	<p>IV mode</p> <p>IV mode for SHA1/SHA224/SHA256/SHA384/SHA512/MD5 or constants</p> <p>0: use initial constants defined in FIPS-180</p> <p>1: use input iv</p>
15	R/W	0x0	<p>Last HMAC plaintext</p> <p>0: not the last HMAC plaintext package</p> <p>1: the last HMAC plaintext package</p>
14:9	/	/	/

<b>Bit</b>	<b>Read/Write</b>	<b>Default/Hex</b>	<b>Description</b>
8	R/W	0x0	OP DIR Algorithm Operation Direction 0: Encryption 1: Decryption Configure according to the requirements of encryption or decryption.
7	/	/	/
6:0	R/W	0x0	Algorithm Type 0x0: AES 0x1: DES 0x2: Triple DES (3DES) 0x10: MD5 0x11: SHA-1 0x12: SHA-224 0x13: SHA-256 0x14: SHA-384 0x15: SHA-512 0x16: HMAC-SHA1 0x17: HMAC-SHA256 0x20: RSA 0x30: TRNG 0x31: PRNG Others: reserved

### Symmetric Control

<b>Bit</b>	<b>Read/Write</b>	<b>Default/Hex</b>	<b>Description</b>
31:24	/	/	/
23:20	R/W	0x0	KEY_SELECT key select for AES 0000: Select input CE_KEYx (Normal Mode) 0001: Select {SSK} 0010: Select {HUK} 0011: Select {RSSK} 0100-0111: Reserved 1000-1111: Select internal Key n (n from 0 to 7)

<b>Bit</b>	<b>Read/Write</b>	<b>Default/Hex</b>	<b>Description</b>
19:18	R/W	0x0	CFB_WIDTH AES-CFB width 00: CFB1 01: CFB8 10: CFB64 11: CFB128
17	R/W	0x0	PRNG_LD Load new 15 bits key into linear feedback shift register (LFSR) for PRNG. When the algorithm type is PRNG, it is necessary to post-process the random number output by PRNG through the linear shift operation to generate the operand. When the PRNG_LD is configured to 1, use iv_addr[14:0] as the input number for linear shift operation, and do XOR operation between the data and the random number output by PRNG to generate the post-processing result of further operation.
16	R/W	0x0	CTS_LPKG AES CTS last package flag When set to '1', it means this is the last package for AES-CTS mode (the size of the last package is larger than 128 bits).
15:12	/	/	/
11:8	R/W	0x0	ALGORITHM_MODE CE algorithm mode 0000: Electronic Code Book (ECB) mode 0001: Cipher Block Chaining (CBC) mode 0010: Counter (CTR) mode 0011: Cipher Text Stealing (CTS) mode 0100: Output Feedback (OFB) mode 0101: Cipher Feedback (CFB) mode Other: reserved
7:4	/	/	/
3:2	R/W	0x0	CTR WIDTH Counter width for CTR mode 00: 16-bit Counter 01: 32-bit Counter 10: 64-bit Counter 11: 128-bit Counter

Bit	Read/Write	Default/Hex	Description
1:0	R/W	0x0	AES KEY SIZE 00: 128-bit 01: 192-bit 10: 256-bit 11: Reserved

**Asymmetric Control**

Bit	Read/Write	Default/Hex	Description
31	/	/	/
30:28	R/W	0x0	RSA Public Modulus Width 000: 512-bit 001: 1024-bit 010: 2048-bit Other: reserved
27:19	/	/	/
18:16	R/W	0x0	RSA MODE RSA algorithm mode. For modular computation: 000: modular exponent(RSA) 001: modular div 010: modular mul 011: modular inv others: reserved
15:0	/	/	/

**Key Descriptor**

Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	Key Address The address of KEY that needs to be stored.

**IV Descriptor**

Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	IV Address The address of IV that needs to be stored.

**Counter Descriptor**

Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	CTR Data Output Address The address of CTR data output that needs to be stored.

**Data Length**

Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	Data Length Configure the data length of the corresponding segment. The data length size needs to be consistent with dst_data_length (destination data length 0 +... + destination data length 7).  The data length field in the task descriptor has different meanings for different algorithms.  For AES-CTS, the data length field indicates byte numbers of source data, for others indicate word numbers of source data.  For PRNG, the data length should be 5 words aligned.  For TRNG, it should be 8 words aligned.

**Source Address 0~7**

Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	Source Data Address The address of the source data that needs to be stored.

**Source Data Length 0~7**

Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	Source Data Length The length of the source data. Unit: byte

**Destination Address 0~7**

Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	Destination Data Address The address of the destination data that needs to be stored.

**Destination Data Length 0~7**

Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	Destination Data Length The length of the destination data. Unit: byte

**Next Descriptor Address**

Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	Next Task Address The address where the descriptor of the next task in a task-chain is saved. If there is the only task or the last task of a task-chain, the next task address must be 32'h0.

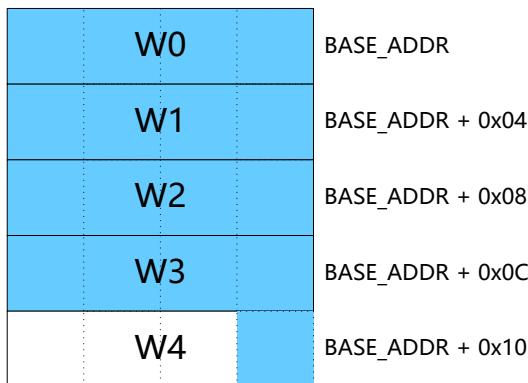
**9.1.3.12 Storing Message**

In the application, a message may not be stored contiguously in the memory, but divided into multiple segments. Or a piece of continuously stored messages can be artificially split into multiple pieces as needs.

Then each segment corresponds to a set of the source address and source length in the descriptor. Multiple segments correspond to groups 0-7 source address/source length in sequence.

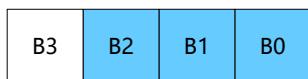
Each task supports up to 8 message segments, and the data volume of each message segment supports up to 4 GWord (AES-CTS is 1 GByte). The total amount of all segments in a task (that is a package) supports up to 4 GWord (AES-CTS is 1 GByte). If a message is divided into multiple packages, all others are required to be whole words; when the last package of AES-CTS is less than one word, 0 needs to be padded, and those less than one word are counted as one word. The following figure shows the address order structure.

**Figure 9-11 Word Address of Message**



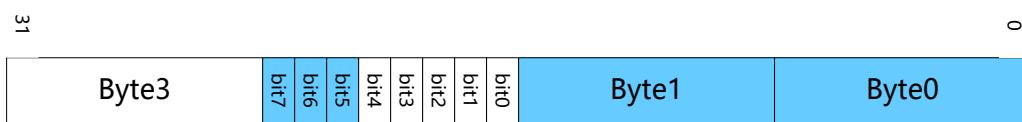
Byte order: low byte first, high byte last. When the data is less than one word, the low byte is filled first. The following figure shows the byte order structure (blue means it is filled by the message).

**Figure 9-12 Byte Order**



Bit order: high bit first, low bit last. When the data is less than one Byte, the high bit is filled first. The following figure shows the bit order structure.

**Figure 9-13 Bit Order**



#### 9.1.3.13 Storing Key

The length of KEY must be an integer multiple of word, refer to section 9.1.3.15 "[Algorithm Length Properties](#)".

### 9.1.3.14 Storing IV

For different algorithms, the length of IV is different. But they are integer multiples of word. To keep the byte order of IV and HASH digest output consistent, the byte order of IV is different from that of the message. For the multi-packet operation, the first address of the digest output result of the previous HASH can be directly configured to the first address of the next IV, and the software does not need to do any processing on the digest.

The following figure shows the storage method of 32-bit IV value.

Figure 9-14 The Storage Method of 32-bit IV

IV0[31:0]	BASE_ADDR
IV1[31:0]	BASE_ADDR + 0x04
.....	.....
IV7[31:0]	BASE_ADDR + 0x1C

The following figure shows the storage method of 64-bit IV value.

Figure 9-15 The Storage Method of 64-bit IV

IV0[63:32]	BASE_ADDR
IV0[31:00]	BASE_ADDR + 0x04
IV1[63:32]	BASE_ADDR + 0x08
IV1[31:00]	BASE_ADDR + 0x0C
.....	.....
IV7[63:32]	BASE_ADDR
IV7[31:00]	BASE_ADDR + 0x3C

### 9.1.3.15 Algorithm Length Properties

The algorithm length has different requirements for different algorithms.

**Table 9-1 Symmetric Algorithm Configuration Properties**

Algorithm	Length Setting				Alignment	Software Padding
	Source Size	Destination Size	KEY	IV		
AES (except CTS)	< 4 GWord	< 4 GWord	AES-128: 4 Word AES-192: 6 Word AES 256: 8 Word	4 Word	Word-aligned	need
AES-CTS	< 1 GByte	< 1 GByte	AES-128: 4 word AES-192: 6 word AES 256: 8 word	4 Word	Word-aligned	need
DES	< 4 GWord	< 4 GWord	2 Word	2 Word	Word-aligned	need
TDES	< 4 GWord	< 4 GWord	6 Word	2 Word	Word-aligned	need

**Table 9-2 Hash Algorithm Configuration Properties**

Algorithm	Length Setting				Alignment	Software Padding
	Source Size	Destination Size	KEY	IV		
MD5	< 4 GWord	4 Word	Fixed to 0	4 Word	Word-aligned	need
SHA-1	< 4 GWord	5 Word	Fixed to 0	5 Word	Word-aligned	need
SHA-224	< 4 GWord	8 Word	Fixed to 0	8 Word	Word-aligned	need
SHA-256	< 4 GWord	8 Word	Fixed to 0	8 Word	Word-aligned	need
SHA-384	< 4 GWord	16 Word	Fixed to 0	16 Word	Word-aligned	need
SHA-512	< 4 GWord	16 Word	Fixed to 0	16 Word	Word-aligned	need
HMAC-SHA1	< 4 GWord	5 Word	16 Word	5 Word	Word-aligned	need
HMAC-SHA256	< 4 GWord	8 Word	16 Word	8 Word	Word-aligned	need

**Table 9-3 RNG Configuration Properties**

Algorithm	Length Setting				Alignment	Software Padding
	Source Size	Destination Size	KEY	IV		
TRNG	< 4 GWord	< 4 GWord	Fixed to 0	4 Word	Word-aligned	need
PRNG	< 4 GWord	< 4 GWord	6 Word	4 Word	Word-aligned	need

**Table 9-4 Asymmetric Algorithm Configuration Properties**

Algorithm	Length Setting				Alignment	Software Padding
	Source Size	Destination Size	KEY	IV		
RSA512	16 Word	16 Word	16 Word	Not use IV	Word-aligned	need
RSA1024	32 Word	32 Word	32 Word	Not use IV	Word-aligned	need
RSA2048	64 Word	64 Word	64 Word	Not use IV	Word-aligned	need

#### 9.1.3.16 Security Operation

When the CPU issues request to the CE module, the CE module will save the secure mode of CPU. When executing this request, this state bit works as a access flag for the inner and system resources. For access to SID module through the AHB bus, only the secure mode can succeed, or else these keys will be read to 0 or cannot write. When issuing MBUS read and write requests, the CE will use send this secure mode bit to BUS, so secure requests can access secure and non-secure space, but non-secure requests only can access non-secure space.

#### 9.1.3.17 Error Detection

The CE module includes error detection for task configuration, data computing error, and authentication invalid. When the algorithm type in task descriptor is read into the CE module, the CE will check whether this type is supported through checking algorithm type field in common control. If the type value is out of scope, the CE will issue interrupt signal and set error state. Each type has certain input and output data size. After getting a task descriptor, the input size and output size configuration will be checked to avoid size error. If the size configuration is wrong, the CE will issue interrupt signal and set error state.

### 9.1.3.18 Clock Requirement

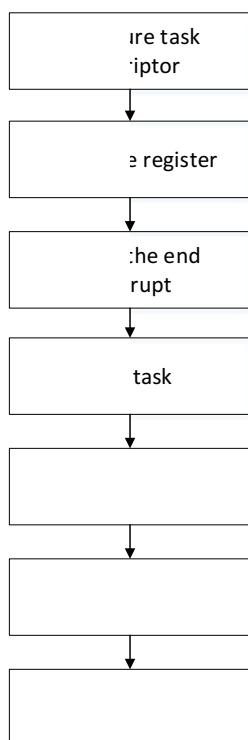
Clock Name	Description	Requirement
hclk	AHB bus clock	24 MHz – 200 MHz
mclk	MBUS clock	24 MHz – 400 MHz
ce_clk	CE work clock	24 MHz – 400 MHz

### 9.1.4 Programming Guidelines

#### 9.1.4.1 Symmetrical/Asymmetrical/Hash/RNG Algorithm Operation Process

The following figure shows the process of an algorithm operation.

Figure 9-16 Task Request Process



**Step 1** The software should configure a task descriptor in memory, including the related fields in the descriptor. The channel id corresponds to one channel in CE. According to algorithm type, the software should set the fields in common control, symmetric control, asymmetric control, then provide key/iv/ctr address and the data length of this task. The source and destination address and size are set based on the upper application. If there is another task concatenating after this task, then set its descriptor address at the next descriptor field. For more details for task descriptor, see section 10.1.4.2, section 10.1.4.3 and section 10.1.4.4.

- Step 2** The software should set registers. Configure the first address of the task descriptor structure to [CE Task Descriptor Address Register](#). Configure the source/destination address to [CE Current Source Address Register/CE Current Destination Address Register](#).
- Step 3** Enable the end interrupt of the corresponding task channel by setting [CE Interrupt Control Register](#).
- Step 4** The software reads [CE Task Load Register](#) to ensure that the bit0 is 0. If the bit0 is not read out to be 0, wait until it is 0, then configure the bit0 to be 1 to start task.
- Step 5** Wait for interrupt status by reading [CE Interrupt Status Register](#).
- Step 6** Read the result from the destination address.
- Step 7** Clear the interrupt.

#### 9.1.4.2 Configuring Task Descriptor of AES

- **Common control:** Configure [Common Control](#)[6:0] to 0x0 to select AES algorithm type.
- **Symmetric control:** According to the corresponding algorithm requirements, configure [Symmetric Control](#) to select the key size, CTR width, CTS last package flag, CFB width, and AES algorithm mode, and so on.
- **Asymmetric control:** The symmetric algorithm does not need to be configured for this field.
- **Key descriptor:** Because the storage of the key requires word alignment, ensure that this descriptor is the first address of the KEY (word address).
- **IV descriptor:** In the task that requires the IV value, configure the first address of the storage space where the IV is stored here. Because the storage of the IV requires word alignment, ensure that this descriptor is the first address of the IV (word address).
- **Data length:** Configure the data length of the corresponding segment. The data length size needs to be consistent with dst\_data\_length (destination data length 0 +... + destination data length 7). When the algorithm is CTS mode, the higher 30-bit of the data length is the word numbers of data volume; when the data\_length[1:0] is 0, the data length is the higher 30-bit, otherwise it is increased by 1. For AES CTS, the data length indicates the byte numbers of the source data; for other algorithms, it indicates the word numbers.
- **Source address:** The first address of source data segments. Because the storage of the source data requires word alignment, ensure that this descriptor is the first address (word address).
- **Source data length:** The data volume of source data segments. The unit is word, and those less than one word are counted as one word. Note that only the last word of the entire message is allowed to be non-integer words, and the others must be integer words.
- **Destination address:** The first address of destination data segments. Because the storage of the destination data requires word alignment, ensure that this descriptor is the first address (word address).

- **Destination data length:** The data volume of destination data segments. The unit is word, and those less than one word are counted as one word. Note that only the last word of the entire message is allowed to be non-integer words, and the others must be integer words.
- **Next descriptor:** The first address of the next task descriptor. Because the storage of the descriptor requires word alignment, ensure that this descriptor is the first address (word address).
- **Reserved:** Configure to 0x0.

#### 9.1.4.3 Configuring Task Descriptor of HASH

- **Common control**
  - **Algorithm type:** Configure [Common Control](#)[6:0] to select SHA or HMAC algorithm type.
  - **Last HMAC plaintext:** If the algorithm type is HMAC, and the task is the last package of the message or if the message has only one package, then [Common Control](#)[15] needs to set to 1.
  - **IV mode:** The [Common Control](#)[16] (IV MODE) bit is only set to 1 in the following two scenarios, except that the bit must be configured to 0. (1). When the message is split into multiple packages, the [Common Control](#)[16] bit of other packages needs to be set to 1, except that the bit of the first package needs to be cleared to 0. (2). In special applications, if you need to customize the IV value to form the initial value of a certain HASH algorithm, you need to set the [Common Control](#)[16] bit of the first (or only one) package to 1, and the first address of the storage space where the customized IV value is stored in IV address.
- **Key descriptor:** Because the storage of the key requires word alignment, ensure that this descriptor is the first address of the KEY (word address).
- **IV descriptor:** In the task that requires the IV value, configure the first address of the storage space where the IV is stored here. Because the storage of the IV requires word alignment, ensure that this descriptor is the first address of the IV (word address).
- **Data length:** Configure the data length of the corresponding segment. The data length size needs to be consistent with dst\_data\_length (destination data length 0 +...+ destination data length 7).
- **Source address:** The first address of source data segments. Because the storage of the source data requires word alignment, ensure that this descriptor is the first address (word address).
- **Source data length:** The data volume of source data segments. The unit is word, and those less than one word are counted as one word. Note that only the last word of the entire message is allowed to be non-integer words, and the others must be integer words.
- **Destination address:** The first address of destination data segments. Because the storage of the destination data requires word alignment, ensure that this descriptor is the first address (word address).

- **Destination data length:** The data volume of destination data segments. The unit is word, and those less than one word are counted as one word. Note that only the last word of the entire message is allowed to be non-integer words, and the others must be integer words.
- **Next descriptor:** The first address of the next task descriptor. Because the storage of the descriptor requires word alignment, ensure that this descriptor is the first address (word address).
- **Reserved:** Configure to 0x0.

#### 9.1.4.4 Configuring Task Descriptor of RSA

- **Common control:** Configure [Common Control](#)[6:0] to 0x20 to select RSA algorithm type.
- **Asymmetric control:** Configure [Asymmetric Control](#)[30:28] to select the RSA width.
- **Key descriptor:** Because the storage of the key requires word alignment, ensure that this descriptor is the first address of the KEY (word address).
- **Data length:** Configure the data length of the corresponding segment. The data length size needs to be consistent with dst\_data\_length (destination data length 0 +... + destination data length 7).
- **Source address:** The first address of source data segments. Because the storage of the source data requires word alignment, ensure that this descriptor is the first address (word address).
- **Source data length:** The data volume of source data segments. The unit is word, and those less than one word are counted as one word. Note that only the last word of the entire message is allowed to be non-integer words, and the others must be integer words.
- **Destination address:** The first address of destination data segments. Because the storage of the destination data requires word alignment, ensure that this descriptor is the first address (word address).
- **Destination data length:** The data volume of destination data segments. The unit is word, and those less than one word are counted as one word. Note that only the last word of the entire message is allowed to be non-integer words, and the others must be integer words.
- **Next descriptor:** The first address of the next task descriptor. Because the storage of the descriptor requires word alignment, ensure that this descriptor is the first address (word address).
- **Reserved:** Configure to 0x0.

#### 9.1.5 Register List

Module Name	Base Address
CE_NS (Non-Security CE)	0x03040000

Module Name	Base Address
CE_S (Security CE)	0x03040800

Register Name	Offset	Description
CE_TDA	0x0000	Task Descriptor Address
CE_CTL	0x0004	Control Register
CE_ICR	0x0008	Interrupt Control Register
CE_ISR	0x000C	Interrupt Status Register
CE_TLR	0x0010	Task Load Register
CE_TSR	0x0014	Task Status Register
CE_ESR	0x0018	Error Status Register
CE_CSA	0x0024	DMA Current Source Address
CE_CDA	0x0028	DMA Current Destination Address
CE_TPR	0x002C	Throughput Register

### 9.1.6 Register Description

#### 9.1.6.1 0x0000 CE Task Descriptor Address Register (Default Value: 0x0000\_0000)

Offset: 0x0000			Register Name: CE_TDA
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	Task Descriptor Address Configure as the first address of the descriptor structure.

#### 9.1.6.2 0x0004 CE Control Register (Default Value: 0x0000\_0000)

Offset: 0x0004			Register Name: CE_CTL
Bit	Read/Write	Default/Hex	Description
31:4	/	/	/
3	R/W	0x0	RSA CLK Gating Enable (only for CE_S) 0: RSA clock gating enable 1: RSA clock gating disable
2:0	/	/	/

### 9.1.6.3 0x0008 CE Interrupt Control Register (Default Value: 0x0000\_0000)

Offset: 0x0008			Register Name: CE_ICR
Bit	Read/Write	Default/Hex	Description
31:4	/	/	/
3:0	R/W	0x0	Task Channel3~0 Interrupt Enable 0: Disable 1: Enable

### 9.1.6.4 0x000C CE Interrupt Status Register (Default Value: 0x0000\_0000)

Offset: 0x000C			Register Name: CE_ISR
Bit	Read/Write	Default/Hex	Description
31:4	/	/	/
3:0	R/W1C	0x0	Task Channel3~0 End Pending 0: Not finished 1: Finished It indicates whether task is completed. Write the corresponding channel bit of the register to clear the end flag. When the last task in the task chain ends, the operation of the task chain will end normally. If the task fails in the middle, the task chain will be aborted. The CE_ISR register will be automatically updated when it ends normally or aborts abnormally. And it is determined whether to generate an interrupt signal according to the IE configuration (bit31) of <a href="#">Common Control</a> when the current task ends or aborts. If using interrupt, after receiving the interrupt, read the corresponding channel bit of CE_ISR to judge whether it ends successfully or stops failure. If not using interrupt, the CE_ISR status register can be continuously queried for the channel bit until the successful end flag is set or the failure stop flag is set. Write the corresponding channel bit of the register to clear the end flag. If it fails to stop, you can read the error code on the channel corresponding to the <a href="#">CE_ESR</a> register.

#### 9.1.6.5 0x0010 CE Task Load Register (Default Value: 0x0000\_0000)

Offset: 0x0010			Register Name: CE_TLR
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	Task Load When set, the CE can load the descriptor of task if the task FIFO is not full.

#### 9.1.6.6 0x0014 CE Task Status Register (Default Value: 0x0000\_0000)

Offset: 0x0014			Register Name: CE_TSR
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1:0	R	0x0	Running Channel Number 00: Task channel0 01: Task channel1 10: Task channel2 11: Task channel3

#### 9.1.6.7 0x0018 CE Error Status Register (Default Value: 0x0000\_0000)

Offset: 0x0018			Register Name: CE_ESR
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:12	R/W1C	0x0	Task Channel3 Error Type xxx1: Algorithm not support xx1x: Data length error x1xx: keysram access error for AES 1xxx: Reserved
11:8	R/W1C	0x0	Task Channel2 Error Type xxx1: Algorithm not support xx1x: Data length error x1xx: keysram access error for AES 1xxx: Reserved

Offset: 0x0018			Register Name: CE_ESR
Bit	Read/Write	Default/Hex	Description
7:4	R/W1C	0x0	Task Channel1 Error Type xxx1: Algorithm not support xx1x: Data length error x1xx: keysram access error for AES 1xxx: Reserved
3:0	R/W1C	0x0	Task Channel0 Error Type xxx1: Algorithm not support xx1x: Data length error x1xx: keysram access error for AES 1xxx: Reserved

#### 9.1.6.8 0x0024 CE Current Source Address Register (Default Value: 0x0000\_0000)

Offset: 0x0024			Register Name: CE_CSA
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	CUR_SRC_ADDR Current source address

#### 9.1.6.9 0x0028 CE Current Destination Address Register (Default Value: 0x0000\_0000)

Offset: 0x0028			Register Name: CE_CDA
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	CUR_DST_ADDR Current destination address

#### 9.1.6.10 0x002C CE Throughput Register (Default Value: 0x0000\_0000)

Offset: 0x002C			Register Name: CE_TPR
Bit	Read/Write	Default/Hex	Description
31:0	R/WC	0x0	TP_NUM It indicates the throughput writing to this register at last time. Writing to this register will clear it to 0.

## 9.2 Security ID

The Security ID (SID) is used to program and read keys which include chip ID, thermal sensor, HASH code, and security key, and so on.

The SID module has the following features:

- 2 Kbit electrical fuse (eFuse)
- The module register is non-secure forever, the eFuse has secure zone and non-secure zone
- Backup eFuse information by using SID\_SRAM
- A fuse only can program one time
- Burning the key to the SID
- Reading the key use status in the SID
- Loading the key to the CE



Before performing the burning operation, ensure that the power supply of the eFuse power pin is stable. After the burning operation is completed, cancel the power supply of the eFuse power pin.

---

## Appendix: Glossary

The following table contains acronyms and abbreviations used in this document.

Term	Meaning
<b>A</b>	
ADC	Analog-to-Digital Converter
AE	Automatic Exposure
AEC	Audio Echo Cancellation
AES	Advanced Encryption Standard
AF	Automatic Focus
AGC	Automatic Gain Control
AHB	AMBA High-Speed Bus
ALC	Automatic Level Control
ANR	Active Noise Reduction
APB	Advanced Peripheral Bus
ARM	Advanced RISC Machine
AVS	Audio Video Synchronization
AWB	Automatic White Balance
<b>B</b>	
BROM	Boot ROM
<b>C</b>	
CIR	Consumer Infrared
CMOS	Complementary Metal-Oxide Semiconductor
CP15	Coprocessor 15
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
<b>D</b>	
DDR	Double Data Rate
DES	Data Encryption Standard
DLL	Delay-Locked Loop
DMA	Direct Memory Access
DRC	Dynamic Range Compression
DVFS	Dynamic Voltage and Frequency Scaling
<b>E</b>	
ECC	Error Correction Code
eFuse	Electrical Fuse, A one-time programmable memory
EHCI	Enhanced Host Controller Interface
eMMC	Embedded Multi-Media Card
ESD	Electrostatic Discharge
<b>F</b>	
FBGA	Fine Pitch Ball Grid Array
FEL	Fireware Exchange Launch
FIFO	First In First Out

Term	Meaning
<b>G</b>	
GIC	Generic Interrupt Controller
GPIO	General Purpose Input Output
<b>I</b>	
I2C	Inter Integrated Circuit
I2S	Inter IC Sound
<b>J</b>	
JEDEC	Joint Electron Device Engineering Council
JPEG	Joint Photographic Experts Group
JTAG	Joint Test Action Group
<b>L</b>	
LCD	Liquid-Crystal Display
LFBGA	Low Profile Fine Pitch Ball Grid Array
LSB	Least Significant Bit
LVDS	Low Voltage Differential Signaling
<b>M</b>	
MAC	Media Access Control
MIC	Microphone
MIPI	Mobile Industry Processor Interface
MLC	Multi-Level Cell
MMC	Multimedia Card
MPEG	Motion Pictures Expert Group
MSB	Most Significant Bit
<b>N</b>	
N/A	Not Application
NMI	Non Maskable Interrupt
NTSC	National Television Standards Committee
NVM	Non Volatile Storage Medium
<b>O</b>	
OHCI	Open Host Controller Interface
OTP	One Time Programmable
OWA	One Wire Audio
<b>P</b>	
PAL	Phase Alternating Line
PCM	Pulse Code Modulation
PHY	Physical Layer Controller
PID	Packet Identifier
PLL	Phase-Locked Loop
POR	Power-On Reset
PRCM	Power Reset Clock Management
PWM	Pulse Width Modulation
<b>R</b>	
R	Read only/non-Write

Term	Meaning
RGB	Read Green Blue
RGMII	Reduced Gigabit Media Independent Interface
RMII	Reduced Media Independent Interface
ROM	Read Only Memory
RSA	Rivest-Shamir-Adleman
RTC	Real Time Clock
<b>S</b>	
SAR	Successive Approximation Register
SD	Secure Digital
SDIO	Secure Digital Input Output
SDK	Software Development Kit
SDRAM	Synchronous Dynamic Random Access Memory
SDXC	Secure Digital Extended Capacity
SLC	Single-Level Cell
SoC	System on Chip
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
<b>T</b>	
TDES	Triple Data Encryption Standard
TWI	Two Wire Interface
<b>U</b>	
UART	Universal Asynchronous Receiver Transmitter
UDF	Undefined
USB DRD	Universal Serial Bus Dual Role Device
UTMI	USB2.0 Transceiver Macrocell Interface

**Copyright©2022 Allwinner Technology Co.,Ltd. All Rights Reserved.**

This documentation is the original work and copyrighted property of Allwinner Technology Co.,Ltd ("Allwinner"). No part of this document may be reproduced, modify, publish or transmitted in any form or by any means without prior written consent of Allwinner.

**Trademarks and Permissions**

Allwinner and the Allwinner logo (incomplete enumeration) are trademarks of Allwinner Technology Co.,Ltd. All other trademarks, trade names, product or service names mentioned in this document are the property of their respective owners.

**Important Notice and Disclaimer**

The purchased products, services and features are stipulated by the contract made between Allwinner Technology Co.,Ltd ("Allwinner") and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Please read the terms and conditions of the contract and relevant instructions carefully before using, and follow the instructions in this documentation strictly. Allwinner assumes no responsibility for the consequences of improper use (including but not limited to overvoltage, overclock, or excessive temperature).

The information in this document is provided just as a reference or typical applications, and is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents. Allwinner is not responsible for any damage (including but not limited to indirect, incidental or special loss) or any infringement of third party rights arising from the use of this document. All statements, information, and recommendations in this document do not constitute a warranty or commitment of any kind, express or implied.

No license is granted by Allwinner herein express or implied or otherwise to any patent or intellectual property of Allwinner. Third party licences may be required to implement the solution/product. Customers shall be solely responsible to obtain all appropriately required third party licences. Allwinner shall not be liable for any licence fee or royalty due in respect of any required third party licence. Allwinner shall have no warranty, indemnity or other obligations with respect to third party licences.